A Survey on Handling Data in Business Process Models (Discussion Paper)

Andrea Marrella[1], Massimo Mecella[1], Alessandro Russo[1],
Sebastian Steinau[2], Kevin Andrews[2], and Manfred Reichert[2]

[1] Sapienza Università di Roma, Dip. di Ingegneria Informatica, Automatica e Gestionale
{marrella,mecella,arusso}@dis.uniroma1.it
[2] Ulm University, Institute of Databases and Information Systems
{sebastian.steinau,kevin.andrews,manfred.reichert}@uni-ulm.de

Abstract. Traditional activity-centric process modeling languages treat data as simple black boxes acting as input or output for activities. Many alternate and emerging process modeling paradigms, such as case handling and artifact-centric process modeling, give data a more central role. This is achieved by introducing lifecycles and states for data objects, which is beneficial when modeling data- or knowledge-intensive processes. We assume that traditional activity-centric process modeling languages lack the capabilities to adequately capture the complexity of such processes. To verify this assumption, we conducted a survey among Business Process Management experts. The survey results allow us to identify the problems of contemporary modeling languages in regard to the modeling of business data. To this end, survey respondents rated the data modeling capabilities of a variety of business process modeling tools and notations. Overall, the paper confirms the need of data-awareness in process modeling notations in general.

1 Introduction

In recent years, an ever increasing interest in Business Process Management (BPM) approaches and technologies could be witnessed. Nowadays, the automation of processes not only spans classical business domains (e.g., banks and governmental agencies), but also new settings such as healthcare [10] or the coordination of workforces in the field (e.g. during emergencies on building sites). More and more such processes are cyber-physical, as the information flowing through the process is often produced either manually by human activities or is acquired by sensors and software services. In turn, the effects of the processes are not only visible in information systems, but also in the real world through actuators. Consequently, the execution context of processes is becoming more complex as increasing amounts of data influence the running of process instances.

Accompanying this trend, the interest of both practitioners and researchers is shifting from the simple modeling of the control flow to more advanced features for treating data as a first class citizen in modeling approaches. Recent research developments, known in literature as object-aware processes, artifact-centric approaches, data-driven processes, or case handling are receiving increasing attention from the process management community. In this context, this paper presents empirical research that aims to document the thoughts of practitioners and researchers on the topic of data awareness in

BPM, in order to understand what are the needs in terms of modeling and enactment capabilities for data. Specifically, Section 2 provides an overview of existing activity- and data-centric process modeling approaches. Section 3 describes the methodology underlying the empirical research and presents the main results of the analysis performed, which are then discussed in Section 4 to provide a critical view and insights on them.

2 Background

Traditional notations for business process modeling are imperative and activity-centric, i.e., a process is composed of activities representing units of work and control flow elements determine the order of activity execution. Examples of graphical activity-centric notations, mostly used for documenting business processes, include the Business Process Model and Notation (BPMN), Event-driven Process Chains (EPC) and UML Activity Diagrams (UML AD). Additionally, code-based activity-centric notations, such as the Web Service Business Process Execution Language (WS-BPEL) exist, providing a way to specify processes executable in process management systems. Activity-centric processes may also be defined in a declarative fashion with notations such as Declare [7]. In both imperative and declarative approaches, data is represented by data elements, which can be used as input or output for activities.

Alternatively to the activity-centric paradigm, processes may be specified using the data-centric paradigm. A data-centric process progresses based on the availability of data and their values at a given point in time. Artifact-centric process models [4] are a specific form of data-centric process models. An artifact consists of an information model holding relevant data, as well as a lifecycle model which describes possible changes to the information model and interactions with other artifacts. The lifecycle model of an artifact can be defined imperatively, using a finite state machine, or declaratively with the help of the Guard-Stage-Milestone (GSM) meta model [5]. GSM is a rule-based framework that allows to define the lifecycle of an artifact using stages associated with guards and milestones. Stages group individual activities and may be nested within other stages. Guards provide entry conditions to a stage. Milestones represent operational objectives and are completed by fulfilling their associated conditions. GSM provides also the basis for the Case Management Model and Notation (CMMN). Case management [9] (often referred to as case handling) focuses on the case as the central element, e.g., a medical or judicial case, and constitutes a data-driven paradigm for modeling flexible processes. Process participants may see all information relevant to a case, instead of just getting fragmented, task-centered, views.

Recently, the framework of relational Data-Centric Dynamic Systems (DCDSs) was proposed for the formal specification and verification of data-centric processes [1]. A DCDS fully captures the connection and interplay between the process and the data perspectives, and can be considered as a pristine formalization of the artifact-centric variants (including GSM). Basically, a DCDS includes a relational data layer, holding the data of interest, and a process layer characterizing the dynamic behavior of the system and evolving the data based on a declarative rule-based process specification.

Finally, PHILharmonicFlows [6] is a framework for modeling and executing object-aware business processes, whose basic concepts are similar to those of artifact-centric

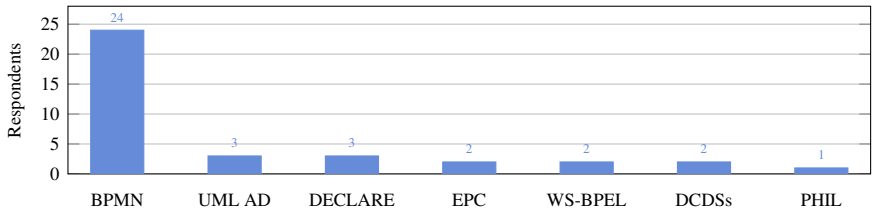

Fig. 1. Usage of the process modeling languages among the respondents

process models. However, the processes describing the interactions between the artifacts, referred to as objects in object-aware process management, are separated from those processes that describe the lifecycle of the objects. Objects reach states in the course of their lifecycle, depending on available data. The interactions between objects are coordinated on a higher level of granularity, depending on the states of the objects.

3 Research Methodology and Survey Results

The empirical research of this paper aims at understanding how data-awareness is perceived in current R&D activities in the BPM field. To this end, we conducted a survey, asking experts from research and industry what their intuition of this concept is. Specifically, we built an online questionnaire based on an Internet surveying technique called CAWI (Computer-Assisted Web Interviewing [2]), and we provided it via personal e-mail to the potential respondents. We originally invited 68 representatives from the following groups: *(i) researchers* who have a relation with the BPM discipline, *(ii)* workflow *industry experts*, and *(iii)* BPM *practitioners*. The questionnaire was made available to the invited participants between September and mid-December 2014. During this period, a total of 37 respondents completed the questionnaire, with a response rate of about 54%. The questionnaire was designed using a three-pronged strategy: *(i)* get information on the perceived value of data-awareness in the BPM field, *(ii)* get information on the current support provided by existing process modeling languages and tools with respect to data management, and *(iii)* get feedback on what features are required to inject data-awareness into existing business process modeling languages and tools.

In the following we present an overview of the main results of the analysis performed on the data collected through the survey. While we recognize that the available dataset is relatively small, thus limiting the ability to draw statistically significant conclusions and generalize the results beyond the sample population, we believe that both a qualitative and quantitative analysis can provide relevant insights and findings.

Participants profile. The vast majority of the 37 participants are *academic researchers* in BPM and related topics (26 respondents, 70%), while 7 respondents (19%) are BPM *practitioners*. The participation of *industrial researchers* (2 respondents) and BPM *end users* (1 respondent) was quite limited. Figure 1 shows that BPMN is the most used modeling language among the respondents (65%). Among the 24 respondents using BPMN as their primary modeling language, there are academic researchers (18), BPM

practitioners (4), an industrial researcher and a respondent who reports having expertise in all fields. Languages such as Declare, DCDSs or PHILharmonicFlows are used only by academic researchers. This can be easily explained when considering the academic nature of these languages and the low level of maturity and tool support, especially in comparison with a standardized and widely supported language such as BPMN. Furthermore, UML AD are adopted by BPM practitioners (2 respondents) and end users (1 respondent), while EPC are used by one BPM practitioner and by an industrial researcher. No respondent stated to use artifact-centric languages or CMMN.

Domains of process modeling. Reported application domains span multiple fields, ranging from research-oriented academic use cases, to real-world scenarios involving healthcare providers, manufacturing, the public sector, and even energy companies. Specifically, we observed a multi-domain applicability of BPMN, while Declare, DCDSs and PHILharmonicFlows are mainly used in academic examples with a tendency towards healthcare scenarios (where declarative models and data-awareness play a fundamental role). Finally, EPC users target the automotive domain, WS-BPEL users focus on Business-to-Business integration scenarios and UML AD are used in traditional BPM scenarios, including financial services and administrative settings.

Control flow and data modeling. Moving closer to our research objective, survey participants rated how easy they perceive the definition of control flow and process data in the modeling language they declared to mainly use. The questions were answered using a 5-level Likert scale and the results are summarized in Figure 2. As shown in the first chart, most of the respondents who use an activity-centric imperative language (BPMN, EPC, UML AD, and WS-BPEL) rate control flow modeling as *easy*. On the other hand, despite the powerful declarative constraint-based approach characterizing Declare, 2 out of 3 respondents using this language consider it *difficult* to model control flow. Additionally, it is interesting to note that the respondents using languages explicitly designed to be data-aware (DCDSs and PHILharmonicFlows) have a rather *neutral* opinion on the easiness of control flow modeling. However, when focusing on data modeling (see the second chart in Figure 2), the advantages of data-centric languages become evident as DCDSs and PHILharmonicFlows users rate the definition of process data as an *easy* or *very easy* task. Similarly, the low degree of support provided by Declare concerning the data perspective is confirmed by 2 respondents who consider data modeling in Declare as *very difficult*, whereas, surprisingly, 1 respondent rates it as *easy*. With the exception of WS-BPEL users (both consider data modeling as an *easy* task), there is no common perception of the simplicity of data modeling among respondents using the other activity-centric imperative languages (BPMN, EPC and UML AD). In general, we can observe a shift towards negative answers (on left side of the chart), in comparison with the positive ratings for control flow modeling simplicity.

Kind of data objects. Participants were also asked to specify the kind of data objects required in their process models. Interestingly, data objects are not limited to *atomic data elements*, as reported by 26 of our respondents (70%). Many participants go beyond atomic elements and also deal with complex object types and their relationships. In detail, 18 respondents (49%) deal with object types with one instance at run-time and 22 respondents (59%) consider object types with several instances at run-time. Moreover, 46% of the respondents recognise that even during the execution of their processes, the

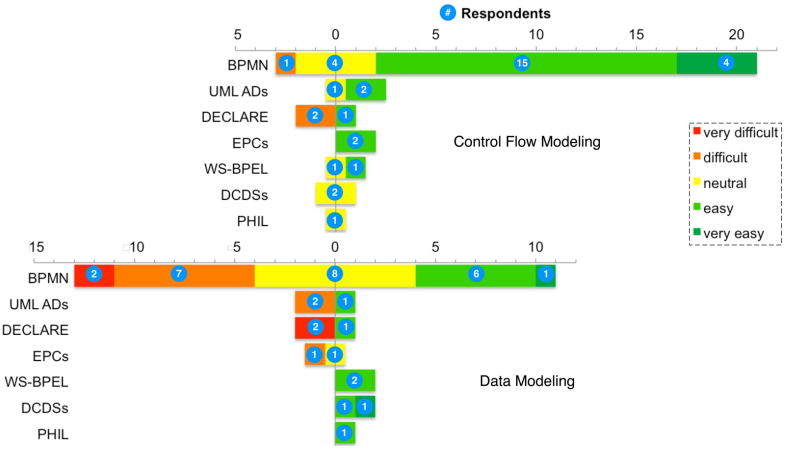

Fig. 2. Ratings for modeling control flow and in the different languages

relationships between object instances need to be considered. 23% of the respondents also provided additional free-text answers. In particular, some respondents highlight the correspondence between process data and business documents, while 1 respondent focuses on scenarios where it is required to both correlate object instances to process instances and enable the use of one object instance in multiple process instances.

Role of data objects in process models. While some respondents report that data objects play a very minor role, others provide detailed answers describing the importance of data at different levels of abstraction. For example, EPC users, both dealing with processes in the automotive domain, focus on data elements used to drive *branching decisions* and define or capture data required for forms. However, as these data may be rather complex, a respondent highlights that they are only informally stored in a "free text" attribute, due to the lack of language support. The role of data objects as a means for *(i)* capturing domain-relevant data, *(ii)* defining I/O elements for individual tasks and *(iii)* expressing split conditions and recording decisions is also reported by a respondent using UML AD and by several BPMN users. In some cases, data objects are defined for documentation purposes and on a very abstract level, i.e, for documenting data required for performing an activity or provided by an activity. However, when data management has to go beyond simple documentation purposes and requires a detailed modeling and a concrete implementation, our respondents using activity-centric languages (specifically, UML AD and BPMN) highlight the need to rely on external data management tools. In particular, some of the answers indicate that business data objects are often not modeled using features of the process language (e.g., BPMN), but rather relying on specific data modeling languages, such as UML diagrams and entity-relationship diagrams. This then requires, in turn, to design and build an integrated and consistent information architecture where data objects in the process are linked to their concrete implementation in the underlying data management subsystem, e.g., through object-relational mapping (ORM) techniques. The prominent and complex role of data objects also emerges from the comments of the respondents using data-centric or object-aware languages. While

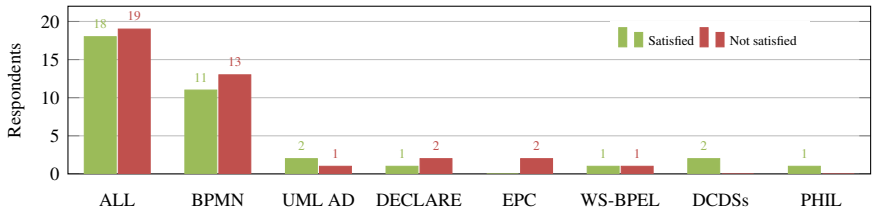

Fig. 3. Respondents' satisfaction and modeling languages

the PHILharmonicFlows user reports that data is vital to capture *user decisions*, DCDSs users relate data objects to the definition of the key domain entities/artifacts that drive the dynamics of the system and are simultaneously used to maintain information about both the business domain and the process execution.

Participant satisfaction. As reported in Figure 3, we also investigated on participants' satisfaction with respect to the data modeling capabilities provided by the language they stated to use primarily. In the case of respondents using an activity-centric language (BPMN, EPC, UML AD and WS-BPEL), we generally observe a slight prevalence of negative answers, whereas the respondents using data-centric or object-aware languages (DCDSs and PHILharmonicFlows) stated that they are satisfied with the data modeling capabilities provided by those languages. To sum up, the overall 37 respondents are almost equally partitioned into *satisfied* (49%) and *unsatisfied* (51%) users.

Data awareness for satisfied users. Satisfied users indicate a common perception of *data awareness*, which is often defined as "being aware of the business data model and data flow in the specific process domain, between data creation to data delivery to end users". This is complemented by the possibility of explicitly modeling and representing both the data elements and the data flow. The claim that data awareness is already present in the modeling languages is generally motivated by referring to the availability of modeling constructs to represent data elements and data flows. These modeling features include graphical notational symbols to represent data at the business level (as provided by UML AD and BPMN to model data objects, persistent data stores and data flows), as well as the possibility of modeling data and data flows via XML schemas and XPath expressions (cf. WS-BPEL). More focused definitions are given by the respondents using data-centric languages. One of the DCDSs users defines data awareness as the "ability of a process modeling language to explicitly account for data at the extensional and intensional level, and to explicitly tackle how they interact with the process control-flow". This covers both how the data drives the process execution, and how the process manipulates the data over time. Similarly, the PHILharmonicFlows user refers to data awareness as the fact that the processes are not only centered on the activities that must be performed, but also in the data relevant to the process, by letting process participants access process data at any point in time during process execution.

Data awareness for unsatisfied users. Unsatisfied responders provide definitions of *data awareness* that go beyond the ability to explicitly model data elements produced/consumed by process activities and specify how this data is used for making

decisions within a process instance. One of the Declare users, for example, defines data awareness as "the explicit representation of the intertwining among control-flow, resources, contextual information, and side effects on/from data changes at large". While satisfied responders using activity-centric languages mainly consider the data perspective *subordinate* to the control flow, unsatisfied respondents give an "equal importance to control flow and data flow" so that, for example, enabling activities is driven by both the control flow and the data flow. According to BPMN users, data awareness relates to the capability of expressing the influence of *complex* data objects on activities and entire processes, and vice versa. Data objects and their corresponding instances must be *distinguishable* and possible *dependencies* between them must be considered. Moreover, at the instance level, current values of these objects as well as the information about changes should be available to the process, so that it can react to data changes.

Injecting data awareness into process modeling languages. Unsatisfied respondents provide an extensive list of features they would like have in the modeling languages to make them "data-aware". The limited support for data elements in Declare leads its users to crave basic features, including the possibility of defining data variables in the description of activities, data monitoring points and data checks on activity constraints. Although she considers the language as data-aware, the WS-BPEL user requires more flexible ways of defining and modifying data objects, as the fixed focus on XML prevents an easy handling of data elements. In the case of UML AD, the respondents highlight the lack of support for making more explicit the *semantical* relation between processes and processed data objects. Required features thus include: *(i)* the modeling of data-dependent *rule-based* control flow aspects, possibly easy to read and maintain; *(ii)* the explicit definition of *process states* whose reachability depends on user-defined constraints expressed over *data object states*; *(iii)* the possibility of *verifying* data flows in the models. BPMN users also stress the need to define (or improve) the expressiveness of *data objects semantics* by considering data objects' behavior and lifecycle.

4 Discussion and Conclusion

Providing a reference definition of "data awareness" is not easy. As confirmed by the survey, the perception of the role of data in business processes is highly subjective and hence varies considerably. The same holds when evaluating current support of the data perspective by existing process modeling languages. On one hand, it could be claimed that activity-centric languages were originally defined to support control flow modeling. Thus, the lack of a more advanced support for the data perspective should be considered as a *design choice* rather than as a missing feature. On the other, when dealing with real-world scenarios and processes, it often can be observed that the support of the data perspective is limited (even in terms of input/output parameters), preventing the successful adoption of contemporary process management technology in practice.

Typically, the notational symbols provided by activity-centric languages for defining data objects and data flows are sufficient to represent the data perspective of processes at a high level of abstraction, e.g., for documentation purpose or for discussing them with business stakeholders. However, when it comes to concretely implement the modeled processes as well as to manage complex and evolving data structures, the lack

of a properly defined *data semantics* becomes a major obstacle for both process designers and engineers [11]. Thus, it is common practice to combine data and process engineering methods. However, these are applied rather independently and at different layers of an information system resulting in high maintenance efforts—in [3] this phenomenon is also denoted as *impedance mismatch* between process layer on one hand and data layers on the other. In order to capture the complex interdependencies as well as to integrate the control with the data perspective, the definition of *process state* must be extended to take data objects and their lifecyles into account as well. Due to the widespread use of BPMN one may argue that it would be best to extend BPMN to overcome its current limitations in respect to the data perspective. According to one of the respondents, however, extending BPMN towards data-awareness would weaken some of its existing properties and make BPMN process models even more complex to understand. As discussed by [8], BPMN is already an "over-engineered" language. Hence, adding a complex set of data-related properties and features might make it unusable.

All properties missing in activity-centric approaches with respect to the data perspective are more or less provided by data- and object-aware process modeling approaches. The survey confirms that the different groups working on data- and object-aware process support have done a good job and are moving in the "right" direction. Although considerable progress has been made, however, it should be clear that existing implementations of corresponding tools have not yet reached the same level of maturity as activity-centric modeling tools. In this context, based on the preliminary findings of the survey, we plan to conduct a systematic literature review (SLR), creating a comprehensive overview on the current research regarding data- and object-aware business process support and devising a framework for evaluating respective approaches.

References

1. Bagheri Hariri, B., Calvanese, D., De Giacomo, G., Deutsch, A., Montali, M.: Verification of Relational Data-centric Dynamic Systems with External Services. In: Proc. PODS'13 (2013)
2. Dix, A., Finlay, J., Abowd, G., Beale, R.: Human-Computer Interaction. Prentice Hall (2004)
3. Dumas, M.: On the convergence of data and process engineering. In: Proc. ADBIS'11 (2011)
4. Hull, R.: Artifact-Centric Business Process Models: Brief Survey of Research Results and Challenges. In: Proc. OTM'2008, pp. 1152–1163 (2008)
5. Hull, R., et al.: Introducing the Guard-Stage-Milestone Approach for Specifying Business Entity Lifecycles. In: Proc. Web Services and Formal Methods. pp. 1–24 (2011)
6. Künzle, V., Reichert, M.: Towards Object-Aware Process Management Systems: Issues, Challenges, Benefits. In: Proc. BPMDS'09, pp. 197–210 (2009)
7. Pesic, M., Schonenberg, H., van der Aalst, W.M.P.: DECLARE: Full Support for Loosely-Structured Processes. In: Proc. EDOC'07. p. 287 (2007)
8. Recker, J.C.: Opportunities and constraints : the current struggle with BPMN. Business Process Management Journal 16(1), 181–201 (2010)
9. Reijers, H.A., Rigter, J. H. M., van der Aalst, W. M. P.: The Case Handling Case. Int J Coop Inf Sys 12(03), 365–391 (2003)
10. Russo, A., Mecella, M.: On the evolution of process–oriented approaches for healthcare workflows. Int J Business Process Integration and Management 6(3), 224–246 (2013)
11. Weber, B., Mutschler, B., Reichert, M.: Investigating the effort of using business process management technology: Results from a controlled experiment. Sci Comp Program 75(5), 292–310 (2010)