Data in Business Process Models. A Preliminary Empirical Study

Andrea Marrella*, Massimo Mecella*, Alessandro Russo*,
Sebastian Steinau†, Kevin Andrews†, Manfred Reichert†
*Sapienza Università di Roma, Italy - Email: {firstname.lastname}@uniroma1.it
†Ulm University, Germany - Email: {firstname.lastname}@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 an online interview among BPM experts. The results not only allow us to identify various profiles of persons modeling business processes, but also the problems that exist in contemporary modeling languages w.r.t. the modeling of business data. Overall, this preliminary empirical study confirms the necessity of data-awareness in process modeling notations in general.**

I. Introduction

In recent years an ever increasing interest in process management approaches, methods and technologies could be witnessed. Nowadays, the automation of workflows not only spans classical business domains (e.g., insurances, banks and governmental agencies), but also new settings such as healthcare [1] or the coordination of workforces in the field [2]. More and more such processes are cyber-physical [3], 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. The execution context of processes is becoming more and more complex as increasing amounts of data influence the execution of the 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. Traditional approaches to workflow modeling and verification focus on formalisms like Petri nets. More 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 order to fully understand what is really perceived as data awareness and what are the needs in terms of modeling and enactment capabilities for data. The aim of this work is to provide, if any, an empirical justification to the research work currently undertaken on data awareness in Business Process Management (BPM). If the results of the study do not confirm this, however, it should be clarified why, according to the community, there is no need for data-awareness in BPM.

The rest of the paper is organized as follows: Section II describes relevant background information on activity-centric and data-aware process modeling approaches. Section III introduces the methodology underlying the empirical research. Section IV presents the findings obtained when performing the survey, which are then discussed in Section V to provide a critical view and insights on them; finally, we conclude the paper by outlining future developments of this research.

II. 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), UML Activity Diagrams (UML AD), YAWL/Workflow nets. 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 [4]. 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 [5] 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 [7]. 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 [11]. 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 [13] is a framework for modeling and executing object-aware business processes, whose basic concepts are similar to those of artifact-centric 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.

III. Research Methodology

The empirical research of this paper aims at understanding how data-awareness is perceived in current research & development 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 (CAWI – Computer-Assisted Web Interviewing), 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.

IV. Results

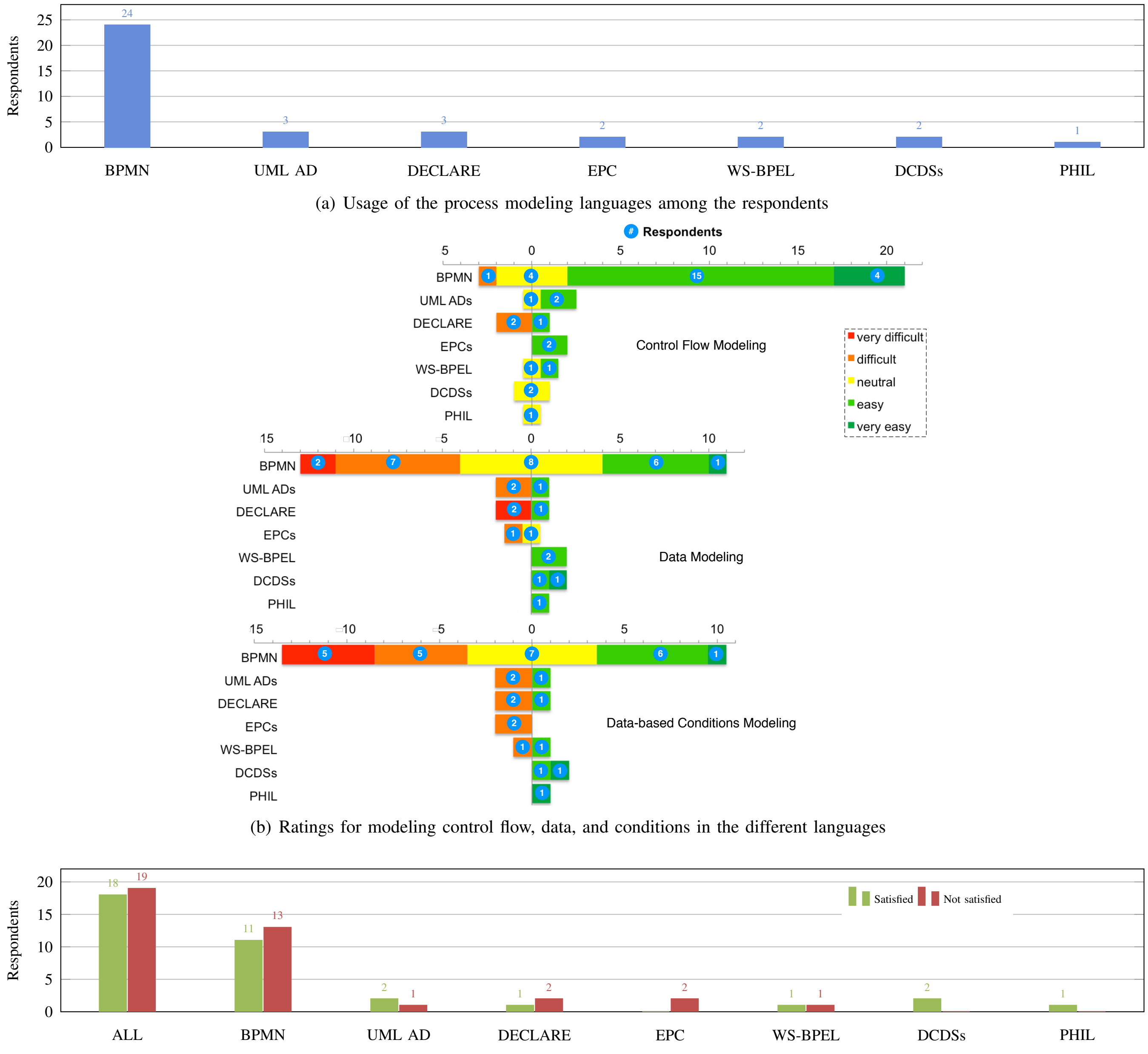
In the following we present the main results of the analysis performed on the data collected through the questionnaire. While we recognize that the available dataset is relatively small, thus limiting the ability to draw statistically significant conclusions and generalize the results beyond the sample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

Participant Profiles. With the help of the general questions concerning the participant's role and expertise, we were able to identify profiles fitting to the respondents. 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. The declared *years of expertise* in BPM is concentrated in the range of 5–10 years for most respondents (25 respondents, 67%). Of the 26 academic researchers, 18 stated an expertise of 5–10 years, 4 an expertise of less than 5 years, and the remaining 4 an expertise of 11–15 years. Only 1 respondent, an industrial researcher, stated an expertise of more than 15 years in the BPM field.

Process Modeling Languages. Fig. 1(a) shows that BPMN is by far the most used modeling language among the respondents (65%). This suggests an increasing adoption of the BPMN standard in different domains, including both research activities and industrial projects. 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 easily be 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 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 the highly experienced industrial researcher. Although foreseeable, it is interesting to note that among our respondents the adoption of activity-centric imperative languages (31 total respondents), is significantly higher than the usage of declarative and data-centric languages (6 total respondents). No respondent stated to use artifact-centric languages or CMMN, as well as no one stated to use YAWL/Workflow nets.

Modeling Tools. Participants were asked to complete their profile by specifying the modeling tools they have worked with, the target application domains, and the main purpose of their process modeling efforts. Regarding process modeling tool support, survey participants mentioned more than 40 different tools in total, ranging from advanced BPM platforms to general purpose charting and diagramming tools. Among the most popular software in use, the Signavio Process Editor was mentioned by 20 respondents and dominates the list of top 5 tools, followed by Microsoft Visio (13 respondents) and the ARIS platform by Software AG (8 respondents). Also mentioned were IBM tools including Blueworks, WebSphere Lombardi/IBM Business Process Manager (8 respondents), as well as the the AristaFlow BPM Suite (5 respondents).

Application Domains. Considerable variability was also observed regarding the application domains, with more than 25 different answers. Reported application domains span multiple fields, ranging from research-oriented academic examples and use cases, to real-world scenarios involving healthcare providers, financial services and banking, manufacturing, retail and logistics, the public sector, and even energy companies. In particular, the healthcare domain was mentioned by at least 10 respondents, followed by the automotive and financial



(a) Usage of the process modeling languages among the respondents

(b) Ratings for modeling control flow, data, and conditions in the different languages

(c) Respondents' satisfaction and modeling languages

Fig. 1: Visual representation of the interviews' results

domains. When combining the application domains with the modeling languages and other findings reported before, we mainly observe a multi-domain applicability of BPMN. In addition, despite the low number of respondents using other languages, the characteristics of the languages are reflected in the reported application domains. 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), EPC users target the automotive domain, WS-BPEL users focus on Business-to-Business integration scenarios, and UML AD are mainly used in well established BPM application scenarios, including financial services, banking and administrative settings.

Purpose of Process Modeling. Regarding the intended purposes of process models, the participants' answers show that the created models are mainly used to address the classical stages of the process lifecycle. Most participants (76%) use process models for documentation purpose, while process enactment and optimization was mentioned by 49% and 43% of the respondents. Additionally, 8 respondents (22%) also target more advanced application areas, such as process simulation, process discovery, and process quality analysis and assurance.

Process Modeling Perspectives. Moving closer to our research objective, survey participants were asked to rate how easy they perceive the modeling and definition of control flow, process data, and data-based conditions to be 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 Fig. 1(b). Although the number of respondents per language varies greatly, participants' ratings are insightful but not surprising. 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 Fig. 1(b)), 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), especially in comparison with the positive ratings for control flow modeling simplicity. Given the number of respondents, this is particularly evident for the BPMN language. While only 1 out of 24 BPMN users rates control flow modeling as *difficult*, the number of respondents that consider data modeling as *difficult* increases to 7, and 2 respondents even rate it as *very difficult*. One third (8 respondents) of BPMN users have a *neutral* perception of the simplicity of data modeling, whereas the remaining 7 respondents still rate it as *easy* (6 respondents) or *very easy* (1 respondent). Similar results can be observed for all languages regarding the modeling of data conditions (see the third chart in Fig. 1(b)). Again, the respondents using data-centric or object-aware languages (DCDSs and PHILharmonicFlows) rate the definition of data conditions as *easy* or *very easy* task. Similarly, the shift towards negative answers observed in activity-centric languages regarding data modeling is observable for data conditions as well. Particularly, regarding BPMN we observe an increased number of respondents considering data conditions as *very difficult* to express (5 respondents). One can note again the lack of a common perception of the simplicity of data conditions modeling. While 7 BPMN users out of 24 rate it as *neutral* and 7 provide positive answers (including *easy* and *very easy* ratings), a slight majority of BPMN users (10 respondents) provide negative answers (including *difficult* and *very difficult* ratings).

Kind of Data Objects. In order to understand how data elements and their possible relationships are perceived, participants were asked to specify the kind of data objects required in their process models. Participants' answers generally indicate that different types of data elements, with varying degrees of complexity, often coexist and relate to each other. 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 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. Participants' answers further clarify the 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. 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. Moreover, data objects are related to gates used in a process to define *milestones* whose achievement guarantees for certain quality properties of the process, expressed as conditions on data objects. However, as these conditions 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 (e.g., customer and product data, patient data, medical orders, etc.), *(ii)* defining I/O elements for individual tasks, *(iii)* expressing split conditions, and *(iv)* 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. According to this approach, business information is regarded as a first class concept within the overall enterprise architecture, whereas process models provide a process-oriented view focused on the activity flow. 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. A comprehensive, well-articulated definition of the role of data objects is provided by an academic researcher using BPMN. She relates data elements to the definition of *pre- and postconditions* of activities. Preconditions must be met to enable the execution of an activity, and postconditions refer to results of activity execution. In addition, at every point in time each data object has a specific *state* (defined by the current values of its attributes) that describes a specific business situation of interest. Data objects then evolve as a result of process activities that act on them and change their state. The prominent and complex role of data objects also emerges from the comments of the respondents using data-centric or object-aware languages. While 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 and Data Awareness. With the aim of introducing an additional analysis dimension, we focused on participants' satisfaction in respect to the data modeling capabilities provided by the language they stated to use primarily. As reported in Fig. 1(c), respondents' satisfaction reflects their perception of the simplicity of data modeling discussed before. 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. Focusing again on the 24 BPMN respondents, they are almost equally partitioned into *satisfied* and *unsatisfied* users. This suggests that there is no common perception of the quality of data modeling capabilities provided by the language, resulting in the need for further investigation. Most of the subsequent result analysis has then been performed in terms of two groups of participants, i.e., *satisfied* (49%) and *unsatisfied* (51%) respondents, further partitioned according to the modeling language they use.

Satisfied Respondents. The definitions of *data awareness* provided by the participants being satisfied with the data modeling capabilities of the used activity-centric language generally indicate a common perception of the subject. Data awareness 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 in a process modeling language. Data awareness is thus related to the degree to which data involved in business processes is explicitly modeled and easy visible. Some respondents further clarify that the awareness of what data is processed also has to cover: *(i)* data evolution (in terms of the states the data elements go through), to correlate the final output of a process with the provided inputs; and *(ii)* the data elements needed to make (business) decisions in the process flow. In addition, one of the respondents refers to data awareness as the possibility to change the metamodel of the used data over time, with the ability to automatically propagate the changes to the process model. 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. DCDSs users obviously motivate their claim that data awareness is already present in the modeling language by stating that the language itself was specifically designed with data awareness as a primary objective, so as to capture the essential features of data-awareness. However, as an important remark, they stress that the DCDS framework is not meant as a modeling language targeted to end users, but rather as a well-founded "internal representation format". In this respect, DCDSs have the same advantages and limitations of rule-based approaches: any control-flow construct can be modeled, although not always in a straightforward way. 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, focusing on the capability of letting process participants access process data at any point in time during process execution. Data awareness is considered part of the framework, as processes are defined in terms of *object behaviors* and *object interactions*. Process execution is data-driven, i.e., process progression does not depend on activity completion states, but solely on data availability.

Unsatisfied Respondents. 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. Some of the definitions provided by BPMN and EPC users clearly indicate that the ability to effectively integrate data objects into process models requires to: *(i)* explicitly treat data as a *first class citizen* in the modeling language; *(ii)* extend the notion of *process state* to also cover data objects; and *(iii)* properly define or extend process execution *semantics* by considering data objects' behavior and lifecycle.

When asked to specify what they think is missing in the modeling languages, a majority of the unsatisfied respondents (10 out of 19) simply focused on features for modeling process relevant data. This is the case of Declare users, EPC users and 7 of the 13 unsatisfied BPMN users. In addition, UML AD, WS-BPEL users and 2 BPMN users agree on the lack of support for expressing conditions over data elements, while the possibility of synchronizing object behavior (and object interactions) with process execution is considered as the main missing feature only by some of the BPMN users. Unsatisfied respondents provide several arguments to motivate their position on the lack of data awareness in the modeling languages. In detail, EPC users highlight that the language does not provide support for the definition of *stateful* data objects and does not allow to explicitly define data-based rules to be used for expressing branching conditions or for specifying data-dependent task assignment policies. To achieve this in practice, the designer is forced to introduce in the models many *informal annotations*, which are often disregarded when the models have to be concretely implemented or when they are archived. Regarding BPMN, several shortcomings emerge from the sometimes contrasting participants' comments. On the one hand, one of the respondents recognizes that BPMN was designed as a control flow oriented process modeling language and thus data modeling was not a primary concern in the standard. On the other hand, a different respondent complains that "data handling is simply left undefined in the standard, with only some vague reference to XML". Most participants focus on the limited expressiveness of the language regarding the *semantics* of data objects, specifically in terms of *(i)* dependencies between data objects; *(ii)* split conditions and task enactment preconditions; and *(iii)* the reuse of data objects, e.g., in multiple tasks. In general, the available options for modeling data objects and working with them (also in existing tools) are considered as "extremely limited". According to one of the respondents, the support for the specification of data objects and I/O parameters is even "rudimentary", and often results in cluttered process models with no added value for the designer. In her view, this also prevents to analyze the data flow to identify hidden interdependencies between different activities in a process. Some of the respondents claim that BPMN *does* support the modeling of stateful data elements and data flow relations between activities, but recognize that these elements are to be aligned with the modeled control flow (i.e., the data perspective is *subordinate* to the control flow), concluding that "data is only a kind of add-on to the models".

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 stress again the need to define (or improve) the expressiveness of *data objects semantics*. According to them, this requires the definition of standard *object-oriented abstractions* and specific data object symbols with defined behavior, to represent, classify, instantiate, specialize, compose and reuse data objects.

Possible correlations and interdependencies between data and processes should be made explicit both at the modeling stage (e.g., enabling the creation of a separate data model to be linked with the process model) and at runtime, when process instances and data object instances coexist and interact with each other. One of the respondents explicitly mentions the need for an integrated *methodology* for aligning the states and lifecycles of business data objects with the process logic defined by the control flow. Participants' comments suggest that this should help the designer in better integrating the different perspectives (including roles/actors) and in understanding how objects and object relationships influence process execution and vice versa. On the same line, additional required features include the explicit handling of data for decision making and proper support for analyzing the interdependencies between activities based on the data produced and consumed. Existing connections and dependencies should also be taken into account at a lower level, allowing to *(i)* "trace, link and realize data objects from BPMN to other specialized data modeling languages such as UML"; and *(ii)* connect the modeled data objects with the "real data" (e.g., exploiting ORM techniques). While this is technically achievable, some respondents point out that this is not part of the actual BPMN specification and ask for an explicit definition of mandatory languages (e.g., XML, JSON) for defining data. Finally, from a visualization perspective, some respondents ask for an *integrated* view of processes and data, while others would prefer a *separate* view for the data perspective and suggest the introduction of a "data flow based process visualization" perspective.

V. Discussion and Future Work

Providing a reference definition of "data awareness" is not easy. As confirmed by the results, 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 [14]. 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 [15] this phenomenon is also denoted as impedance mismatch between process layer on one hand and business logic and data layers on the other.

In general, the creation of data objects during runtime as well as evolving data object states have a direct impact on the running processes and vice versa. 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. In the survey, process designers and engineers familiar with activity-centric languages indicate that this requires (1) to explicitly treat data as a first class citizen in the respective process modeling language, (2) to extend the notion of process state to cover data objects and their states, and (3) to properly define or extend process execution semantics by considering the behavior and lifecycle of data objects as well as the semantical relations between them. Note that these properties are exactly what data- and object-aware process support paradigms claim to provide.

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. Exactly this concern has been one of the drivers leading to the design of data-driven, artifact-centric and object-aware process modeling approaches. In particular, to some extent, these approaches were designed "from scratch" rather than by extending or complementing existing languages. As discussed by [16], 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; i.e., the features enjoyed by the users of data- and object-aware process modeling approaches are largely missing in activity-centric approaches, whose users are therefore often dissatisfied with the way data is handled in respective tools. This is somehow obvious as the various approaches targeting at data- and object-aware process support have been designed with the goal to overcome the limitations of existing activity-centric 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-centrice modeling tools. In this context, it would be interesting and insightful to systematically analyze and compare existing data- and object-aware process management technology in order to better understand current state-of-the-art as well as the features still missing.

Finally, the presented results have to be viewed in light of several limitations. First, the number of respondents (i.e. 37) is relatively low, hampering result generalization. However, it is noteworthy that the response rate of 54% (37 of the 68 invited subjects participated in the survey) is considerable. Moreover, to the best of our knowledge, in the BPM field the typical number of users participating in similar empirical studies has not been so far very high, e.g., in [10] only 10 practitioners were used. A second limitation is that, even though 10 of the 37 respondents are from industry, the majority of survey participants indicated academic background, limiting result generalization as well.

The presented empirical investigation is part of a larger one on data-aware process support approaches. Based on the 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-ware business process support. Additionally, with the help of the SLR, we shall devise a framework for evaluating and comparing respective approaches. Finally, recognizing the need for a fully object-aware business process management system we address a reference implementation of such a system.

Acknowledgements. This work has been partly supported by the EU Project VOICE, the Italian cluster SM&ST (Social Museum and Smart Tourism) and the Sapienza award SPIR-ITLETS.

References

[1] A. Russo and M. Mecella, "On the evolution of process–oriented approaches for healthcare workflows," *Int J Business Process Integration and Management*, vol. 6, no. 3, pp. 224–246, 2013.

[2] A. Marrella, A. Russo and M. Mecella, Planlets: "Automatically Recovering Dynamic Processes in YAWL," *20th Int. Conf. on Cooperative Information Systems (CoopIS)*, 2012.

[3] C. Di Ciccio, A. Marrella, and A. Russo, "Knowledge-Intensive Processes: Characteristics, Requirements and Analysis of Contemporary Approaches," *Journal on Data Semantics, vol. 4, no. 1, pp. 1–29*, 2014.

[4] M. Pesic, H. Schonenberg, and van der Aalst, W.M.P., "DECLARE: Full Support for Loosely-Structured Processes," in *Proc. EDOC'07.*

[5] R. Hull, "Artifact-Centric Business Process Models: Brief Survey of Research Results and Challenges," in *Proc. OTM'2008.*

[6] D. Cohn and R. Hull, "Business artifacts: A data-centric approach to modeling business operations and processes," *IEEE Data Eng Bulletin*, vol. 32, no. 3, pp. 3–9, 2009.

[7] R. Hull *et al.*, "Introducing the Guard-Stage-Milestone Approach for Specifying Business Entity Lifecycles," in *Proc. WS-FM 2011.*

[8] OMG, "Case Management Model and Notation, Version 1.0," 2014.

[9] H. A. Reijers, Rigter, J. H. M., and van der Aalst, W. M. P., "The Case Handling Case," *Int J Coop Inf Sys*, vol. 12, no. 03, pp. 365–391, 2003.

[10] H. A. Reijers, T. Slaats, C. Stahl, "Declarative Modeling - An Academic Dream or the Future for BPM?", in *Proc. BPM 2012.*

[11] B. Bagheri Hariri, D. Calvanese, G. De Giacomo, A. Deutsch, and M. Montali, "Verification of Relational Data-centric Dynamic Systems with External Services," in *Proc. PODS'13.*

[12] A. Russo, M. Mecella, M. Montali, and F. Patrizi, "Towards a Reference Implementation for Data Centric Dynamic Systems," in *Proc. BPM Workshops 2014.*

[13] V. Künzle and M. Reichert, "Towards Object-Aware Process Management Systems: Issues, Challenges, Benefits," in *Proc. BPMDS'09.*

[14] B. Weber, B. Mutschler, and M. Reichert, "Investigating the effort of using business process management technology: Results from a controlled experiment," *Sci Comp Program*, vol. 75, no. 5, 2010.

[15] M. Dumas, "On the convergence of data and process engineering," in *Proc. ADBIS'11.*

[16] J. C. Recker, "Opportunities and constraints : the current struggle with bpmn," *Business Process Management Journal*, vol. 16, no. 1, 2010.