Software & Systems Modeling (2019) 18:2679–2716
https://doi.org/10.1007/s10270-018-0695-0

REGULAR PAPER

DALEC: a framework for the systematic evaluation of data-centric approaches to process management software

Sebastian Steinau[2] · Andrea Marrella[1] · Kevin Andrews[2] · Francesco Leotta[1] · Massimo Mecella[1] · Manfred Reichert[2]

Received: 20 July 2017 / Revised: 23 August 2018 / Accepted: 24 August 2018 / Published online: 8 September 2018
© The Author(s) 2018

Abstract

The increasing importance of data in business processes has led to the emergence of data-centric business process management, which deviates from the widely used activity-centric paradigm. Data-centric approaches set their focus on data, aiming at supporting data-intensive business processes and increased process flexibility. The objective of this article is to gain profound insights into the maturity of different data-centric approaches as well as their capabilities. In particular, this article will provide a framework for systematically evaluating and comparing data-centric approaches, with regard to the phases of the business process lifecycle. To this end, a systematic literature review (SLR) was conducted with the goal of evaluating the capabilities of data-centric process management approaches. The SLR comprises 38 primary studies which were thoroughly analyzed. The studies were categorized into different approaches, whose capabilities were thoroughly assessed. Special focus was put on the tooling and software of the approaches. The article provides the empirically grounded DALEC framework to evaluate and compare data-centric approaches. Furthermore, the results of the SLR offer insights into existing data-centric approaches and their capabilities. Data-centric approaches promise better support of loosely structured and data-intensive business processes, which may not be adequately represented by activity-centric paradigms.

Keywords Systematic literature review · Data-centric BPM · DALEC framework · Systematic evaluation

1 Introduction

Over the last decade, organizations and companies have started adopting process management methodologies and tools, with the aim of increasing the level of automation support for their operational business processes. Business Process Management (BPM) has therefore become one of the leading research areas in the broader field of information systems [73].

In the BPM research area, various languages, techniques, methodologies, paradigms, and environments have been proposed for modeling, analyzing, executing, and evolving business processes [62]. Furthermore, a new generation of information systems, known as Process Management Systems (PrMSs), has emerged. A PrMS is a system created to support the management and execution of business processes involving humans, applications, and external sources of information. The general characteristic of PrMSs is that process logic is not hard-coded, but explicitly expressed in terms of *process models* [20]. Particularly, process models constitute the major artifact enabling comprehensive process

Communicated by Dr. Manuel Wimmer.

✉ Andrea Marrella
marrella@diag.uniroma1.it

Sebastian Steinau
sebastian.steinau@uni-ulm.de

Kevin Andrews
kevin.andrews@uni-ulm.de

Francesco Leotta
leotta@diag.uniroma1.it

Massimo Mecella
mecella@diag.uniroma1.it

Manfred Reichert
manfred.reichert@uni-ulm.de

[1] Dipartimento di Ingegneria Informatica Automatica Gestionale, Sapienza Università di Roma, via Ariosto 25, 00185 Rome, Italy

[2] Institute of Databases and Information Systems, Ulm University, Building O27 Level 5, James-Franck-Ring, 89081 Ulm, Germany

support, as they provide an executable representation of the business process.

So far, PrMS usage has not been as widespread as expected by software vendors [28]. Although some software systems already integrate specific process engines or components, no generic paradigm exists that is capable of *fully* supporting *all* processes that can be found in contemporary application software [39]. Most PrMSs require many workarounds and proprietary implementations to support all processes of a company. A major reason for this is the *lack of integration between business processes and business data*, which can be explained by the fact that traditional PrMSs follow the principle of separating concerns. This means that business data, business processes, and business functions are managed by different kinds of systems. As a consequence, *traditional PrMSs are unable to provide integrated access to business data*.

The role of data in major process modeling languages is evaluated in [49] by comparing the data modeling capabilities and the level of data-awareness in respect to these languages. The evaluation confirms that the *general level of data support is low*. While in most cases the representation of data is supported, data manipulation by process activities is often under-specified or completely abstracted away. Furthermore, neither the relationships between data nor the role these relationships have in the context of a process is considered.

1.1 Running example: study plan process

To support the above claims, this section introduces a running example that describes the procedure for managing the application, review and acceptance of study plans submitted by MSc students at Sapienza University of Rome. We use this process as a running example throughout the article.

Example 1 (Study plan process)
After the enrollment into the two-year program for the MSc in Engineering in Computer Science, students must prepare and submit a study plan indicating the university courses (and associated exams) they wish to attend. The review and approval of the study plans is performed by a commission, which usually includes one or more professors appointed by the University.
The preparation, submission, review, and acceptance of a study plan is managed through a dedicated web application system. Before students may submit their study plan for approval, they must log into the system with their sensitive information, i.e., university ID and password. Furthermore, they need to specify their personal information, i.e., name, surname, e-mail address, birthday, and residence. The personal information is only required the first time student accesses the system. Finally, students must decide in which exams they want to participate during the course of their studies.
To update an already approved study plan, students must directly contact a member of the commission to request permission for updating their existing study plan. In addition to the request, students must provide details about the study plan items that shall be modified. At this point, the commission member may decide to approve or reject the student's update request. In case of a positive decision, the commission member will delete the existing study plan and notify the student requesting the update. This way, the student may now prepare and submit an updated study plan.
The submission of a new study plan feeds a database that may be accessed individually by any member of the commission at any time. Note that the start of the reviewing process does not depend by the submission/update of a study plan, but it is performed occasionally by a commission member, without any specific rule that steers its enactment. Specifically, each commission member uses the system to retrieve the set of study plans that are waiting for approval and to review a subset of them from a scientific and technical point of view. If a study plan is deemed eligible, it will be immediately approved and a notification is sent to the student who submitted it. However, if the commission member proposes its rejection, a reason must be provided as part of the rejection notification sent to the student.

Figure 1 depicts the study plan process represented in the Business Process Model and Notation (BPMN). Note that BPMN has been chosen to visualize the "Study Plan" running example as the notation is understandable by non-domain experts. Further, it allows to explicitly identify which business data are required to properly execute a process. BPMN provides two kinds of business data, namely data objects and data stores. Data objects are used to model local information (e.g., documents, files, material) flowing in and out of activities. Data stores represent places containing data objects that need to persist beyond the duration of a process instance. Process activities can extract/store data objects from/in data stores.

If, on the one hand, modeling business objects in BPMN may help the reader to identify the flow of information in the process, on the other hand the price to pay is an increased complexity of the model in terms of readability and understandability. The latter derives by the fact that BPMN does not provide a well-formalized semantics for the business objects, making their use in the process model highly ambiguous [37].

In addition, as extensively investigated in [47], the main issue is that data objects in activity-centric notations, i.e., BPMN, are under-specified. BPMN places no restrictions or recommendations on data objects. Process modelers must

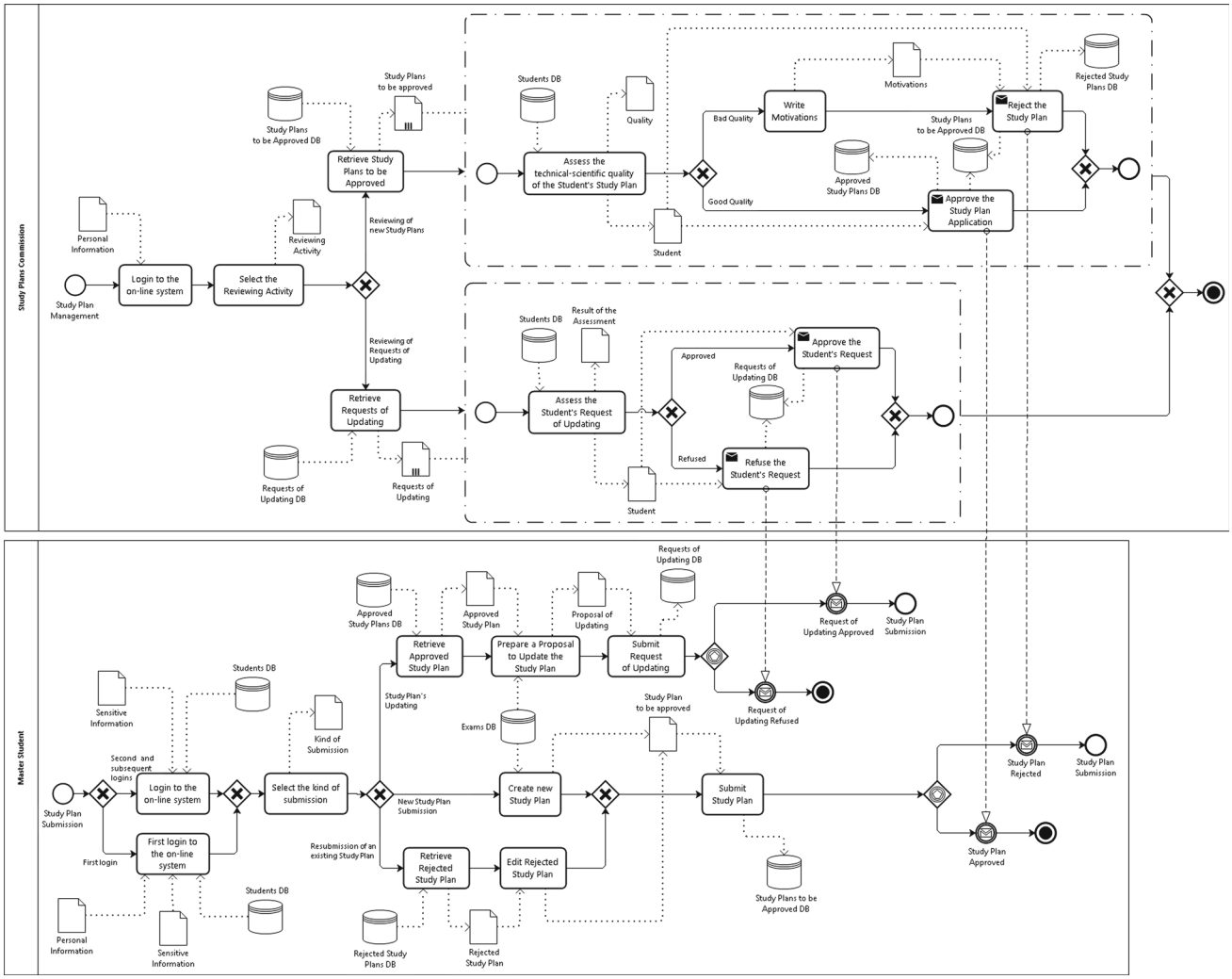


Fig. 1 The study plan management process represented as a BPMN process model

choose their level of expressiveness of data objects. Therefore, standard data types, e.g., string, integer, boolean, and files, are prevalent. When structured data are actually needed by the modeler, the choice is completely arbitrary how to represent such structured data. A modeler may choose any formal notation or no formal notation at all. This creates a high ambiguity and fluctuation between models, making them difficult to compare and interpret. In any case, process and data remain separate.

1.2 Problem statement

The process described in Example 1 can be used to showcase the shortcomings of some process modeling approaches, as the process participants often need access not only to process information, but also to business data, in order to complete their tasks. However, such an integrated view on data and processes is lacking in the BPMN model of the running example: a student is allowed to create or update study plans, but the process model does not show how the data structures for the study plans and their attributes may be accessed and edited. Note that without such an integrated view, relevant context information might be missing during process execution. Moreover, when making a decision on a particular study plan application, the commission member has no access to other applications.

In contrast to database management systems, current PrMSs are not broadly used for implementing application systems. This originates from the common activity-centric paradigm used by many PrMSs. The activity-centric paradigm has several limitations when not being used for the support of highly structured, repetitive business processes. This means that PrMSs enforce a particular work practice and predefined activity sequences, which leads to a lack of flexibility during process execution [62].

However, many of the processes that can be found in real-world scenarios, such as the one from Example 1, are often characterized as unstructured or semi-structured. In addition, they are considered as being knowledge-intensive and driven by user decisions [18]. This means that work practice may vary between users. Thus, different activity sequences need to be supported. For example, while one commission member may work on only one study plan at the same time, another member may want to approve or reject several study plans in one go. This requires increased flexibility during process execution, which is usually not provided by the activity-centric paradigm.

When executing processes in real-world scenarios, typically, business data are represented through *data objects*. Each data object comprises a number of object *attributes* that are created, modified, and deleted during process execution. In this context, user tasks, typically executed through user forms, play a crucial role. Such forms are indispensable for assigning or changing attribute values. However, which input fields shall be displayed within a particular user form not only depends on the user executing an activity, but also on the progress of the respective process instance.

Example 2 (Data in business processes)
Students must provide their personal information and choose which exams they want to attend before they may submit their study plans. However, if a student has already submitted the study plan for approval, he or she may no longer change the values of the information provided. Note that this requires a multitude of user forms, the implementation of which is a cumbersome and costly task.

Hence, Example 2 shows that the activity-centric paradigm is not particularly well suited for managing business data.

Finally, we notice that many data objects of different types are processed during the execution of a process instance. In this context, the processing of one data object may depend on the processing state of other data objects.

Example 3 (Process dependencies)
If a student has already submitted a study plan, a new study plan that replaces an existing one may only be prepared and submitted if a commission member gives her/his approval and removes the existing study plan from the database.

Moreover, individual data objects may be in different processing states at a given point in time. Several study plans might be under review concurrently. While the review of a particular study plan might have just been initiated, others might have already been approved or rejected. These aspects are ignored by most implementations of the activity-centric paradigm.

1.3 Contribution

It has been acknowledged by various authors that many of the limitations of contemporary PrMSs can be traced back to the missing integration of processes and data [19,49,60, 62]. To tackle the issue of integrating data and processes, *data-centric approaches* have emerged. They adopt a fundamentally different view on process management, where data objects are considered as "first-class citizens" and as main drivers for process modeling and execution. Data-centric approaches aim at providing a complete integration of the process and data perspectives. Therefore, they rely on design methodologies in which the identification and definition of process activities are induced by the specification of a data model [6,12].

Until now, however, a general understanding of the inherent relationships that exist between processes and data is still missing. Whereas many data-centric approaches solely focus on modeling aspects (i.e., the design phase), only few approaches take the entire business process lifecycle, comprising implementation, execution, diagnosis, and optimization, into account. In a nutshell, there is a lack of profound methods and comprehensive frameworks for systematically assessing, analyzing, and comparing existing data-centric approaches. In this paper, we aim at filling this gap through a twofold contribution:

1. We present results from a systematic literature review (SLR) of data-centric process management approaches. Besides elaborating the state of the art, we systematically analyze existing data-centric approaches regarding their ability to cope with the limitations of traditional (i.e., activity-centric) process management approaches. Based on this evaluation, we discuss the strengths and weaknesses of each approach.
2. Based on the empirical evidence and the results provided by the SLR, we derive the *Data-centric Approach Lightweight Evaluation and Comparison (DALEC)* framework. The framework may be used for evaluating, categorizing and comparing data-centric approaches in each stage of the business process lifecycle.

The results obtained by the application of the framework reveal that the field of data-centric process management is still in an early development stage, as it lacks consolidation and strong tool support. In this direction, we consider the framework as beneficial for broadening the use of data-centric process management as it allows for the systematic evaluation and comparison of data-centric approaches.

The remainder of the paper is organized as follows. Section 2 provides an overview of the main modeling approaches and introduces the business process lifecycle and its related PrMS support. Section 3 explains the research methodology

applied during the literature review. The results of the SLR are presented in Sect. 5 while Sect. 4 highlights possible limitations and discusses threats to validity of this work. Then, the comparison framework for data-centric approaches is introduced in Sect. 6, whereas Sect. 7 shows the application of the framework to a selection of data-centric approaches identified in the SLR. Section 8 examines similar literature reviews in the BPM research field. Finally, to conclude our paper, Sect. 9 comprises a discussion of our results and Sect. 10 contains a summary and an outlook.

2 Background

In this section, we present the relevant background to understand the paper. Specifically, in Sect. 2.1, we first provide an overview of the existing modeling approaches to process management. Then, in Sect. 2.2, we discuss the various steps of the process lifecycle and the related PrMS support.

2.1 Overview of main process modeling approaches

Traditional notations for business process modeling are *imperative* and *activity-centric*, i.e., a process is composed of activities representing units of work. The order of the activities, in turn, is described by *control flow*. Common patterns of control flow include sequences, loops, and parallel as well as alternative branches. Examples of graphical activity-centric modeling notations include the Business Process Model and Notation (BPMN), Event-driven Process Chains (EPC), and UML Activity Diagrams (UML AD). Especially, BPMN has been widely adopted in current practice and can be considered as the de-facto standard for business process modeling.

As an alternative to the imperative modeling notations, activity-centric processes may also be defined in a declarative fashion with notations such as Declare [57], which allows defining constraints to restrict the choice or ordering of activities for a more flexible process execution compared to imperative approaches.

Activity-centric approaches, in particular BPMN, support the modeling of data in terms of abstract data objects, which may be written and read by activities. Structured data, i.e., logically grouped data values, are not considered. In addition, data objects are often omitted or under-specified to reduce the complexity of the process model. According to [19], this leads to an "impedance mismatch" problem between the process and the data perspectives.

As an alternative to the activity-centric process modeling paradigm, processes may be specified according to a *data-centric* modeling paradigm.

In data-centric modeling approaches, the process model definition (and, hence, the progress of a process) is based on the availability and values of data rather than on the completion of activities.

One of the first approaches that has dealt with data-centric process management is *Case Handling* [75]. In this approach, a case contains all the necessary information to achieve a business goal.

Activities do not have a pre-specified order, but become enabled when required data becomes available, i.e., data objects are filled by activities and allow other activities to become enabled. Therefore, the existence of data, i.e., information within data objects, drives process execution instead of the completion of activities (i.e., control flow as in activity-centric approaches).

Artifact-centric process models [33] constitute a specific form of data-centric process models. An artifact-centric process model encapsulates data and process logic into artifacts.

Artifacts consists of an information model holding the data and a lifecycle model describing the changes to the information model.

An artifact, in turn, consists of an information model, holding relevant data, as well as a lifecycle model that 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 declarative Guard-Stage-Milestone (GSM) meta model [34].

The Guard-Stage-Milestone meta model substantially influenced the *Case Management Model and Notation (CMMN) standard* [55]—the recently standardized notation for case management as proposed by OMG. In this context, case management 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 [63].

The framework of relational *Data-centric Dynamic Systems (DCDSs)* was originally proposed for the formal specification and verification of artifact-centric processes [4].

Since then, it has developed into a full process modeling approach capturing the connection and interplay between processes and data [65]. DCDSs use a declarative, rule-based process specification for capturing the formalization and progress of the data perspective.

PHILharmonicFlows [39] constitutes a framework for modeling, executing, and monitoring *object-aware business processes*.

The approach organizes data into structured objects.

Each object is associated with a lifecycle process describing how data is acquired.

A business goal is realized by the interactions of one or more objects, which requires sophisticated coordination.

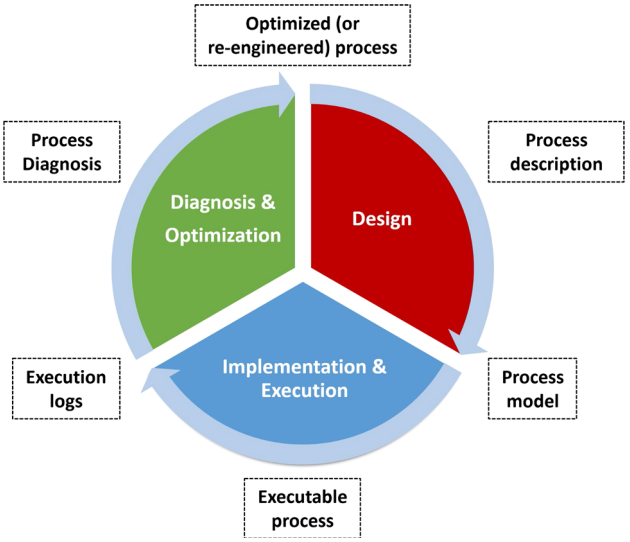


Fig. 2 The lifecycle of a business process

2.2 PrMSs and the business processes lifecycle

PrMSs emerged out of a demand for business processes to work with existing enterprise software applications as well as to benefit from automation as well. Traditional, manual methods for creating, enacting, and managing workflows (i.e., executable processes) became too cumbersome compared to the possibilities of digital technology. Early PrMSs provided only a basic activity list with a user interface to move work around the organization. Particularly, considerable customization efforts were required in order to integrate software applications. Current PrMSs, however, offer advanced capabilities for managing business processes, such as enhanced support for human collaboration, flexible activity execution [62], mobile access to processes [58], and analytic and real-time decision management. As such, PrMSs are now seen as the bridge between Information Technology (IT), business analysts, information system engineers, and end users, by offering process management features and tools in a way that provides benefits for both business users and engineers [20]. Finally, PrMSs hold the promise of facilitating the everyday operation of many enterprises and work environments, by supporting business processes in all phases of their lifecycle [20].

In BPM literature, there are many different definitions of a process lifecycle, e.g., [19,29,31,73,79]. We decided to adopt a slightly modified version of the process lifecycle as proposed by van der Aalst [73] due to its succinctness and relevance. As shown in Fig. 2, the business process lifecycle consists of three major phases: Design, Implementation & Execution, and Diagnosis & Optimization.

Design In the design phase, analyses of the business processes as well as of their organizational and technical environment are conducted. Based on these analyses, a process is identified and modeled using a suitable business process modeling language. The resulting process model must then be verified in order to eliminate process modeling errors that can lead to run-time problems such as deadlocks. The process model also needs to be validated to ensure that it fits the intended behavior.

Implementation & Execution As soon as a process model has been designed, verified, and validated, it can be implemented and executed in a PrMS. First, the process model is enhanced with technical information required for its execution on the PrMS. Then, the process model is configured according to the organizational environment of the enterprise, e.g., by including the interactions of the employees and the integration with existing software systems. Once the process model has been configured, it is deployed on the PrMS. A deployed model can be instantiated to obtain an executable process instance. The PrMS actively controls the execution of process instances, i.e., process activities are performed according to the constraints (e.g., control flow) specified by the process model. In general, PrMSs enable real-time monitoring of running process instances. Furthermore, PrMSs log all events related to process execution, e.g., the start and end of an activity, writing of data values, or the occurence of errors during process execution. These execution logs can, in turn, be used in the Diagnosis & Optimization phase to derive process improvements.

Diagnosis & Optimization In this phase, event logs are evaluated based on business activity monitoring (BAM) and process mining techniques. Both aim at identifying problems that occurred during the enactment of the process instances. For example, BAM might detect that a certain activity always takes longer to complete than expected. This information, in turn, can be used to identify the causes and remedy them. Process mining, in turn, analyses the event logs of process instances, allowing for the detection and correction of process model errors as well as for the improvement of the process models. Furthermore, process mining is used to verify that process instances are compliant with the process model from which they have been derived, or to automatically construct process models from event logs. The information gained from analyzing process event logs may subsequently be used to improve and optimize the original process model. In this context, the term *schema evolution* describes the adaptation and improvement of existing process models [78]. Of particular interest in regard to schema evolution is the migration of the running instances to the evolved process model [61].

3 Methodology

A systematic literature review (SLR) was conducted with the goal of analyzing different data-centric approaches to process management. An SLR is a method to identify, evaluate, and interpret relevant scientific works with respect to a specific topic. We designed a protocol for conducting the SLR that follows the guidelines and policies presented by Kitchenham in [36] in order to ensure that the results are replicable and the means of knowledge acquisition are both scientific and transparent. Additionally, the probability of any bias occurring during the SLR is reduced [36].

The necessary steps to guarantee compliance with the SLR guidelines include the formulation of the research questions (cf. Sect. 3.1), the composition of the search string (cf. Sect. 3.2), the selection of the data sources on which the search is performed (cf. Sect. 3.3), the identification of inclusion and exclusion criteria (cf. Sect. 3.4), the questions regarding quality assessment (cf. Sect. 3.5), the study selection (cf. Sect. 3.6), the method of extracting data from the studies, and the analysis of the data (cf. Sect. 3.7).

3.1 Research questions

One goal of the SLR is to identify approaches that define data-centric processes or extend the existing approaches with better support for data. The first step when conducting an SLR is the formulation of research questions [36], which poses a particular challenge. Previously conducted research concerning data-centric approaches shows that different approaches use very different means to specify data and processes. The data-centric approaches known to us before conducting the SLR use objects with lifecycles, Petri nets in the colored and non-colored variant, and declarative descriptions. As opposed to objects with lifecycles, there are approaches where processes use structured data similarly to the way data objects in BPMN are used. However, the data-centric approaches unknown to us prior to conducting the SLR might have been entirely different from known approaches, employing known techniques differently or utilizing entirely new concepts and languages for defining data-centric processes.

In regard to the formulation of the research questions, this heterogeneity must be accounted for. It is therefore mandatory to find terms for different concepts that do not exclude potential data-centric approaches based on the phrasing of the research questions. In order to account for the heterogeneity of the different representations of data in different data-centric approaches, we define the term *data representation construct (DRC)*.

Definition 1 *(Data Representation Construct)* A Data Representation Construct is a general term for any form of structured data.

Common established examples of DRCs are artifacts in artifact-centric process management and objects in object-aware process management. Another relevant concept for data-centric approaches is *behavior*.

Definition 2 *(Behavior)* Behavior describes the means by which an approach acquires data values for its data representation constructs or to perform other activities.

For example, behavior refers to the lifecycle process of a DRC in artifact-centric process management. For approaches without a DRC lifecycle, behavior refers to the process that provides data values to the associated DRCs. For example, in an activity-centric process, activities and control flow are considered as behavior.

A single DRC with its lifecycle usually does not constitute a meaningful business process. Therefore, different DRCs or processes, depending on the approach, need to collaborate. As this requires DRCs to interact with one another, an *interaction* concept must be described by the respective data-centric approach.

Definition 3 *(Interactions)* Interactions describe the means by which the DRCs or processes of an approach communicate with each other.

For instance, in the artifact-centric paradigm for process management, the individual artifacts interact with each other at predefined points in their lifecycles by accessing information present in other artifacts. To facilitate such access, the artifact-centric approach offers an expression framework. Approaches that do not utilize DRC lifecycles may employ other techniques, such as messages.

As the terms DRC, behavior, and interactions are intentionally designed to cover a wide variety of different concepts, a certain level of uncertainty remains with respect to the formulation of research questions. However, this uncertainty cannot be eliminated entirely. Approaches may have several concepts that fit the definition of either a DRC, behavior, or interactions. As there is no obvious solution, ambiguities in the interpretation of an approach were discussed by the authors and resolved by majority vote. Consequently, other researchers might come to different conclusions regarding the answers to the research questions.

Based on these considerations, we formulated the following research questions, which will be discussed in the following:

- **RQ1:** What constructs are used to represent data? How are they defined?
- **RQ2:** How is behavior represented?

- **RQ3:** How are interactions represented?
- **RQ4:** Which mechanisms drive process execution? Is the execution data-driven?
- **RQ5:** How is process granularity managed?
- **RQ6:** Which parts of the process lifecycle are supported by tool implementations?

As research literature refers to various approaches for data-centric process management (cf. Sect. 2.1), where the data perspective is as important as the process perspective, we are interested in identifying what kind of constructs have been used to represent data of any complexity in such approaches (RQ1).

In addition, the SLR shall provide an overview of the way data may evolve during process progression, namely how the behavior of data is represented in data-centric approaches (RQ2), and investigate whether relations and interactions between DRCs (i.e., processes) play a role for process modeling and execution (RQ3).

A common feature of data-centric approaches is that the availability of data as well as data values (instead of the completion of activities) drives process execution. Therefore, the SLR shall create an in-depth understanding of the specific mechanisms used by data-centric approaches to execute processes (RQ4).

As illustrated in the study plan process (cf. Example 1.1), a process model may concern different granularity levels. Accordingly, the SLR shall provide insights about the way granularity is managed by existing data-centric approaches (RQ5).

Finally, in order to assess the practical applicability of existing data-centric approaches, the SLR shall further identify the available tools supporting these approaches along the different phases of the process lifecycle (RQ6).

In the following, we elaborate on the intentions behind the research questions and provide the necessary insights.

3.1.1 RQ1: What constructs are used to represent data? How are they defined?

RQ1 focuses on the analysis of the different types of data structures employed by data-centric approaches. Taking existing knowledge on data-centric approaches into account, we may assume that the majority stores data in a well-structured form, e.g., in terms of artifacts, objects, or tuples. Consequently, we introduced the concept of DRC (Data Representation Construct, cf. Definition 1) as an umbrella term for the various concepts for storing and representing data in a structured way.

3.1.2 RQ2: How is behavior represented?

RQ2 investigates how behavior is represented in the existing data-centric approaches. In general, DRC behavior (cf. Definition 2) is expressed through a lifecycle process, which describes the processing states of a single DRC, i.e., each DRC is characterized by its specific lifecycle process. If a DRC is not associated with a lifecycle process, behavior describes the means of data acquisition in general.

3.1.3 RQ3: How are interactions represented?

In general, a business process comprises multiple instances of the same DRC or different DRCs. Different processes, e.g., the lifecycle processes of DRCs, must collaborate to deliver a specific product or service. The interactions between the lifecycle processes, in turn, must be described and coordinated by the data-centric approach.

Regarding Example 1.1, the process for creating and submitting a study plan and the process for assessing a study plan need to interact with each other to reach the overall process goal, i.e., the approval of the study plan. In the following, we use DRC interactions (cf. Definition 3) as a shorthand term for denoting interaction between the lifecycles of the respective DRCs. For approaches without DRC lifecycle processes, denoted as non-lifecycle approaches, we consider the interactions between processes in general.

RQ3 focuses on the understanding of what types of interactions between DRCs with lifecycles or other behavior processes are supported by existing data-centric approaches and on how these interactions are represented.

3.1.4 RQ4: Which mechanisms drive process execution? Is the execution data-driven?

In data-centric approaches, the acquisition, manipulation, and evolution of data is the driving force for enacting business processes. While the term data-driven is most often intuitively understood, we did not find a suitable, formal definition. For research question RQ4, an execution mechanism of a process is considered as *data-driven* if Definition 4 is satisfied.

Definition 4 *(Data-driven)* In order to be considered as data-driven, all of the following criteria must be fulfilled:

1. The process has full visibility on all process-relevant data.
2. Interacting with data constitutes progress in process execution.
3. Any non-trivial process model must interact with process-relevant data at least once during process enactment.

According to the definition of the Workflow Management Coalition (WMC) [32], *process-relevant data* consists of decision information or parameters passed between activities or sub-processes. Conversely, *application data* are managed or accessed exclusively by the external applications interacting with a running process instance and are therefore not accessible to the PrMS.

In order to accomplish the first criterion, i.e., to make all process-relevant data fully visible to a business process, a straightforward solution would be to incorporate process-relevant data into the process model through the use of specific DRCs. The property of "full visibility" implies that the PrMS is aware of any manipulation over process-relevant data, even when made by an external application. Note that if some process-relevant data are not visible to the process or under the control of the PrMS, the execution mechanism of an approach is considered as "partially data-driven" at best.

The second criterion requires that the progress of an instance of a data-centric process depends on the availability of process-relevant data as well as their specific values at a given point in time. Consequently, the execution mechanism provided by a data-centric approach must be able to directly interact with process-relevant data, e.g., through standard operations (e.g., create, read, update, or delete). If interacting with data is not considered as relevant for progress in process execution (i.e., the first criterion would be sufficient for an approach to be considered as data-driven), the following problem arises: It would be possible to devise an approach that would be considered data-driven for the mere possibility of interacting with data, but all progress is achieved by some different means.

While criteria one and two provide a solid foundation for data-driven processes, an inconsistency still persists. A potentially data-driven process is not yet required to actually interact with data. According to the first and second criteria, a process that specifies no data and does not interact with data is considered as data-driven. To prevent this, the third criterion requires that a process instance interacts with process-relevant data at least once during its execution in order to be considered as data-driven. Process instances derived from trivial process models are exempt from this criterion. A trivial process model consists only of the bare necessities to create a syntactically correct process model, e.g., a process model solely consisting of start and end nodes, and which does not contain any activities. The exemption of trivial process models is desirable, as data-centric approaches might need to define trivial process models for special purposes, e.g., bootstrapping process modeling. If trivial process models were considered for the definition of data-driven, these trivial process models would prevent approaches from being classified as data-driven. This would be the case despite that they might fulfill all other criteria. Therefore, only process models of sufficient complexity (i.e., non-trivial process models) must handle data.

It needs to be emphasized that a data-driven execution is by no means necessary for a data-centric approach. Furthermore, from the fact that an execution mechanism is data-driven, it should not be concluded that it is superior to execution mechanisms not being data-driven.

3.1.5 RQ5: How is process granularity managed?

Process granularity represents the level of detail with which a process is modeled. For a process model to be executable, in general, the level of abstraction needs to be low enough to allow an engine to follow it step-by-step (i.e., a high level of detail). Furthermore, when coordinating different processes, varying granularity levels might create problems, e.g., when a process on a high abstraction level must be coordinated with a process on a low abstraction level. The abstraction used by programming languages over machine code can be considered as an analogy to process granularity.

The management of process granularity consists of choosing levels of granularity in order to achieve certain goals, most prominently the executability of the process models. Without intermediate transformations steps, in general, a process model requires a low level of granularity to be executable. With transformations, an abstract process model can be converted to an executable one. For example, BPMN process models can be converted to BPEL process models, i.e., to a language that was specifically designed to describe executable process models. Though managed process granularity has its benefits, trade-offs need to be considered, including decreased freedom in modeling and increased modeling efforts required to achieve the desired level of detail. With RQ5, we want to figure out whether data-centric approaches define levels of granularity, and which effects the approaches want to achieve.

3.1.6 RQ6: Which parts of the process lifecycle are supported by tool implementations?

The availability of tools for an approach supports its applicability and maturity. With RQ6, we look at the tool support of an approach for the different phases of the process lifecycle, for instance we check whether there is tool support for modeling or monitoring processes.

3.2 Search string

In order to perform a search over the selected data sources (cf. Sect. 3.3), we elaborated a search string by building combinations of keywords derived from our knowledge of the subject matter, e.g., "data-centric process." We put quotation marks around any combination to force the search engine

provided by the data sources to look for exact matches. In addition, we connected the combinations through the logical operator OR and we ensured that the terms "business" and "workflow" appeared in the search string. There are many fields and domains that involve data-centric processes, but do not relate to business process management. The final search string derived for the SLR is as follows:

> "data-aware process" OR "data-driven process" OR "data-oriented process" OR "data-centric process" OR "product-based process" OR "artifact-centric process" OR "artifact-based process" OR "knowledge-based process" OR "knowledge-driven process" OR "knowledge-intensive process" +workflow +business

The search string resulted from iteratively refining an initial set of search terms. The refinement was performed by conducting pilot searches to find a suitable set of search terms that maximizes the yield of different candidate studies. Search terms that yielded no additional studies were removed from the search string. Finally, the retrieved set of studies was continuously checked by subject matter experts in order to ensure that the set contained the studies known to be relevant for the SLR.

3.3 Data sources

During the refinement of the search string, we discovered that the search engines of the most popular scientific libraries had very different capabilities when specifying the search string. The examined libraries were SpringerLink, IEEE Xplore Digital Library, ACM Digital Library, Elsevier Science Direct, and Google Scholar. In summary, the limitations were so severe that the same search string could not be applied to all libraries, e.g., due to character limitations or non-supported Boolean operators. Circumvention techniques, e.g., splitting the search string into parts, had also proven to be unsuccessful, as different splits produced totally different results. Applying different search strings to each database is undesirable as it affects the consistency of the results as well as the replicability of the SLR. Therefore, we decided against such measures to ensure the integrity of the SLR methodology and the consistency of the data. In consequence, we initially decided to use only Google Scholar as our primary data source. Due to a character limit in the search window of Google Scholar, each search term was searched for separately (e.g., "artifact-centric process" +workflow +business). The individual results were merged to obtain the combined result of the entire search string. While Google products are known to personalize search results by reordering them, our search string was precise enough to allow us to examine *all* results, making their order of appearance irrelevant. Furthermore, Google Scholar has a coverage high enough to be used as a primary data source for a systematic review [7,24].

Nevertheless, we employed means to reduce the chance of missing a relevant study due to only using one source and to compensate for the limited amount of data sources. Therefore, an extensive *backward reference search* was performed by considering literature cited by the studies themselves (cf. Sect. 3.6). Additionally, to also obtain recently published relevant studies, studies that cited the already included relevant studies were evaluated as well. Furthermore, the backward reference search was not limited to Google Scholar. After the time we formally completed the SLR, in February 2017, it was discovered that the other libraries had expanded their search capabilities significantly. The search string could now be applied to the various data sources without adaptations. Therefore, we executed the search string on SpringerLink, IEEE Xplore Digital Library, ACM Digital Library, and Elsevier Science Direct to ensure that we had not biased our work by initially only relying on Google Scholar. We provide the raw results of our initial search as well as the results of the later searches in other libraries online [1].

Furthermore, the results of the additional searches were again evaluated by applying the inclusion and exclusion criteria and no new studies were discovered that were not already included in the SLR. The searches confirmed the validity of our original assumption, that the results from Google Scholar as well as the initial backward search would cover all relevant studies for the SLR.

3.4 Inclusion and exclusion criteria

In order to identify the relevant studies for the SLR, we defined the following inclusion and exclusion criteria.
Inclusion criteria:

1. Approach deals with data management in processes.
2. Approach defines and manages data-centric processes.
3. Extension to an existing data-centric approach.
4. Extension improving/detailing the concepts of already included approaches.

Exclusion criteria:

1. The study is not entirely written in English.
2. The study is not electronically available or access to the paper requires the payment of access fees[2].
3. The study is not peer-reviewed (e.g., an editorial or technical report).

[1] Raw search string results: https://bit.ly/2EZwG5b.

[2] This applies to access fees which are not already covered by the subscriptions from the Universities of Sapienza and Ulm.

4. The study merely mentions data in processes or data-centric processes as a related topic.
5. All relevant aspects of the study are described in another, more complete (superset) study.
6. The study is merely a comparative analysis of existing approaches.

A study was included in the SLR if it satisfied at least one of the inclusion criteria, but none of the exclusion criteria. If a study matched any exclusion criterion, the study was discarded from the SLR. Note that a study was considered without regard to its publication date.

3.5 Quality assessment

The field of data-centric BPM is considered to be rather immature compared to other BPM topics [62]. Most approaches are only covered in few papers and do not consider the entire business process lifecycle. Applying rigorous quality criteria, e.g., insisting on a proper evaluation of the approach, would have probably led to the exclusion of several (potentially relevant) studies, further reducing the already rather low number of included studies. As the purpose of the SLR is to discover "fresh" data-centric approaches and perform a comparison between them, we decided against an additional selection with quality criteria.

3.6 Selecting the studies

The search string defined in Sect. 3.2 was used to conduct a Google Scholar search. The search query yielded a total of 980 potentially relevant studies. For a better analysis, the relevant metadata was exported to an Excel file[3]. Metadata included the title, author, source, number of citations, and URL. Based on the metadata, each study was reviewed for investigating its relevance to the SLR, using the inclusion and exclusion criteria defined in Sect. 3.4.

The review started with examining the title of the studies. Studies having titles that clearly did not deal with data and processes were immediately discarded as they did not match any of the inclusion criteria. This filtering yielded a total of 88 potentially relevant studies, which were provisionally included in the SLR. Then, an extensive *backward reference search* was performed by considering literature cited by the studies themselves. Additionally, to obtain recently published relevant studies, studies that cited the already included relevant studies were evaluated as well. In the end, we obtained 89 additional studies which were added provisionally to the SLR.

To reduce the chance of missing a relevant study, we used Google Scholar's "Cited by" feature, which allows extracting any literature that references a particular paper. However, this way we did not identify further studies. Finally, a Google Scholar alert using the search string was established to keep the authors informed about newly published studies that might be relevant. The alert contributed one additional study for the SLR. To sum up, the search string, the backward reference search, and the Google Scholar alert yielded 178 provisionally included studies in total.

Each of the 178 studies was read thoroughly and assessed systematically through the inclusion and exclusion criteria. This in-depth analysis resulted in the identification of 38 *primary studies* (cf. Table 1) that were included in the final SLR, while the other 140 studies were discarded. The workload was divided up between the authors of this paper. Random studies were checked by other authors to ensure consistency and correctness. The final decision whether or not to include the study was reached by majority rule.

Table 1 List of primary studies

Study identifier, authors, and bibliography reference	
S01-Meyer et al. [48]	S20-Hariri et al. [3]
S02-Neumann et al. [52]	S21-Calvanese et al. [8]
S03 Bagheri Hariri et al. [4]	S22-Calvanese et al. [9]
S04-Belardinelli et al. [5]	S23-Russo et al. [66]
S05-Bhattacharya et al. [6]	S24-Westergaard et al. [80]
S06-Cangialosi et al. [10]	S25-Kumaran et al. [38]
S07-Damaggio et al. [15]	S26-Zhang et al. [82]
S08-Deutsch et al. [17]	S27-Künzle [39]
S09-Eckermann et al. [22]	S28-Künzle et al. [41]
S10-Hull et al. [34]	S29-Künzle et al. [40]
S11-Liu et al. [45]	S30-Eshuis et al. [23]
S12-Nigam et al. [54]	S31-Küster et al. [44]
S13-Solomakhin et al. [69]	S32-Ryndina et al. [67]
S14-Vaculín et al. [72]	S33-Wahler et al. [77]
S15-Xu et al. [81]	S34-Haddar et al. [25]
S16-van der Aalst et al. [75]	S35-van der Aalst et al. [74]
S17-Kurz et al. [42]	S36-Vanderfeesten et al. [76]
S18-Kurz et al. [43]	S37-Haesen et al. [26]
S19-Müller et al. [50]	S38-Haesen et al. [27]

3.7 Data extraction and analysis

All 178 provisional studies were subjected to a data extraction process with the intent to gain answers to the research questions (cf. Sect. 3.1). The extraction process consisted of three stages, and every result was captured in an Excel sheet. In detail, the extraction process was as follows:

– Stage 1: For each study, general information was extracted, i.e., title, authors, publication year, and venue.

[3] The Excel file can be found at https://bit.ly/2EZwG5b.

If applicable, the study was categorized according to the underlying process management approaches, e.g., artifact-centric or object-aware.
- Stage 2: The study was analyzed according to the inclusion and exclusion criteria. If the study was included in the SLR, the data extraction progressed at Stage 3. Otherwise, the study was excluded and the data extraction was considered as complete.
- Stage 3: For each research question, answers were extracted from all included studies. Remarkable and significant properties of the approach described in the study, which were outside of the scope of the research questions, were identified as well.

The gathered data were aggregated and displayed using descriptive techniques. Additionally, different terms with the same meaning were unified in order to improve overall consistency and facilitate statistical analyses.

4 Threats to validity

This section discusses factors that may call the results of the SLR conducted in this paper into question or diminish the meaningfulness of the results. These factors are denoted as *threats to validity*.

As we consider *selection bias* to be the primary threat to validity for the SLR conducted in this article, the SLR carefully adheres to the guidelines outlined in [36] in order to minimize selection bias. Concretely, we used well-known literature sources and publication libraries. These include the most important conference proceedings and journals on the topic of data-centric process management. Backward reference searching and Google Scholar Citation lists were scanned to find studies that were not found in the initial search using the search string. As a reference for the quality of the study selection, we ensured that relevant literature previously known to us was found by the SLR as well. This way, we ensured that the study selection was as complete as possible, thereby minimizing the risk of excluding relevant papers. Furthermore, as the literature search was conducted in 2016 and 2017, we kept up-to-date with more results by means of Google Scholar Alerts throughout the analysis and writing phase. The finalization of this work was achieved in early 2017, therefore papers published after February 2017 was not included in the SLR.

The studies identified by the literature search were divided up among the authors to determine, for each paper individually, whether it should be included in the SLR. Each author was continuously checked by another author to ensure the consistency of the selection process and the correct application of the inclusion and exclusion criteria. Disagreements on study inclusion were discussed and resolved by majority vote. Papers with similar or identical content were eliminated by trying to find a "superset" paper, i.e., selecting a paper which completely contains the relevant content of the other. This superset paper selection was performed by at least two authors. The date of the publication and the relevance to the research questions were factored in.

The second threat to validity consists of possible *inaccuracies in the data extraction and analysis*. As with our efforts to minimize *selection bias*, we adhered to the strict guidelines of [36] for an objective and replicable data extraction process to reduce bias. For data extraction and analysis, the studies were again divided among the authors. The work of each author was reviewed by at least one other author. Studies that did not provide clear, objective information were reviewed by all authors. In the review, the authors discussed the problems with the study, resolving issues by majority vote.

Another threat to validity is the low number of primary studies. Of the 38 primary studies that were included in the SLR, on average there are one or two studies per approach (with exception of the *Artifact-centric Approach*) containing information regarding research questions. This might endanger the overall accuracy of the representation of an approach in the SLR. Additionally, studies might not describe existing features or concepts of an approach, i.e., there might be an information gap between the information published in research papers and the actual status of an approach. Possible reasons for this information gap include the prototypical or unfinished status of a feature or concept. Furthermore, the respective feature or concept of an approach might not have been published due to its perceived irrelevance for the research community. This information gap adds to the inaccuracy when representing an approach in the SLR.

Finally, the SLR may be threatened by *insufficient reliability*. To address this threat, we ensured that the search process can be replicated by other researchers. Of course, the search may then produce different results, as databases and internal search algorithms of libraries may have been changed or updated. Additionally, as the process of creating an SLR also considers subjective factors, such as varying interpretations considering inclusion criteria, other researchers might come to different conclusions and, hence, will not obtain exactly the same results as presented in this paper.

5 Results

This section presents the major results of the SLR. We performed an initial analysis of the primary studies by classifying them based on their modeling approaches. Table 2 summarizes the results.

The majority of papers belong to the Artifact-centric Approach (13 studies). This is due to the high attention the verification of artifact-centric system has spawned. Data-

Table 2 Process modeling approaches adopted by the primary studies

Approach	Study ID	# of studies
Enhanced Activity-centric Approach	S01	1
Document-based Approach	S02	1
Artifact-centric Approach	S03–S15	13
Case Handling Approach	S16	1
Case Management Approach	S17–S18	2
Corepro Approach	S19	1
Data-centric Dynamic Systems Approach	S20–S23	4
Constraint-based Data-centric Approach	S24	1
Information-centric Approach	S25	1
Distributed Data Objects Approach	S26	1
Object-aware Approach	S27–S29	3
UML Object-centric Approach	S30	1
Object-centric Approach	S31–S33	3
Opus Approach	S34	1
Proclet Approach	S35	1
Product-based Approach	S36	1
Stateless Process Enactment Approach	S37–S38	2

centric Dynamic Systems (4 Studies) have evolved from such a verification approach into a full data-centric process modeling approach. Notable in the number of studies are the Object-centric (4 studies) and Object-aware (3 studies) approaches, as well as Case Management (2 studies). The remainder of the studies belong to other approaches (11 studies).

The remainder of this section presents the detailed results of the SLR, answering each research question separately (cf. Sects. 5.1, 5.2, 5.3, 5.4, 5.5 and 5.6).

5.1 Data representation constructs

This section presents the results related to research question RQ1, which focuses on the identification and definition of constructs used to represent data. We use the term *data representation construct (DRC)* (cf. Definition 1) to address the different definitions of structured data in the context of data-centric approaches.

Table 3 answers RQ1 by providing an overview as well as a short description of the DRCs used in the data-centric approaches identified in the SLR. Note that sometimes there may exist slightly different DRC definitions for the same approach, as the approach may be discussed in several papers with different goals in mind. To untangle this issue, we decided to use a common denominator reflecting the essentials of each DRC.

Before conducting the SLR, our expectation was that the majority of data-centric approaches use a kind of entity (e.g., objects, artifacts) that comprises a set of attributes to form a semantically related group. Out of the 16 identified approaches, 11 use DRCs with attributes, confirming our expectations. While these approaches are similar regarding the basic DRC descriptions they provide (i.e., entities with attributes), they vary significantly in regards to the data types of the attributes as well as the nesting of DRCs.

More precisely, some approaches limit the values of individual attributes to primitive data types (e.g., strings, integers), while others allow for more complex data types (e.g., lists, maps). Furthermore, some approaches support nesting, allowing a DRC to contain other DRCs. Consider the DRC representing a study plan (cf. Example 1) which may contain a DRC representing an exam description.

However, a data-centric approach does not necessitate an entity with attributes, as evidenced by the *Proclet Approach*, the *Document-based Approach*, the *Constraint-based Data-centric Approach*, the *Product-based Approach*, and the *Data-centric Dynamic Systems Approach*. These approaches operate on possibly unstructured data, as they have no formal requirement regarding the structure of the data. The *Document-based Approach*, for example, operates on documents (e.g., PDF or Excel files) referred to as Alphadocs, which may be subdivided into Alphacards.

The *Constraint-based Data-centric Approach* uses colored Petri net tokens to represent data. The data are not grouped into a parent entity, whereas the *Proclet Approach* uses a separate knowledge base for each Proclet. Proclets are lightweight processes that are defined with Petri nets. The

Table 3 Overview of the data representation constructs employed by data-centric approaches

Approach	DRC	Description
Enhanced Activity-centric Approach	Data Object	An extension of BPMN data objects: A data object consists of an object identifier, a set of attributes, a dedicated lifecycle, and a set of fields to express correlations
Document-based Approach	Alphadoc	Alphadocs are either content documents or coordination documents. Content documents are documents in the traditional sense (.txt or .pdf). Alphadocs are divided into Alphacards
Artifact-centric Approach	Business Artifact	Business artifacts consist of an information model and a lifecycle model. The information model provides attributes that may be atomic values or complex nested entities
Case Handling Approach	Case File	A case consists of a collection of data objects. Each data object may hold a single value or a collection of values
Case Management Approach	Case File and Data Objects	Data objects consist of a finite set of attributes and are grouped in a case file
Corepro Approach	Objects	Objects have a finite set of data attributes as well as an attached lifecycle process
Data-centric Dynamic Systems Approach	Tuples	Data is represented as tuples in a relational database
Constraint-based Data-centric Approach	CPR tokens	Default definition of a colored petri net token
Information-centric Approach	Business Entity	A business entity has an associated data model and behavior model existing in the context of a process
Distributed Data Objects Approach	Data Object	Data objects consist of a finite set of attributes and a finite set of states. Nesting is possible
Object-aware Approach	Object	Objects have a finite set of data attributes and an attached lifecycle process
UML Object-centric Approach	Object	Stateful objects in UML Activity Diagrams.
Object-centric Approach	Object	Objects. No details on attributes or lifecycles are specified
Opus Approach	Data Structure	A Data Structure has a finite set of attributes and a finite set of tuples. Each tuple entry corresponds to a value for an attribute
Proclet Approach	Knowledge Base	A Proclet has a knowledge base, storing relevant information. The knowledge base is not formally defined
Product-based Approach	Product Data Model	A Product Data Model is a directed acyclic graph representing all data items needed for a business process
Stateless Process Enactment Approach	Business Object	A business object consists a finite set of attributes

Table 4 Behavior description of the different approaches

Approach	Behavior description
Enhanced Activity-centric Approach	Lifecycle: extended BPMN
Document-based Approach	Custom
Artifact-centric Approach	Lifecycle: Guard-Stage-Milestone
Case Handling Approach	Lifecycle: custom
Case Management Approach	Lifecycle: CMMN
Corepro Approach	Lifecycle: unspecified
Data-centric Dynamic Systems Approach	Atomic actions/Tasks
Constraint-based Data-centric Approach	Declare, Dynamic Condition Response Graphs and Colored Petri Nets
Information-centric Approach	Lifecycle: State Machines
Distributed Data Objects Approach	Lifecycle: Colored Petri Nets
Object-aware Approach	Lifecycle: micro-process (custom)
UML Object-centric Approach	Lifecycle: Hierarchical UML State Charts
Object-centric Approach	Lifecycle: Business State Machines
Opus Approach	Colored Petri Nets
Proclet Approach	Petri Nets
Product-based Approach	Operations (custom)
Stateless Process Enactment Approach	Preconditions, postconditions and effects

contents of the knowledge base of a Proclet are arbitrary and may be defined as needed. In particular, the knowledge base contains the *performatives* (messages) exchanged between Proclets.

The *Data-centric Dynamic Systems Approach (DCDS)* abstracts from entities and represents data as tuples in a database. DCDS relies on a well-formalized approach to represent processes and data, which facilitates the application of verification techniques. The *Product-based Approach* defines its DRCs through a Product Data Model, which corresponds to a directed acyclic graph representing all required data items. As such, it does not aim to provide generic process support, but instead aims at directly supporting the delivery of an *informational product*. It is assumed that this informational product, e.g., a decision on an mortgage claim [76], is assembled from different components, e.g., interest rates and gross income per year. Thereby, the identified product data model is in charge of describing these components, i.e., the respective data items.

In the SLR analysis, we found one approach (*Enhanced Activity-centric Approach* [48]) devoted to extend a non-data-centric approach with advanced data-centric capabilities. Specifically, the *Enhanced Activity-centric Approach* improves a traditional data element of BPMN by replacing it with a data object, which contains attributes, has a dedicated lifecycle, and can be correlated with other data objects as well.

5.2 Behavior

Regarding Research question RQ2, we want to investigate how a DRC acquires the data relevant to achieving process goals. More precisely, RQ2 investigates how an approach defines *behavior* in this context. Table 4 summarizes the different methods and notations used for specifying behavior.

Ten approaches use a lifecycle model to specify behavior: *Enhanced Activity-centric Approach*, *Artifact-centric Approach*, *Case Handling Approach*, *Case Management Approach*, *Distributed Data Objects Approach*, *Information-centric Approach*, *UML Object-centric Approach*, *Corepro Approach*, *Object-aware Approach*, and *Object-centric Approach*. Coincidentally, the majority of these approaches represent a DRC as an entity with attributes. Though lifecycle processes increase the cohesion between process and data, it is by no means superior to other kinds of behavior, i.e., non-lifecycle behavior specification.

A popular choice for describing behavior are Petri nets and, especially, colored Petri nets, as they explicitly consider data. This choice was made, for example, in the *Opus Approach*, which provides formal semantics and allows for comprehensive correctness verification of behavior. For the same reason, two approaches (i.e., *Case Handling Approach* and *Information-centric Approach*) use state machines for specifying behavior. Finally, *UML Object-centric Approach* uses UML statecharts to represent the behavior of a DRC. All other approaches either apply a completely individual way of describing behavior (e.g., *Document-based Approach* or *Product-based Approach*), or combine and customize

Table 5 Interaction descriptions of the different approaches

Approach	Interaction description	Separation of behavior/interactions
Enhanced Activity-centric Approach	Extended BPMN	No
Document-based Approach	Custom: Coordination Documents and Treatment Status Artifacts	Yes
Artifact-centric Approach	Guard-Stage-Milestone	No
Case Handling Approach	[Not applicable]	[No]
Case Management Approach	CMMN	No
Corepro Approach	Coordination Model Machine	Yes
Data-centric Dynamic Systems Approach	CRUD operations and external function calls	Yes
Constraint-based Data-centric Approach	Declare, Dynamic Condition Response Graphs and Colored Petri Nets	No
Information-centric Approach	Synchronized State Machines	Yes
Distributed Data Objects Approach	Colored Petri Nets	No
Object-aware Approach	Macro-Process (custom)	Yes
UML Object-centric Approach	UML Activity Diagram, coordinator object	Yes
Object-centric Approach	Business State Machines	No
Opus Approach	Colored Petri Nets	No
Proclet Approach	Performatives/Petri Nets	No
Product-based Approach	[Not applicable]	[No]
Stateless Process Enactment Approach	Preconditions, postconditions, effects	No

existing methods (e.g., the *Constraint-based Data-centric Approach* uses Declare and Petri Nets together with Dynamic Condition Response Graphs). We consider this as an indication for the rather low maturity of contemporary data-centric approaches, as no consolidation of different concepts and notations has taken place so far. For activity-centric process management, BPMN has been widely accepted as the standard modeling notation.

5.3 Interactions

Research question RQ3 intends to find out whether and, if applicable, how interactions between different processes or DRCs are modeled in existing data-centric approaches. Interactions between processes and DRCs are used either to share different data or coordinate the execution of different processes and DRC lifecycles, respectively. Table 5 summarizes the results we obtained when investigating research question RQ3.

Almost all approaches allow for some kind of interaction between processes and DRCs, respectively. An exception to this is the *Product-based Approach*, which focuses on the product assembly process. To this end, the approach presumes knowledge about the individual components of a product. Accordingly, the process needs no interactions with other processes.

Another exception is the *Case Handling Approach*, where a case subsumes every activity and data object. As the concept requires that all data is part of a case and each case is isolated, cases cannot have interactions.

The SLR revealed two classes of interaction modeling. The first-class comprises approaches which separate interaction modeling from the modeling of the behavior, i.e., they use different languages to describe behavior and interactions. In general, this leads to a loose coupling of behavior and interactions. A representative of this class is *Object-aware Approach*, where a *micro-process* describes the lifecycle of an object (i.e., its behavior), whereas a *macro-process* describes the interactions among different objects. The second class comprises approaches that integrate the descriptions of behavior and interactions, i.e., the description of the interaction is part of the process model and, hence, a tight coupling between behavior and interaction modeling exists. A representative of this class is the *Proclets Approach*, where messages, called *Performatives*, are exchanged between the Petri Nets of different Proclets. Another representative is the *Artifact-centric Approach*, which uses GSM to describe both the behavior of an artifact and its interactions with other artifacts. Which approaches separate behavior modeling from interaction modeling is indicated by column "Separation" in Table 5.

Separating behavior from interaction modeling offers several advantages, in particular the loose coupling which

Table 6 Process enactment mechanisms of the different approaches

Approach	Enactment mechanism	Data-driven
Enhanced Activity-centric Approach	extended BPMN XML	No
Document-based Approach	active properties/documents.	Yes
Artifact-centric Approach	GSM (Tasks and PAC-Rules)	Yes
Case Handling Approach	State Transition diagrams and ECA rules	Yes
Case Management Approach	GSM (ECA rule application)	Partially
Corepro Approach	Abstract State Machines	Partially
Data-centric Dynamic Systems Approach	Enactment of atomic actions	Yes
Constraint-based Data-centric Approach	Task execution and constraint satisfaction	Partially
Information-centric Approach	States changed by activities	No
Distributed Data Objects Approach	Colored Petri nets	Partially
Object-aware Approach	Custom, based on ECA Rules	Yes
UML Object-centric Approach	UML State Machines (Behavior),Token-passing (Interactions)	Partially
Object-centric Approach	Activities, ECA Rules	No
Opus Approach	Colored Petri Nets	Yes
Proclet Approach	Petri Nets	No
Product-based Approach	Custom	Yes
Stateless Process Enactment Approach	Preconditions, postconditions, effects	Partially

allows changing the interaction constraints without affecting behavior. As a drawback, process models might be less comprehensible. Depending on the concrete goals of the respective approach, either integration or separation of behavior and interactions has proven to be more suitable. For example, the *Proclet Approach* integrates interactions of Proclets with the Petri nets describing Proclet behavior. This allows verifying the soundness of both process behavior and process interactions based on a well-defined formalism. Note that this becomes more complex when separating interaction modeling from behavior specification.

5.4 Process enactment mechanisms

This section presents the results related to research question RQ4, i.e., the enactment mechanisms of data-centric approaches. As opposed to traditional activity-centric approaches, in data-centric approaches, modeling and enactment of processes is driven by the acquisition of data instead of exclusively by control flow.

Table 6 presents the answers to research question RQ4. It can be noted that each approach has its individual enactment method. Guard-Stage-Milestone (GSM), however, is employed multiple times, both in the *Artifact-centric Approach* and in the *Case Management Approach*, where it forms the core of the *Case Management Model and Notation (CMMN)*.

Colored Petri nets, employed in the *Opus Approach* and the *Distributed Data Objects Approach*, are used multiple times as well. Note that such a variety of enactment methods provides evidence for the rather low maturity level of data-centric process management, as no consensus regarding the enactment of data-centric processes has been reached.

The identified enactment methods can be roughly classified into three categories: Petri nets, state machines, and rule-based enactment. Each of these categories offers specific advantages to data-centric processes. For example, Petri nets provide well-established, formal correctness verification techniques, whereas rule-based enactment allows for an increased flexibility when enacting processes.

The third column of Table 6 indicates which approaches may be considered as *data-driven* according to the criteria provided in Definition 4.

5.5 Management of process granularity

Research question RQ5 investigates how process granularity (i.e., the levels of abstraction of a process) is managed in the data-centric approaches.

As can be seen in Table 7, most approaches choose not to enforce any restrictions regarding the level of process abstraction. This allows for a variety of process models, ranging from abstract models, e.g., models for documentation, to less abstract models, e.g., executable process models. However, this variety comes with the usual drawbacks associated with different levels of granularity. Namely, these drawbacks include process models not being executable right away and process models of heterogeneous granularities being difficult to coordinate.

Table 7 Management of process granularity in different approaches

Approach	Levels	Differentiation	Sub-processes
Enhanced Activity-centric Approach	Unspecified	–	✓
Document-based Approach	Two	Units of Validation: Alphadocs and Alphacards	✗
Artifact-centric Approach	Unspecified	–	✗
Case Handling Approach	Unspecified	–	✓
Case Management Approach	Unspecified	–	–
Corepro Approach	Two	Object Alignment and Object Coordination	–
Data-centric Dynamic Systems Approach	Unspecified	–	–
Constraint-based Data-centric Approach	Unspecified	–	✓
Information-centric Approach	Unspecified	–	✓
Distributed Data Objects Approach	Unspecified	–	–
Object-aware Approach	Two	Object behavior and interactions	✗
UML Object-centric Approach	Two	Object behavior and interactions	✗
Object-centric Approach	Unspecified	–	✗
Opus Approach	One	Object alignment	–
Proclet Approach	One	Object alignment	–
Product-based Approach	One	Product Data Model	–
Stateless Process Enactment Approach	Unspecified	–	–

✓: Has support, ✗: No support, –: Unknown

Approaches that use objects as DRCs align their processes with business objects to facilitate overall coordination (e.g., *Proclet* and *Object-aware approaches*). Some of these approaches manage granularity to make each process executable at any time (e.g., *Object-aware Approach*, *UML Object-centric Approach*, and *Proclet Approach*). The approaches that separate behavior and interactions (e.g., *Corepro Approach*, *Object-aware Approach*, and *UML Object-centric Approach*, cf. Table 5) introduce a second level of granularity in addition to the object alignment. This second level of granularity explicitly deals with the interactions between the different objects of a process.

The *Document-based Approach* comprises two different levels of granularity. However, their distinction is based on the validation of a document, but not on the separation of concerns between behavior and interactions. As the *Document-based Approach* uses documents called Alphadocs, the primary level of granularity is the validation of an entire Alphadoc. The second level comprises the Alphacards of an Alphadoc. Each Alphacard can be validated individually and, depending on the outcome, different measures can be taken, e.g., invalidating the entire Alphadoc or taking measures to correct validation errors.

5.6 Tool support

Tool support for modeling and enacting processes in the context of data-centric approaches is indispensable in practice. With research question RQ6, we evaluated which phases of the process lifecycle are supported by tools, e.g., a modeling or run-time environment. Differentiating between the different lifecycle phases allows for better assessment of tool maturity. Table 8 shows the phases of the process lifecycle and whether tool support for this phase is provided by a data-centric approach.

It becomes immediately apparent that no tool support exists in data-centric approaches for the last phase of the business process lifecycle, i.e., the "Diagnosis and Optimization" phase. This can be seen as an indicator for the low maturity of data-centric approaches in general. However, six approaches are merely conceptual at this time, i.e., we could not find evidence of tool support.

Seven approaches provide support for both the "Design" Phase and the "Implementation and Execution" phase, the most prominent being the *Artifact-centric Approach* and the *Case Handling Approach*. Unlike all other approaches, several tools have been developed for the Case Handling Approach. The most widely known tool is for Case Handling is FLOWer [56,75], a process modeling and enactment tool. The *Constraint-based Data-centric Approach*, the *UML Object-centric Approach*, and the *Object-centric Approach* each provide only tool support for the "Design" phase of the business process lifecycle.

6 The DALEC framework

Using the research questions introduced in Sect. 3.1, the SLR results presented in Sect. 5, and the process lifecycle described in Sect. 2.2 as a basis, this section describes the *DALEC (Data-centric Approach Lightweight Evaluation*

Table 8 Tool support for different phases of the process lifecycle

Approach	Design	Implementation and execution	Diagnosis and optimization
Enhanced Activity-centric Approach	✓	✓	–
Document-based Approach	–	–	–
Artifact-centric Approach	✓	✓	–
Case Handling Approach	✓	✓	–
Case Management Approach	✓	✓	–
Corepro Approach	–	–	–
Data-centric Dynamic Systems Approach	✓	✓	–
Constraint-based Data-centric Approach	✓	–	–
Information-centric Approach	–	–	–
Distributed Data Objects Approach	–	–	–
Object-aware Approach	✓	✓	–
UML Object-centric Approach	✓	–	–
Object-centric Approach	✓	–	–
Opus Approach	✓	✓	–
Proclet Approach	–	–	–
Product-based Approach	–	–	–
Stateless Process Enactment Approach	–	–	–

✓: Has support, –: Unknown

and Comparison) framework. DALEC is used for evaluating, categorizing and comparing data-centric approaches. More precisely, for each stage of the process lifecycle, the framework defines a set of evaluation and comparison criteria (cf. Table 9). In addition to criteria specific to the process lifecycle, we also introduce criteria related to the applicability of the approach. The methodology on how the criteria of the DALEC framework were derived is presented in Sect. 6.5. The methodology is placed after the presentation of the criteria, as knowledge of the criteria helps to understand the justification of how these criteria were derived.

Most of the criteria use a 3-value scale consisting of the following values: *not supported*, *partially supported*, and *fully supported*. Finally, two criteria are evaluated using free text.

6.1 Design

The following criteria are related to the design-time phase of the process lifecycle. In particular, the modeling capabilities of the data-centric approaches are considered, including concepts such as verification and variants.

- **D01—Modeling Language**. The first criterion deals with the process modeling language used by the data-centric approach. This may include established languages (e.g., BPMN or EPC), adaptations of existing languages, or completely custom modeling languages specifically tailored to the respective data-centric approach.
- **D02—Specification of DRCs**. DRCs constitute the basic modeling elements of a data-centric approach (cf. Sect. 5.1). As it is likely that every data-centric approach must consider DRCs, we distinguish between *partial* and *full support* by how DRCS are defined. If their specification is fully formalized, the criterion is considered to be *fully supported*. If not fully formalized, a partially formal or informal specification (e.g., data objects in BPMN) is considered as *partial support*.
- **D03—Specification of Behavior**. This criterion refers to the design-time capability of an approach to model the behavior of DRCs at run time. An approach *fully supports* this criterion if it enables the formal specification of a behavior model (e.g., in the form of a DRC lifecycle process) at design time. An approach with a partially formal or informal behavior model specification has *partial support*.
- **D04—Specification of Interactions**. A data-centric approach partially supports this criterion if it provides means to specify interactions. For approaches without DRC lifecycle processes, interactions correspond to the interactions between processes in non-lifecycle processes (e.g., Proclets). The specification of interactions may be integrated with the behavior specification or be separated from it. This criterion is *partially supported* if partially formalized or informal specifications exist. *Full support* requires that the interaction specification is additionally completely formalized.

Table 9 Criteria defined by the DALEC framework

Nr.	Criterion	Criterion description	Scale
Design			
D01	Modeling Language	Describes whether a well-known language (e.g., GSM, BPMN, Petri nets) or a custom language specific to the approach is used.	Free Text
D02	Specification of DRCs	Does the approach allow for the specification of DRCs?	3-value scale
D03	Specification of Behavior	Does the approach allow for the specification of behavior?	3-value scale
D04	Specification of Interactions	Does the approach allow for the specification of interactions between processes?	3-value scale
D05	Support for Managed Process Granularity	Does the approach provide guidelines for managing process granularity? Does it enforce certain granularity levels?	3-value scale
D06	Support for Model Verification	Does the approach allow for the verification of a model (e.g., DRC, behavior) in respect to correctness criteria (e.g., absence of deadlocks and livelocks)?	3-value scale
D07	Support for Model Validation	Does the approach allow for the validation of a model (e.g., comparison with an ontology, trial execution runs)?	3-value scale
D08	Specification of Data Access Permissions (Read/Write)	Does the approach have an authorization concept for restricting or granting access to DRCs?	3-value scale
D09	Support for Variants	Does the approach allow for the definition of variants of DRCs and processes?	3-value scale
Implementation and Execution			
D10	Data-driven Enactment	Is the process execution data-driven (cf. Definition 4)?	3-value scale
D11	Operational Semantics for Behavior	Does the approach provide operational semantics for behavior?	3-value scale
D12	Operational Semantics for Interactions	Does the approach provide operational semantics for the interactions of DRCs and processes?	3-value scale
D13	Support for Ad hoc Changes and Verification	Does the approach support ad hoc deviations from the model(e.g., DRC model, behavior model)? Does the approach allow verifying the ad hoc changes?	3-value scale
D14	Support for Monitoring	Does the approach support monitoring of DRCs and processes?	3-value scale
D15	Batch Execution	Does the approach allow for simultaneous execution of multiple process instances as a batch?	3-value scale
D16	Support for Error Handling	Does the approach have some mechanism to handle or prevent errors at run-time (e.g., recovery)?	3-value scale
D17	Support for Versioning	Does the approach support different versions of DRCs and processes to exist simultaneously?	3-value scale
Diagnosis and Optimization			
D18	DRC Schema Evolution	Does the approach allow modifying DRCs to create new schemas? Does the approach allow for the migration of DRCs to a new schema?	3-value scale
D19	Behavior Schema Evolution	Does the approach allow modifying behavior to create new schemas? Does the approach allow for the migration of behavior to a new schema?	3-value scale
D20	Interaction Schema Evolution	Does the approach allow modifying interactions to create new schemas? Does the approach allow for the migration of interactions to a new schema?	3-value scale

Table 9 continued

Nr.	Criterion	Criterion description	Scale
Tool Implementation and Practical Cases			
D21	Design	Does the approach make a tool for supporting the modeling available?	3-value scale
D22	Implementation and Execution	Does the approach make a tool for supporting implementation and execution available?	3-value scale
D23	Diagnosis and Optimization	Does the approach make a tool for supporting diagnosis and optimization available?	3-value scale
D24	Practical Examples	Has the approach been practically evaluated?	Free Text

- **D05—Support for Managed Process Granularity**. Process granularity characterizes the level of detail of a business process (cf. Sect. 3.1.5). An approach with managed process granularity defines distinct levels of granularity for the processes and enforces them at design time. Managed granularity also exists when levels of granularity for processes are recommended, but not enforced. In contrast, an approach with unmanaged process granularity neither enforces nor recommends any granularity levels. Enforced managed process granularity scores *full support* in this criterion. *Partial support* is offered by approaches that recommend, but do not enforce managed process granularity. Finally, unmanaged process granularity, i.e., approaches that neither enforce, nor recommend granularity levels are considered to have *no support* for managed process granularity.
- **D06—Support for Model Verification**. Verification corresponds to the task of determining whether a process model is compliant with a specified set of correctness criteria. A *full support* of this criterion requires that all aspects of process and DRC modeling have formally specified correctness criteria. It must be formally decidable whether a process or DRC model is compliant or non-compliant with these criteria. If an aspect of the process model is lacking formalized criteria or the correctness criteria are only stated informally and, therefore, cannot be used for formal verification, the support is considered as *partial*. Fully supported verification implicitly depends on fully formalized DRC, behavior models, and interaction models.
- **D07—Support for Model Validation**. Validation ensures that a process model satisfies certain validation requirements that are specified before the modeling. The difference between validation and verification is illustrated with an example:
  If the goal was to model a study plan process, but instead a process model for managing lectures was actually created, the model would pass verification (if built correctly) but fail validation, as a process for managing lectures is not a study plan process.
  Full support of this criterion requires that the approach provides means to automatically validate a model against the validation requirements. Possible means are the comparison with an ontology or the formal specification of the validation requirements, which then can be used to formally validate the model. *Partial support* for this criterion requires the approach to provide means to simplify the validation by the process designers, e.g., by having trial runs. If the approach has neither, this criterion is considered as *not supported*.
- **D08—Specification of Data Access Permissions (Read/Write)**. This criterion evaluates the authorization concept of an approach. In addition to the permissions for executing activities in activity-centric process management, a data-centric approach must define access permissions for reading and writing a DRC and its individual attribute values. The criterion is considered as being *fully supported* if the access to data can be restricted to individual attributes within a DRC, i.e., access control is fine-grained. *Partial support* is provided if access permissions can only be granted to an entire DRC, i.e., a user can only be granted read/write permissions on all attributes of a DRC at once.
- **D09—Support for Variants**. A variant constitutes a derivation from a base entity, most often to adapt to specific circumstances and contexts (e.g., domain-specific, country-specific, or regarding specific legal constraints). For example, a DRC variant may either incorporate an additional attribute or lack an attribute that is unnecessary in the given context. The defining characteristic for a variant is that a variant stays closely related to its base entity. This means that changes made to the base entity propagate to its variants, which has the benefit of avoiding redundant changes. This criterion is considered *fully supported* if data-centric approaches support variants of both processes and DRCs. Having variants of either DRCs or processes constitutes *partial support*.

6.2 Implementation and execution

The following criteria are related to the run-time phase of the process lifecycle. Hereby, the enactment capabilities of a data-centric approach are evaluated along the basic concepts of business process execution.

- **D10—Data-driven Enactment**. The criterion evaluates whether the proposed execution mechanism of an approach is data-driven (cf. Definition 4). To be considered as *fully data driven*, the mechanism driving the execution of the data-aware process must fulfill all three criteria of the definition. If not fully supported, but at least one of the criteria is satisfied, the approach is considered as *partially data-driven*.
- **D11—Operational Semantics for Behavior**. This criterion checks whether an approach defines precise execution semantics for its process models regarding behavior. If all features of the model are supported at run-time, the criterion is considered as *fully supported*. If not fully supported, but at least one feature is supported at run-time, the criterion is considered as *partially supported*.
- **D12—Operational Semantics for Interactions**. This criterion checks whether an approach defines precise execution semantics for its process models regarding their interactions. If all features of the model are supported at run-time, the criterion is considered as *fully supported*. If not fully supported, but at least one feature is supported at run-time, the criterion is considered as *partially supported*.
- **D13—Support for Ad hoc Changes and Verification**. The criterion specifies whether the approach allows for ad hoc deviations from a DRC or process model at run time. Examples of ad hoc changes include the specification of an additional attribute for a DRC instance, the assignment of new permission for a specific instance for a user, and alterations of behavior processes. Ad hoc changes are employed at run time and usually concern specific instances. If the approach allows parts of both DRCs and process models to be altered, it is considered as *fully supported*. If ad hoc changes are limited to either DRCs or process models, the criterion is considered as *partially supported*.
- **D14—Support for Monitoring**. The monitoring of running processes keeps track of the execution status of process instances and DRCs in real time. It allows for the timely detection or prediction of problems in process execution. This may also include the generation of log entries as well as their real-time analysis. *Full support* exists if all aspects of processes and DRCs can be monitored in real time at run time; *partial support* exists if only a subset of aspects can be monitored at run-time or the real-time requirement is not met.
- **D15—Batch Execution**. This criterion specifies if the approach allows specifying batch operations on DRCs, their behavior, or interactions. A batch execution is defined as the simultaneous application of an action to a selection of instances. Examples include canceling all currently unfulfilled orders or the provision of a value to an attribute of selected DRCs. *Full support* exists if DRC, behavior and interactions may all be a target for batch executions, a *partial support* exists if at least one can be a target.
- **D16—Support for Error Handling**. Although many problems can be foreseen and handled at design time, unforeseen circumstances at run time might always occur, hindering or halting process execution. Therefore, it is preferred that an approach is able to cope with the problems at run time in an appropriate manner. Simple error handling mechanisms are the termination or restart of problematic process instances. Advanced error handling mechanisms include the prediction and detection of problems and the application of appropriate countermeasures without having to terminate or restart the process instance. The presence of a simple error handling mechanism (e.g., a `try-catch` mechanism) is considered as *partial support* of the criterion, whereas the presence of an advanced mechanism (e.g., automated recovery procedures) in addition to simple mechanisms is considered as *full support* of the criterion.
- **D17—Support for Versioning**. Similar to variants, versions constitute derivatives of base entities. However, a version does not stay connected to its base entity, i.e., changes of the base entity are not propagated to versions. In general, a version exists independently from other versions. Usually, versions are obtained by evolving DRCs or process models. Managing a myriad of versions of the same model poses a challenging problem for any data-centric approach. The criterion evaluates whether different versions may coexist in the same run-time environment. If both process and DRC versions are allowed, it is considered as *full support*. If either process or DRC versions are supported, support is considered to be *partial*.

6.3 Diagnosis and optimization

Schema evolution describes the process of adapting existing models to changing circumstances. The main difference between schema evolution of an existing model and the creation of a new (i.e., adapted) model, without schema evolution support, is that existing instances of the old model may be migrated to the new schema. Running instances of the old model are then updated with the new model information and continue with their execution. However, a migration might not be possible in all cases, e.g., a requested change

may no longer be possible due to the execution progress of a process instance. The capabilities for schema evolution in data-centric approaches are evaluated with the following criteria. The evolution of DRCs, behavior, and interactions are evaluated separately to give a more detailed overview.

- **D18—DRC Schema Evolution**. The capability to evolve existing DRCs and use the resulting schemas at run time is considered as *partial support*. If existing DRC instances may also be migrated to the new schema, schema evolution is considered to be *fully supported*. It is also considered as *full support* if all running instances are forced to migrate to the new schema and instances that cannot be migrated need to be deleted (no versioning).
- **D19—Behavior Schema Evolution**. Analogously to DRC schema evolution, creating new schemas of behavior models that exist in parallel to old schemas is regarded as *partial support* of this criterion. Allowing the migration of existing instances to the new schema is considered as *full support*.
- **D20—Interaction Schema Evolution**. If interaction specifications are separate from behavior specification in a data-centric approach, the criterion is evaluated separately. Otherwise, it is rated with the same score as schema evolution of behavior. The evaluation follows the same principles as the DRC and behavior schema evolution: Creating new schemas of interaction models that exist in parallel to old schemas is regarded as *partial support* of this criterion. Allowing the migration of existing instances is considered as *full support*.

6.4 Tool implementation and practical cases

A mature tool support is required for designing, implementing, executing, as well as monitoring process models created with a data-centric approach. Note that it is not necessary that distinct tools for each phase of the process lifecycle exist (cf. Sect. 2.2) as some approaches may combine functionality for multiple phases into a single tool.

- **D21—Design**. *Full support* of the "Design" phase signifies the presence of a (GUI-based) tool that allows specifying all aspects of process models and their associated DRCs. If the tool supports at least one, but not all modeling aspects, support of the approach for the "Design" phase is considered as *partial*. If no tool exists that supports the "Design" phase or the tool does not implement at least one modeling feature completely, the criterion is considered as *not supported*.
- **D22—Implementation and Execution**. In regard to the "Implementation and Execution" phase, *Full support* requires that a tool comprises an engine that is able to properly enact the complete operational semantics of the data-centric approach. If the tool merely supports a subset of the operational semantics, the support is considered as *partial*. If no tool exists that supports the "Implementation and Execution" phase, the criterion is considered as *not supported*.
- **D23—Diagnosis and Optimization**. Finally, for the "Diagnosis and Optimization" phase, the criterion is considered as *fully supported* if a tools exists that allows tracking the execution of process executions in real time. Additionally, the capabilities of the tool to use gathered data for improving the process models is considered. If the tool merely allows using gathered data for analyzing and improving the process models with real-time monitoring, the support is considered as *partial*. If no tool exists that supports the "Diagnosis and Optimization" phase, the criterion is considered as *not supported*.
- **D24—Practical Examples**. This criterion checks for practical applications and evaluations of the considered data-centric approach. The results, for example descriptions of applications in industrial settings or projects, are described using free text.

6.5 Criteria derivation

For the motivation of the criteria used in the DALEC framework, the primary source was the research questions. The following criteria were directly derived from the research questions presented in Sect. 3: criteria D01–D05 for the process design category, criteria D10–12 for the implementation and execution category, and criteria D21–24 for the tool implementation and practical use cases.

However, the research questions only cover the current development state of data-centric process management approaches to an extent. Data-centric approaches cover more aspects than captured in the research questions. As such, including only criteria based on the research questions would severely limit the applicability of the DALEC framework, as data-centric approaches gain new features not covered by the framework.

Therefore, we opted to include additional criteria to cover more developments in data-centric process management. For this purpose, we outlined the following meta-criteria for including or excluding these additional criteria in the DALEC framework:

1. Criteria that are subjective (e.g., understandability) are excluded for their potential for ambiguity.
2. A criterion is included when it has:
   (a) relevance from repeated mentioning in the papers considered in the SLR
   (b) significant added value for data-centric approaches, determined by author consensus

3. Final criteria count should be around twenty, with a reasonable distribution among the different BPM lifecycle phases.

In total, we considered 53 different criteria for creation of the DALEC framework, from which we included 24 according to the meta-criteria. The initial search for criteria was non-exhaustive, it was stopped when all authors agreed that there is a suitable pool of criteria to choose from. The additional criteria for the DALEC framework may be mainly categorized as follows, according to their source:

1. *Focus topics* are emphasized by the included data-centric approaches.
2. *Feature analogy* is derived from comparing features of data-centric approaches with activity-centric approaches.

During the analysis of the studies, it became evident that some data-centric approaches are focused on a very specific topic. Most notably, the artifact-centric approach puts a lot of emphasis on the correct verification of its artifact models, and much less focus on topics such as artifact execution. If such a *focus topic* was found during the study analysis, a discussion was started whether to include it as a criterion in the DALEC framework.

In detail, criterion "D06—Support for Model Verification" is primarily motivated by the artifact-centric approach for its particular emphasis of model verification. Also, other approaches recognize model verification as an important cornerstone of the functionality of the approach. One aspect of Case Handling is the recovery from run-time specific errors, as such, criterion "D16—Support for Error Handling" is included for the Case Handling approach. Authorization for data is topic in object-aware process management, therefore "D08—Specification of Data Access Permissions" was added as a criterion.

Activity-centric process management, in comparison with data-centric approaches, has received a significant number of feature extensions since its conception. Much of these features may be transferred to data-centric processes, creating a *feature analogy*. While data-centric approaches also have inspired features for the improvement of activity-centric processes, there is no denial that activity-centric process management possesses a significant advantage in feature count.

Criterion "D09—Support for Variants" is inspired by the large amount of approaches trying to make activity-centric process models more flexible by creating process variants, i.e., process models differing in specific areas from a base model. An overview and comparison for activity-centric variability approaches may be found in [2]. In addition, ad hoc changes to processes at run-time allow making processes more flexible, and are interesting for data-centric approaches. Therefore, criterion "D13—Support for Ad hoc Changes" was added to the DALEC framework. Criteria D18 to D20 are concerned with schema evolution of process models. While this counts as a feature analogy, the idea originally comes for relational databases and was itself adopted by activity-centric process management. Batch activities, captured in criterion "D15—Batch Execution," are both focus topic and feature analogy, as the object-aware approach and [59] introduce batch execution to their respective approaches.

Criterion "D07—Support for Model Validation" is included in the DALEC framework due to the general importance of validation for any model or system. Criterion "D17—Support for Versioning" is seen as a logical consequence to the schema evolution criteria D18–D20. Criterion "D14—Support for Monitoring" was added due to its interesting nature for data-centric process management, as in context of artifacts or the lifecycles of multiple DRCs, the question of status of the overall business process is non-trivial.

While the criteria certainly leave a lot of room for the improvement of data-centric approaches, the criteria included in the DALEC framework are of course only a fraction of what could be included. We concede that the selection, while done with consensus from all authors and in the best interest of overall applicability of the DALEC framework, is to a certain degree arbitrary and does not include many of the criteria other authors would deem important. However, adding hundreds of different criteria defeats the purpose of an applicable comparison framework. Additional criteria may be added as needed when using the DALEC framework for the comparison of data-centric approaches. If possible, these additional criteria should be derived with the guidelines outlined above.

7 Applying the DALEC framework to three prominent approaches

In order to illustrate the way our framework can be applied in practice, we exemplarily assess three selected data-centric approaches found in the context of the SLR. Specifically, our evaluation will consider the *Case Handling*, *Artifact-centric* and *Object-aware* approaches.

The selection of the three aforementioned approaches was performed using the following six-step procedure:

1. We used Google Scholar to collect the number of citations associated with each primary study included in the SLR. The number of citations for each paper was obtained in February 2017.
2. We grouped the studies based on the respective data-centric approach.

3. We selected a set of representative papers, where each representative paper belonged to a different approach and had the most citations of the respective approach.
4. We calculated the median number of citations for the set of representative papers.
5. We filtered out all the representative papers from the set for which the amount of citations was below the median.
6. For the approaches whose representative papers remained above the median, we filtered out the ones that did not provide any software tool supporting both the Design phase as well as the Implementation and Execution phase of the business process lifecycle. This was determined by the mentioning of tool support in the studies or demo papers.

The DALEC framework will be applied to the approaches represented by the remaining three representative papers. These representative papers belong to the *Case Handling, Artifact-centric, and Object-aware approach*.

The results of the above selection procedure are presented in Table 10. The selection procedure ensures that the three selected approaches are (1) well established and highly cited, and are (2) supported by a mature tool implementation.

The results of the application of the DALEC framework on the three selected approaches are investigated in the following section and outlined in Table 11. The values have been obtained from the primary studies of the approaches and the modeling of the running example. In Sects. 7.1 to 7.3, the approaches and their respective scores will be discussed in detail.

7.1 Applying the framework to the Case Handling approach

Case Handling [75] is an approach that was designed for the support of knowledge-intensive business processes. The central concept is the *case*, i.e., a collection of activities, data objects, and actors. In particular, activity execution is mainly driven by data flow instead of exclusively by control flow. An example of a case is the creation and assessment of study plans (cf. Example 1). An excerpt of the case, specifically the fragment referring to a student submitting the study plan, is depicted in Fig. 3 using the Case Handling notation.

When processing a case, *activities* need to be executed. Though these activities may be arranged in a precedence relation, their execution depends exclusively on the availability of case data. Data are represented as a collection of *data objects*. Case and data objects are formalized and consequently "D02—Specification of DRCs" is fully supported. "D03—Specification of Behavior" is fully supported due to the activities in a case as well as their precedence relations. In contrast to the other two approaches evaluated in Sect. 7, the *Case Handling Approach* provides no support for "D04—Specification of Interactions." This can be explained by the fact that Case Handling was developed under assumptions not captured in the research questions. Case Handling assumes that all relevant information is subsumed in one case, therefore cases need not interact with others.

Any activity must be connected to at least one data object through a *form*. Forms are used to present different views on the data objects associated with a particular activity. As shown in Fig. 3, the example case consists of nine activity definitions (e.g., "Login to the system" and "Add personal information") and six forms associated with them. Forms are used to collect relevant data objects for the activities, e.g., Form 1 is associated with the activity "Login to the system," which contains the data objects "University ID" and "Password."

In addition to *free data objects*, which may be associated with an entire case, there are two kinds of specialized data objects explicitly linked to one or more activities.

- Mandatory data objects require that their corresponding data fields in the form are filled in order to complete the corresponding activity. This does not mean that the corresponding activity is responsible for adding the information. The information might have been added by a previously executed activity of the case.
- Restricted data objects can only be modified by the specific activities they are associated with. For example, data object "Password" is both restricted to and mandatory for the activity "Login to the system," while "University ID" is not restricted to the activity. The "University ID" data object is mandatory for all activities of the case, even if its value is determined once during the execution of the first activity of the case, i.e., "Login to the system."

In addition, the *Case Handling Approach* allows for the specification of *roles* associated to process participants. Roles express the ability of a process participant to execute, skip or redo a specific activity. The definition of roles and the presence of specialized data objects enable an implicit mechanism for specifying data access permission. This corresponds to partial support for "D08—Specification of Data Access Permissions (Read/Write)." For example, users with the role "student submitting a study plan" may potentially execute any activity of the case in Fig. 3. When they execute activity "Login to the system," they may write both the "University ID" and "Password" data objects, but may not write any of the other data objects of the case, as they are associated with other activities.

Data objects are used by a case in the context of activities. Values for data objects may be required for the completion of an activity. For example, activity "Login to the system" completes only after providing values for the mandatory data objects "University ID" and "Password." Furthermore, the

Table 10 Study impact analysis (performed in February 2017)

Category	Study	Citations	Tool support
Enhanced Activity-centric Approach	S01	51(*)	✓
Document-based Approach	S02	22(*)	–
Artifact-centric Approach	S03	110	✓
	S04	34	
	S05	127	
	S06	30	
	S07	57	
	S08	1	
	S09	1	
	S10	121	
	S11	149	
	S12	464(*)	
	S13	19	
	S14	50	
	S15	25	
Case Handling Approach	S16	700(*)	✓
Case Management Approach	S17	4	✓
	S18	9(*)	
Corepro Approach	S19	144(*)	–
Data-centric Dynamic Systems Approach	S20	2	✓
	S21	1	
	S22	3	
	S23	4(*)	
Constraint-based Data-centric Approach	S24	22(*)	–
Information-centric Approach	S25	139(*)	–
Distributed Data Objects Approach	S26	0(*)	–
Object-aware Approach	S27	18	✓
	S28	61	
	S29	156(*)	
UML Object-centric Approach	S30	2(*)	–
Object-centric Approach	S31	173(*)	–
	S32	3	
	S33	19	
Opus Approach	S34	3(*)	✓
Proclet Approach	S35	124(*)	–
Product-based Approach	S36	60(*)	–
Stateless Process Enactment Approach	S37	4(*)	–
	S38	3	

✓: Has support, – : Unknown
(*) : highest citation count per approach

presence of certain data object values can also be used as a precondition for enacting activities. For example, activity "Create new study plan" can be executed if, and only if, the value of the data object "Kind of submission" is equal to "New." It is possible to combine such preconditions and the optional skipping or redoing of activities. This allows for

Table 11 Application of the framework to Case Handling, artifact-centric and object-aware

Nr.	Criterion	Case Handling	Artifact-centric	Object-aware
Design				
D01	Modeling Language	Custom	GSM	Custom
D02	Specification of DRCs	✓✓	✓✓	✓✓
D03	Specification of Behavior	✓✓	✓✓	✓✓
D04	Specification of Interactions	✗	✓✓	✓✓
D05	Support for Process Granularity	✓✓	✗	✓✓
D06	Support for Model Verification	✓✓	✓✓	✓✓
D07	Support for Model Validation	✗	✗	✗
D08	Specification of Data Access Permissions (Read/ Write)	✓	✗	✓✓
D09	Support for Variants	✗	✗	✗
Implementation and Execution				
D10	Data-driven Enactment	✓✓	✓✓	✓✓
D11	Operational Semantics for Behavior	✓✓	✓✓	✓✓
D12	Operational Semantics for Interactions	✗	✓✓	✓✓
D13	Support for Ad hoc changes and Verification	✗	✗	✗
D14	Support for Monitoring	✗	✗	✗
D15	Batch Execution	✗	✗	✓
D16	Support for Error Handling	✓	✗	✗
D17	Support for Versioning	✗	✗	✗
Diagnosis and Optimization				
D18	DRC Schema Evolution	✗	✗	✓
D19	Behavior Schema Evolution	✗	✗	✓✓
D20	Interaction Schema Evolution	✗	✗	✓
Tool Implementation and Practical Cases				
D21	Design	✓✓	✓✓	✓✓
D22	Implementation and Execution	✓✓	✓✓	✓✓
D23	Diagnosis and Optimization	✓	✗	✗
D24	Practical Examples	Insurance	Finance	Medical, HR

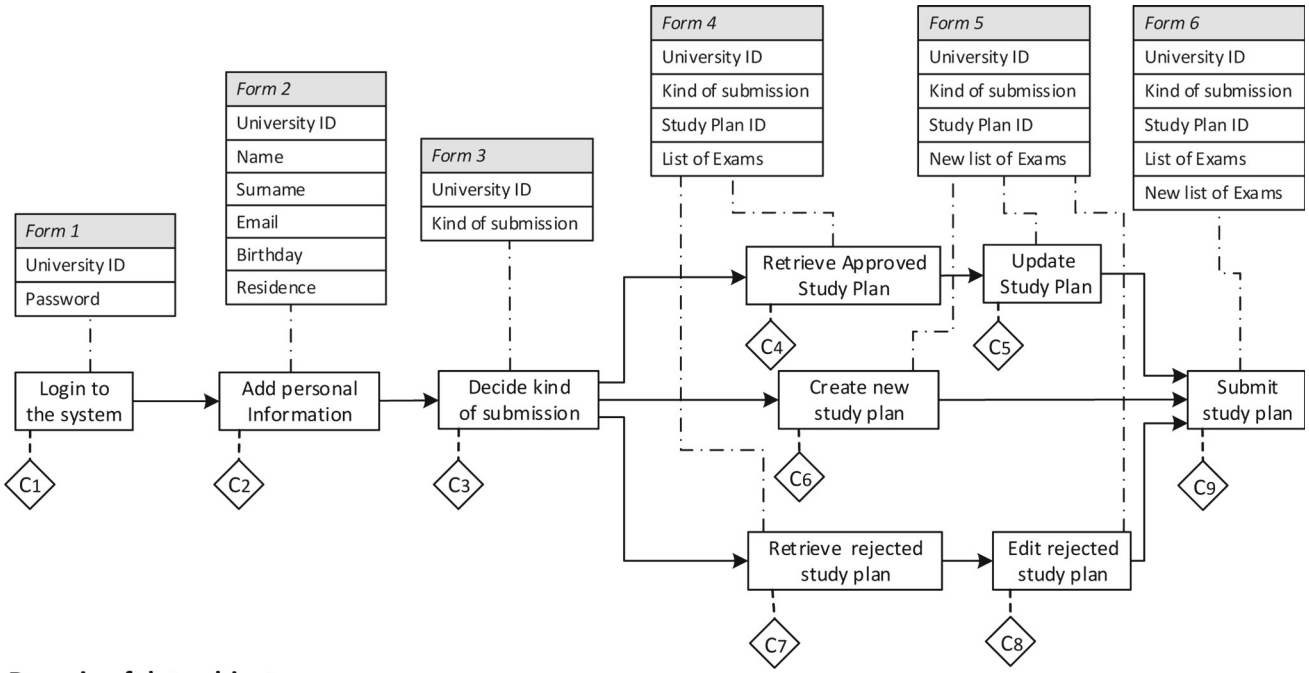
"✓✓" : Full support, "✓" : Partial support, "✗" : No support

the definition of primitive error handling mechanisms, i.e., "D16—Support for Error Handling" is partially supported.

As support for "D04—Specification of Interactions" is missing, consequently, the *Case Handling Approach* does not support "D12—Operational Semantics for Interactions." However, the *Case Handling Approach* has full support for "D11—Operational Semantics for Behavior." Furthermore, the behavior of a case is data-driven, as its progress is determined by the values of the data objects.

According to Definition 4, the *Case Handling Approach* provides full support for "D10—Data-driven Enactment." Cases are divided into complex cases (i.e., having an internal structure) and atomic cases (i.e., without any internal structure); the latter correspond to activities in a complex case. Complex case definitions consist of a number of complex cases and atomic cases, resulting in a hierarchical structuring of cases in sub-cases and activities.

Over the years, several tools were developed for the *Case Handling Approach*, including the Staffware Case Handler [70], COSA Activity Manager [68] and Vectus [46]. Each of these tools covers specific features of the approach. However, the only available tool that is fully consistent with the *Case Handling Approach* meta model and the formal specification is the FLOWer System [56] developed by Pallas Athena. FLOWer consists of a number of software components: *(i)* FLOWer Studio is the graphical environment used to specify cases at design time, *(ii)* FLOWer Case Guide is the client application that handles individual cases at run-time and *(iii)* FLOWer Management Information allows recording and retrieving the entire history of a case, including time stamps, data changes and actors involved in its execution; However, no tool is provided for the analysis of such information. FLOWer has been evaluated in [51,70] through an insurance company's process for handling claims for motor vehicle damage. Regarding the category "Tool Implemen-



Domain of data objects

dom(University ID) = String; *dom*(Password) = String; *dom*(Name) = String; *dom*(Surname) = String; *dom*(Email) = String; *dom*(Birthday) = Date; *dom*(Residence) = String; *dom*(Kind of submission) = {New,Existent,Rejected}; *dom*(Study Plan ID) = String; *dom*(List of exams) = {Exam1,Exam2,...,Exam3}; *dom*(New List of exams) = {Exam1,Exam2,...,Exam3}

Typology of data objects

mandatory(Login to the system, {*University ID,Password*})
restricted(Login to the system, {*Password*})
mandatory(Add personal information, {*Name,Surname,Email*})
restricted(Add personal information, {*Name,Surname,Email,Birthday,Residence*})
mandatory(Decide kind of submission, {*University ID,Kind of Submission*})
mandatory(Retrieve Approved Study Plan, {*University ID,Kind of Submission,Study Plan ID,List of Exams*})
mandatory(Retrieve Rejected Study Plan, {*University ID,Kind of Submission,Study Plan ID,List of Exams*})
mandatory(Create new Study Plan, {*University ID,Kind of Submission,Study Plan ID,New List of Exams*})
mandatory(Update Study Plan, {*University ID,Kind of Submission,Study Plan ID,New List of Exams*})
mandatory(Edit Rejected Study Plan, {*University ID,Kind of Submission,Study Plan ID,New List of Exams*})
mandatory(Submit Study Plan, {*University ID,Kind of Submission,Study Plan ID,List of Exams,New List of Exams*})

Conditions

C1 = {} – C2 = {} – C3 = {} – C5 = {} – C8 = {} – C9 = {}
C4 = {Retrieve Approved Study Plan,{Kind of submission,*Existent*}}
C6 = {Create new Study Plan,{Kind of submission,*New*}}
C7 = {Retrieve Rejected Study Plan,{Kind of submission,*Rejected*}}

Fig. 3 The study plans management procedure represented through the Case Handling approach

tation and Practical Cases" of the DALEC framework, a modeling tool and an enactment tool, each with full capabilities, exist, i.e., "D21—Design" and "D22—Implementation and Execution" are fully supported. The Case Handling tools further have limited capabilities for "D23—Diagnosis and Optimization," resulting in partial support for this criterion.

7.2 Applying the framework to the artifact-centric approach

We use the *Artifact-centric Approach* with the Guard-Stage-Milestone (GSM) meta model to model the running example. The resulting model includes three different artifacts, i.e., "student," "change request," and "study plan" (cf. Figs. 4, 5 and 6). Each artifact consists of an *information model* and a

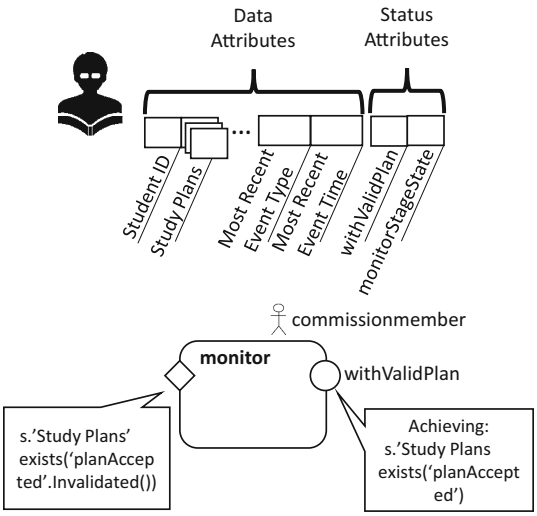


Fig. 4 The *student* artifact represented using the GSM notation

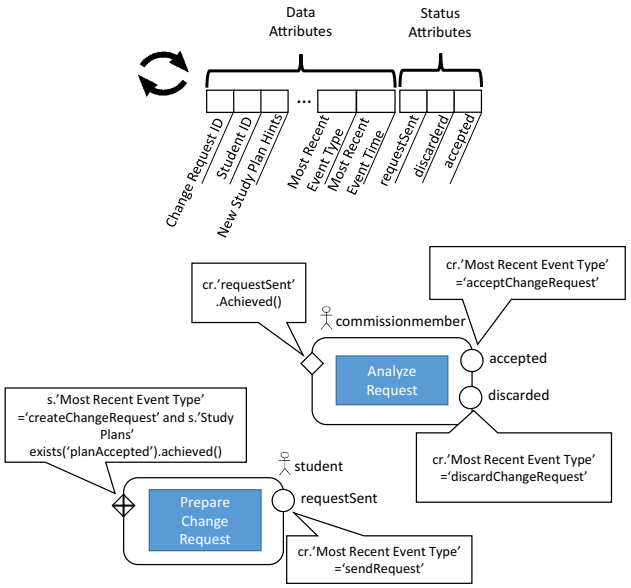


Fig. 5 The *change request* artifact represented using the GSM notation

lifecycle model. The *lifecycle model* is defined in GSM using stages associated with guards and milestones. Hereby, stages group individual activities and guards represent entry conditions to a stage. Finally, milestones represent operational objectives and are completed on the fulfillment of their corresponding conditions. Each *information model* includes two separate sets of attributes, denoted as data attributes and status attributes, respectively. Data attributes contain fields that store business-relevant information as well as fields that store events, e.g., the completion of a milestone. Status attributes are those related to the state of stages (`open` or `closed`) and milestones (`achieved` or `invalidated`). Artifacts are fully formalized, giving full support to the "D02—Specification of DRCs" criterion. The GSM meta model is also fully formalized. Since artifacts combine behavior and interactions in one model, this grants full support for the criteria "D03—Specifications of Behavior" and "D04—Specification of Interactions."

Figure 4 shows the model of the *student* artifact. For simplicity, the only relevant business-related attributes are the student ID and a history of study plans of which only one has the "planAccepted" milestone achieved. When the currently accepted plan is invalidated, the student enters the "monitor" stage that is completed when a new plan reaches the "planAccepted" milestone.

Figure 5 depicts the model for the *change request* artifact. The artifact is created when a new creation event, i.e., an external event requesting the creation of an artifact, is detected and an accepted study plan exists. At this point, the student may prepare the change request by specifying the changes she intends to make to her study plan. The stage is closed whenever the study plan is submitted to a commission member.

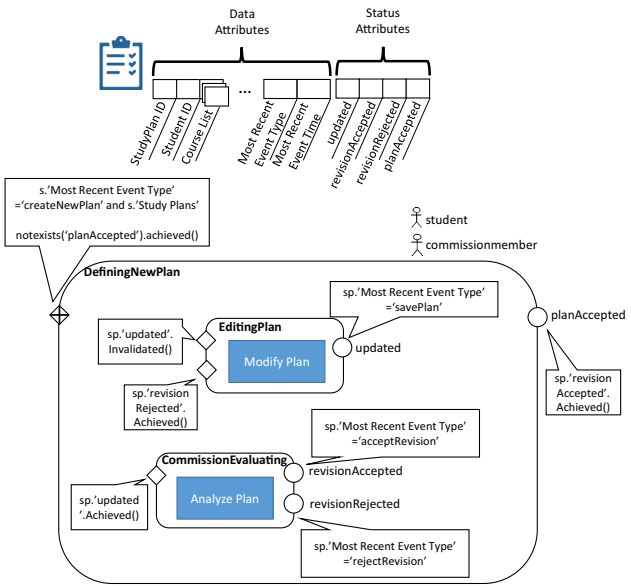


Fig. 6 The *study plan* artifact represented using the GSM notation

Closing milestone "requestSent" opens the stage in which the commission member must decide whether to accept or reject a change request. If the request is accepted, all milestones "planAccepted" of the study plans are invalidated. Otherwise, the request is rejected and a notification is sent to the student.

Figure 6 depicts the GSM model of the *study plan* artifact. A study plan can be created if no accepted study plans exist. The "DefiningNewPlan" stage is a compound stage consisting of two sub-stages, which alternate until the commission member approves the last update of the study plan. In particular, after a first saving operation (milestone "updated"

achieved) the commission member may decide to accept or reject the revision. If she rejects the revision, the "EditingPlan" sub-stage is re-opened (remember that in this case the updated milestone is automatically invalidated). Finally, whenever the "revisionAccepted" milestone is reached, the "planAccepted" milestone is reached as well.

For the *Artifact-centric Approach*, verifying the correctness of the models is of particular importance due to the number of papers concerned with it. [5] introduces a methodology to translate an artifact-centric process model into a so-called Artifact-Centric Multi-Agent System (AC-MAS). Once this translation has been accomplished, well-established verification techniques for logic formulas can be applied [5]. The verification of artifact-centric models has been extensively investigated, for example in [4,15,69], resulting in full support for "D06—Support for Model Verification." The GSM operational semantics is described in [15], covering both the enacting behavior and interactions between DRCs. Therefore, artifact-centric process management fully supports criteria "D11—Operational Semantics for Behavior" and "D12—Operational Semantics for Interactions." According to Definition 4, the operational semantics also support data-driven enactment of artifact-centric processes, which results in full support for "D10—Data-driven Enactment."

The modeling and enactment of artifact-centric process models with GSM is supported by a tool called Barcelona [30]. The tool has been made open source and relabeled BizArtifact[4]. BizArtifact comprises both modeling and execution environments, thereby scoring full support for criteria "D21—Design" and "D22—Implementation and Execution," respectively. The *Artifact-centric Approach* was applied during an extensive case study in the finance sector ([11], cf. Criterion D24).

7.3 Applying the framework to the object-aware approach

Analogously to the examples for *Artifact-centric* and *Case Handling* approaches, this section represents the "Study Plan" example process as an object-aware process model. Based on this model, we discuss the application of the framework to the *Object-aware Approach*.

For the object-aware example we use the modeling notation of the PHILharmonicFlows framework, the implementation of the *Object-aware Approach*. PHILharmonicFlows process models are split into multiple distinct models. The first relevant model is the *data model*, which describes the various *objects* participating in the process, as well the *relations* between them. The data model for the study plan example process is depicted in Fig. 7.

[4] Available for free at https://sourceforge.net/projects/bizartifact/.

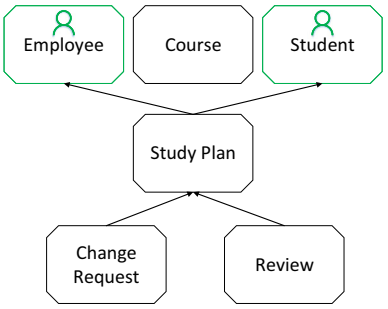


Fig. 7 PHILharmonicFlows data model

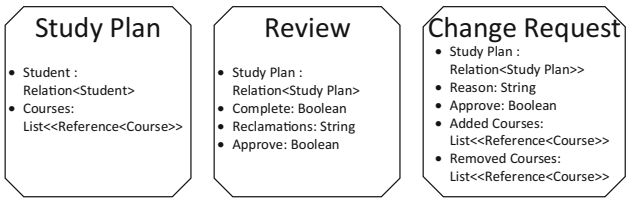


Fig. 8 Objects and attributes

The relations in the data model show a bidirectional link between two objects, e.g., a *Review* belongs to a *Study Plan* and a *Study Plan* has a *Review*. Note that *Course* does not have a relation to *Study Plan*; this is because *Courses* are only referenced by *Study Plans*, as they do not "belong" to them. Instead, each *Study Plan* has a list of references, each pointing to a *Course*.

The data model also allows defining the *attributes* present in each of the objects. The attributes for the most important objects can be seen in Fig. 8. Regarding the framework, the criterion "D02—Specification of DRCs" is fully supported, as the object-aware approach provides complete formal definitions for objects, attributes, relations and the data model.

Furthermore, each of the objects depicted in the data model has a so-called *micro-process* attached to it. The micro-process describes an object's lifecycle during the course of the process execution. The micro-processes are modeled separately for each object and can be viewed in Fig. 9. Micro-processes represent the behavior of a DRC in the *Object-aware Approach*.

Each micro-process consists of multiple *states*, e.g., *Creation*, *Evaluation*, *Approved*, and *Rejected*, for the *Change Request* micro-process. An instance of an object can only be in one state at any given time during the process execution. In an object-aware process, the state of an object is the only information immediately visible to other objects. The state is therefore used to coordinate execution with other objects. An example of this could be the following simple rule: If a *Review* is *Rejected*, the *Study Plan* that the *Review* belongs to shall change its *State* to *Rejected* as well. To determine in which state an object is, each of the states contains a sequential list of *steps*, each referencing one of the attributes of

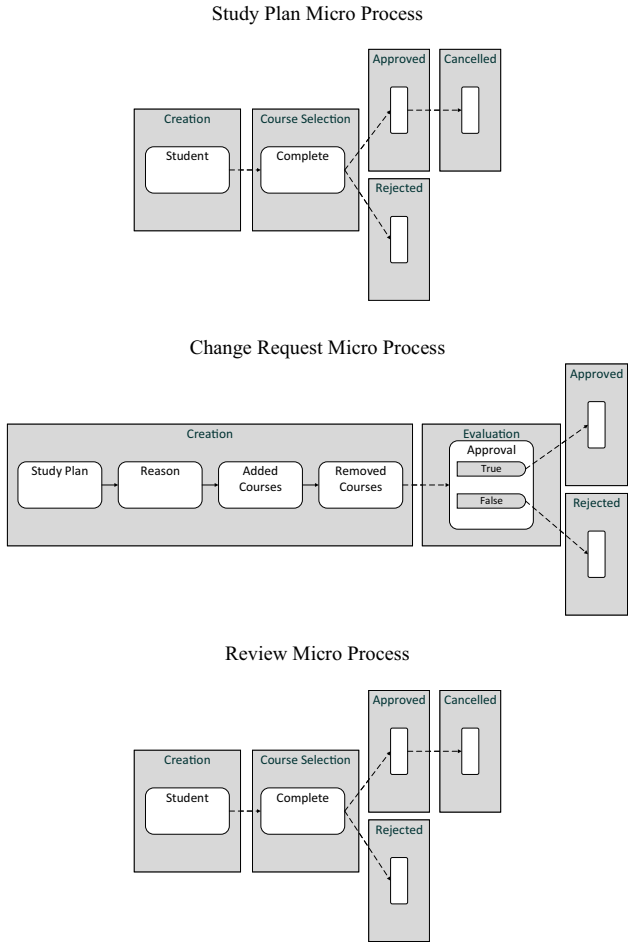


Fig. 9 PHILharmonicFlows micro-processes

the object type which the micro-process belongs to. Once all the attributes referenced by the steps of a certain state have values, the state is completed and the object may transition to the next state. The utilization of steps, states, and transitions, together with modeling elements not utilized in the running example, are formally defined in the *Object-aware Approach* [39]. Therefore, the full formal specification of micro-processes and its constituting elements awards full support for the "D03—Specification of Behavior" criterion.

To ensure that state transitions can be coordinated between different objects, as suggested in the example rule, an Object-aware process model also contains a so-called *macro-process*, which represents the coordination constraints that exist between the various object states. The macro-process is attached to one of the object types, instantiating that object type begins execution of the macro-process at run time.

The example contains a single macro-process attached to the *Study Plan* object. The macro-process, including the aforementioned example rule, is depicted in Fig. 10. In turn, macro-processes are on a different level of granularity than micro-processes, which is also enforced by having two different types of model for macro- and micro-processes. The "D05—Support for Process Granularity" criterion is therefore fully supported. Unsurprisingly, macro-processes must also adhere to a formal specification, which makes criterion "D04—Specification of Interactions" fully supported.

The full formal specification of objects, micro-processes, and macro-processes allows for the definition of formal correctness criteria. The *Object-aware Approach* enables a complete verification of all specified process models, thereby being fully compliant with criterion "D06—Support for Model Verification" of the DALEC framework. The *Object-aware Approach* allows for fine-grained data access down to attribute levels, however, authorization and access permissions were not modeled for the running example for the sake of brevity. Still, the *Object-aware Approach fully supports* "D08—Specification of Data Access Permissions (Read/Write)." The tooling for object-aware process management comprises a run-time environment on which modeled processes can be executed. The various process models are created and verified using the PHILharmonicFlows modeling tool.

The tools provide the core functionality for objects, micro-processes, and macro-processes. This awards full support for both criteria "D21—Design" and "D22—Implementation and Execution" in the section "Tool Implementation and Practical Cases" of the DALEC framework. The run-time tool [1] implements the operational semantics specified for micro- and macro-processes by the *Object-aware Approach*. The *Object-aware Approach* is therefore considered to have full support for criteria "D11—Operational Semantics for Behavior" and "D12—Operational Semantics for Interactions." Additionally, the execution of object-aware processes is fully data-driven, according Definition 4. Moreover, the

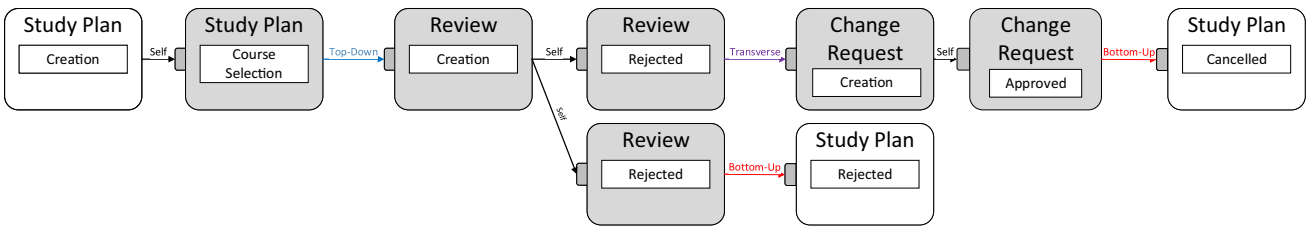


Fig. 10 Macro-process

Object-aware Approach allows for some batch executions on objects, i.e., the provision of data values for attributes of objects of the same type, resulting in partial support for the "D15—Batch Execution" criterion.

Finally, the *Object-aware Approach* has theoretical work on schema evolution as well [13,14]. At the time the research for this paper was conducted, this feature had not yet been implemented in the tooling. Additionally, the work merely focuses on schema evolution for micro-processes. Therefore, object-aware process management only partially supports "D18—DRC Schema Evolution" and "D20—Interaction Schema Evolution," but full support "D19—Behavior Schema Evolution." For other criteria, we could not find evidence that there is support (cf. Table 11).

Overall, the *Object-aware Approach* could closely represent the running example. However, the different models and different notations and concepts make the initial understanding of the approach hard, whereas their clear separation can prove to be an advantage once the initial hurdle is overcome.

8 Related work

The contribution of this paper consists of a framework for the systematic evaluation and comparison of data-centric process management approaches, which do not conform to the traditional activity-centric paradigm. To achieve this, a systematic literature review was conducted. The approaches identified in the literature review were analyzed and grouped according to differentiating criteria (cf. Sect. 6). To the best of our knowledge, there is no published work which has applied the concept of a systematic literature review to the field of business process management using data-centric approaches. There are, however, several literature reviews on related research fields, of which three are presented here.

A renowned publication in the field of data mining provides a comprehensive overview on 87 papers concerning the application of data mining techniques to customer relationship management (CRM) [53]. The authors perform a systematic literature review to derive a framework for classifying the various dimensions of a CRM system or approach. Examples of such dimensions are customer retention and customer identification. Furthermore, [53] provides a framework for the classification of data mining techniques by their capabilities. These two frameworks were then applied to the papers included in the literature review, resulting in an insightful overview on the research field. The result of the paper is the clear identification of domains in the CRM field that need additional research.

In the field of knowledge management (KM), [21] aims at identifying gaps in knowledge management endeavors in small and medium-sized enterprises (SMEs). The authors apply a systematic literature approach to categorize the 36 papers they deemed relevant into the different areas of knowledge management, such as perception, implementation, and utilization. Moreover, the authors describe each category and the criteria that a paper has to meet to fit be included into the respective category. They also present many tables in which all 36 papers are commented and systematically categorized and compared. From these tables, they concluded which of the areas of KM are not well researched in the context of SMEs.

Process variability support in process-aware information systems (PAIS) is considered in [2]. The authors also conduct a systematic literature review to assess the vast amount of approaches in the field of process variability. The paper presents the VIVACE framework for analyzing and comparing process variability approaches. The framework is also intended as a tool for process engineers implementing a PAIS, assisting the selection of appropriate approaches for the support of process variability along the entire process lifecycle. The systematic literature review, which analyzed 63 papers, provides the basis for the VIVACE framework. The papers were categorized not only by the process variability approach that they described, but also by the phase of the business process lifecycle they support, e.g., "process analysis" and "process enactment." The VIVACE framework defines 11 features that support process variability across the various process lifecycle phases. Each of the approaches mentioned in the literature review is categorized by the feature set it supports. However, [2] states that none of the approaches covers all features. Furthermore, the authors identified types of variability that are not yet supported by any approach, such as process variability in the temporal or operational perspectives.

The general conclusion of [53], [21] and [2] is that more research is needed in specific areas of the respective fields, a similar conclusion to the one drawn in this paper. The systematic literature reviews, as well as the categorization frameworks that were developed utilizing the literature review results, assist other researchers in identifying lacking research areas and categorizing their own work as well as new approaches in relation to existing research.

On the topic of the comparison of data-centric approaches, [64] evaluates data-centric approaches with respect to the interests of human modelers, i.e., usability and understandability. In summary, the authors determine that the usability of data-centric approaches is insufficient and must be improved for these approaches to be truly applicable in practice.

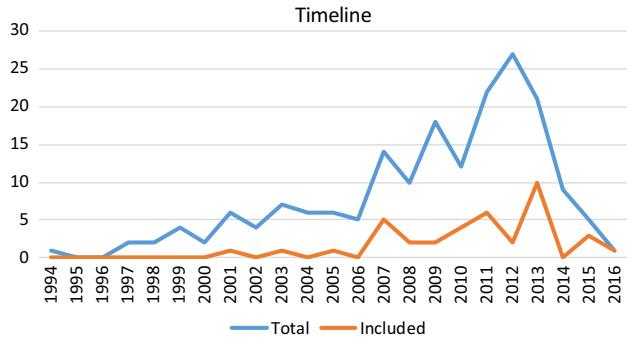


Fig. 11 Number of study publications per year

9 Discussion

The results presented in this paper allow for several interesting observations.

One positive aspect is the general interest in data-centric approaches, as demonstrated by the large amount of considered approaches. This interest can be explained with the more widespread application of BPM in different application domains, e.g., the Internet-of-Things (IoT) and ubiquitous systems, which drives the need for new and different approaches to business process modeling and execution [35]. The desire for data-centric approaches has been also confirmed by a survey among BPM practitioners [47].

When looking at the publication dates of these studies, it can be noted that the main body of papers was published between 2009 and 2014, with a significant peak in 2012, as can be seen in Fig. 11. The total curve in Fig. 11 refers to the 178 provisionally included studies, which showed some relevance to the research questions.

While the interest in data-centric approaches has somewhat subsided toward the year 2016, some approaches are still being developed (cf. [71]) or even emerge (cf. [16]). Note that these papers are just a few that have been published after completing the SLR in 2017. It can therefore be concluded that there is still interest, although at a lower level.

As this paper shows, the interest has spawned a significant number of diverse and interesting approaches; however, it also shows that the general level of maturity is comparatively low. As such, our basic assumption and motivation for the conduction of this SLR has been confirmed. We also do not see the decline in paper publications per year as completely negative. On the contrary, it may be a sign that a consolidation phase has begun and only the approaches with the highest potential survive. A sophisticated tool implementation is a major factor in this regard. In the end, this could be a boon to data-centric process management in particular and to business process management as a whole. Should this indeed be the case, a lower publication count is not unexpected.

As it emerges from the examples presented in Sect. 7, data-centric modeling is quite cumbersome and complex. As also discussed in [47], the practitioners' perception is that modeling with data-centric approaches is more complicated than modeling with activity-centric notations, such as BPMN. This might be a symptom of the low maturity of the approaches, indicating the need for further research. Notably, understandability of data-centric models and simplicity has not been addressed in DALEC. First, understandability and simplicity are rather subjective terms that have a fundamental different reliability than the objective criteria the DALEC framework comprises. While understandability and simplicity are certainly important aspects of data-centric process management, we understand DALEC as a framework that compares tangible features. Therefore, the subjective criteria have been left out. [64] provides a first empirical study regarding the usability of data-centric approaches. We aim at performing extensive experimental evaluations with BPM practitioners in future work; more details are outlined in Sect. 10.

Very few approaches appear to be universal and applicable solutions for data-centric business process management, as most of them focus on particular issues or on a specific domain. Examples of such a particular focus include the DCDS Approach, which deals exclusively with the verification of models, and the Alphaflow Approach, which was developed for working with documents in the medical domain. Of the few approaches, Case Handling and the object-aware approach strive to provide a universal data-centric approach, i.e., an approach that does not focus on a specific domain or topic.

Another interesting observation, with respect to the behavior specification in the various approaches, concerns the kind of notation used. When correlated with the publication date of their papers, approaches using a declarative concept to describe behavior (e.g., Case Management) were published more recently, whereas approaches using imperative description techniques are older. We assume that this is due to the increasing demand for process flexibility, which is more easily achievable using a declarative concept. The artifact-centric approach even switched from an initially imperative, state machine-based behavior model to the declarative Guard-Stage-Milestone framework. GSM was developed for use with artifacts and is also at the core of the CMMN standard for Case Management.

Notably, among all proposals, GSM appears to gain the most popularity among researchers, probably due to its industrial support and the availability of an open-source tool from IBM named BizArtifact. However, GSM is not the simplest approach in terms of complexity.

Furthermore, currently only one data-centric extension to the activity-centric approach exists, all other approaches are designed from scratch with their own constructs. Again, this

diversity, and the absence of a consolidated mainstream base, are symptoms of the low research field maturity.

In summary, the SLR shows that there are many different data-centric approaches, indicating that data-centric process management can offer significant benefits to business process management as a whole. The specification of processes around data, i.e., the DRCs, creates new and improved ways to handle data in business processes. The benefits include the creation of models that adequately represent such data-heavy real-world business processes as well as greater flexibility when enacting such processes. The application of the DALEC framework to the most prominent approaches shows that, in general, these approaches have full support for the modeling of DRCs, behavior and interactions (cf. Criteria D02-D04), as well as their operational semantics (cf. Criteria D10-D12). Considering that most data-centric approaches have been developed in very recent years, we see this a positive indication of growing maturity in data-centric business process management.

However, data-centric approaches generally do not go beyond the basic modeling and execution features, i.e., schema evolution, ad hoc changes and process variants (cf. Criteria D09, D13, and D18–D20) are not supported. This puts them at a decisive disadvantage compared to activity-centric process management, where many of these features have existed for a long time. We feel that this research gap requires serious attention from the BPM community. Moreover, the addition of the elaborate data perspective to these data-centric approaches increases the complexity of process modeling. Therefore, research is also required that effectively helps reduce and manage this added complexity. Otherwise, the benefits of data-centric approaches, including the adequate representation of data and increased flexibility, cannot be applied in practice.

10 Summary and outlook

This paper initially presented the results of a systematic literature review on data-centric approaches to BPM. The main insight gained from the SLR is that the interest in data-centric approaches to business process management has been significant over the last years, although the field itself is young and therefore the maturity of the individual approaches is varying and generally low.

The paper further presented the Data-centric Approach Lightweight Evaluation and Comparison framework (DALEC) for evaluating data-centric approaches. We applied this framework to three of the currently most prominent data-centric approaches, i.e., the *Case Handling Approach*, the *Artifact-centric Approach*, and the *Object-aware Approach*, reinforcing our findings in a practical setting.

As discussed in Sect. 5, the results obtained by the SLR show that data-centric approaches are still at an early development stage. Indicative of this fact are the not yet consolidated methods and languages, the missing tool support, the modeling complexity, and the lack of studies showing practical real-world applications. To make data-centric business process management applicable to real-world projects and systems, tool implementations that cover the whole business process lifecycle, as well as empirical studies that improve the usability and reduce the modeling complexity of data-centric approaches, are necessary. There may be signs that a consolidation phase has started, where at the end mature, practically relevant approaches remain.

As possible future topic, it would be interesting to evaluate how much the approaches analyzed in this paper provide a better, or more convenient, solution to modeling or executing processes in specific scenarios. Empirically, this can be evaluated by taking groups of practitioners and performing a modeling experiment to compare data-centric approaches with activity-centric approaches. To facilitate this, the groups will get the same process modeling assignment, but be instructed to use either a data-centric or activity-centric tool to work on their assignment. Furthermore, the assignments will be conducted all the way through to the deployment phase, to allow comparing results across various phases of the BPM lifecycle. The factors to be compared might be, for example, quality of the produced model, speed of development, quality of the produced software, and user feedback.

We would like to finally encourage the BPM community to continue the valuable research into data-centric business process management and use the DALEC framework presented in this paper to improve upon existing approaches by identifying areas with potential for improvement. Furthermore, we believe that the framework will help researchers when developing new data-centric approaches by highlighting the shortcomings of existing approaches, allowing them to build increasingly mature tools and concepts.

Acknowledgements This work has been partially supported by the projects FIRST (EU H2020-RISE 734599), VOICE (EU FP7 621137), Social Museum & Smart Tourism (Italian CTN01_00034_23154), and Resilience of Metropolitan Areas (ROMA) (Italian SCN_00064). This work is partially funded by the ZAFH Intralogistik, funded by the European Regional Development Fund and the Ministry of Science, Research and the Arts of Baden-Württemberg, Germany (F.No. 32-7545.24-17/3/1)

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

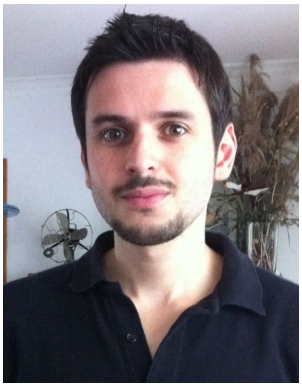
References

1. Andrews, K., Steinau, S., Reichert, M.: A runtime environment for object-aware processes. In: BPM Demo Session 2015 (BPMD), Co-located with the 13th International Conference on Business Process Management (BPM), pp. 6–10. CEUR Workshop Proceedings, CEUR-WS.org (2015)
2. Ayora, C., Torres, V., Weber, B., Reichert, M., Pelechano, V.: VIVACE: a framework for the systematic evaluation of variability support in process-aware information systems. Inf. Softw. Technol. **57**, 248–276 (2015)
3. Bagheri Hariri, B., Calvanese, D., de Giacomo, G., Deutsch, A., Montali, M.: Verification of relational data-centric dynamic systems with external services. In: 32nd Symposium on Principles of Database Systems (PODS), pp. 163–174. ACM, New York, NY, USA (2013)
4. Bagheri Hariri, B., Calvanese, D., Montali, M., Santoso, A., Solomakhin, D.: Verification of Semantically-enhanced Artifact Systems. LNCS, vol. 8274, pp. 600–607. Springer (2013)
5. Belardinelli, F., Lomuscio, A., Patrizi, F.: Verification of GSM-based artifact-centric systems through finite abstraction. In: 10th International Conference on Service-Oriented Computing (ICSOC). LNCS, vol. 7636, pp. 17–31. Springer (2012)
6. Bhattacharya, K., Hull, R., Su, J.: A data-centric design methodology for business processes. In: Handbook of Research on Business Process Modeling, pp. 503–531 (2009)
7. Boeker, M., Vach, W., Motschall, E.: Google scholar as replacement for systematic literature searches: good relative recall and precision are not enough. BMC Med. Res. Methodol. **13**(1), 131 (2013)
8. Calvanese, D., de Giacomo, G., Lembo, D., Montali, M., Santoso, A.: Semantically-governed data-aware processes. Knowledge-Intensive Business Processes, p. 21 (2012)
9. Calvanese, D., Montali, M., Patrizi, F., Rivkin, A.: Implementing data-centric dynamic systems over a relational DBMS. In: CEUR Workshop Proceedings, vol. 1378, pp. 209–212. CEUR-WS, New York, NY, USA (2015)
10. Cangialosi, P., de Giacomo, G., de Masellis, R., Rosati, R.: Conjunctive artifact-centric services. In: 8th International Conference on Service-Oriented Computing (ICSOC). LNCS, vol. 6470, pp. 318–333. Springer (2010)
11. Chao, T., Cohn, D., Flatgard, A., Hahn, S., Linehan, M., Nandi, P., Nigam, A., Pinel, F., Vergo, J., Wu, F.Y.: Artifact-based transformation of IBM global financing. In: 7th International Conference on Business Process Management (BPM), pp. 261–277. Springer (2009)
12. Chiao, C.M., Künzle, V., Reichert, M.: Integrated modeling of process- and data-centric software systems with PHIL harmonic flows. In: 1st IEEE International Workshop on Communicating Business Process and Software Models, pp. 1–10. IEEE Computer Society Press (2013)
13. Chiao, C.M., Künzle, V., Reichert, M.: Schema evolution in object and process-aware information systems: issues and challenges. In: 10th International Conference on Business Process Management (BPM), Workshops, Revised Papers. LNBIP, vol. 132, pp. 328–339. Springer (2013)
14. Chiao, C.M., Künzle, V., Reichert, M.: Towards schema evolution in object-aware process management systems. In: International Workshop on the Evolution of Information Systems and Their Design Methods (EMISA), pp. 101–115. Lecture Notes in Informatics (LNI), Koellen-Verlag (2014)
15. Damaggio, E., Hull, R., Vaculín, R.: On the equivalence of incremental and fixpoint semantics for business artifacts with guard-stage-milestone lifecycles. Inf. Syst. **38**(4), 561–584 (2013)
16. De Giacomo, G., Oriol, X., Estañol, M., Teniente, E.: Linking data and BPMN processes to achieve executable models. In: 29th International Conference on Advanced Information Systems Engineering, CAiSE 2017, Essen, Germany, June 12–16, 2017, Proceedings, pp. 612–628 (2017)
17. Deutsch, A., Li, Y., Vianu, V.: Verification of Hierarchical Artifact Systems. ArXiv e-prints (2016)
18. Di Ciccio, C., Marrella, A., Russo, A.: Knowledge-intensive processes: characteristics, requirements and analysis of contemporary approaches. J. Data Semant. **4**(1), 29–57 (2015)
19. Dumas, M.: On the convergence of data and process engineering. In: 15th International Conference on Advances in Databases and Information Systems (ADBIS), pp. 19–26. Springer (2011)
20. Dumas, M., van der Aalst, W.M.P., ter Hofstede, A.H.: Process Aware Information Systems: Bridging People and Software Through Process Technology. Wiley, Chichester (2005)
21. Durst, S., Runar Edvardsson, I.: Knowledge management in SMEs: a literature review. J. Knowl. Manag. **16**(6), 879–903 (2012)
22. Eckermann, O., Weidlich, M.: Flexible artifact-driven automation of product design processes. In: Enterprise, Business-Process and Information Systems Modeling. LNBIP, vol. 81, pp. 103–117. Springer (2011)
23. Eshuis, R., van Gorp, P.: Synthesizing object-centric models from business process models. In: Business Process Management Workshops. LNBIP, vol. 171, pp. 155–166. Springer (2014)
24. Gehanno, J.F., Rollin, L., Darmoni, S.: Is the coverage of google scholar enough to be used alone for systematic reviews. BMC Med. Inf. Decis. Mak. **13**(1), 7 (2013)
25. Haddar, N., Tmar, M., Gargouri, F.: A framework for data-driven workflow management: modeling, verification and execution. In: 24th International Conference on Database and Expert Systems Applications (DEXA). LNCS, vol. 8055, pp. 239–253. Springer (2013)
26. Haesen, R., de Rore, L., Goedertier, S., Snoeck, M., Lemahieu, W., Poelmans, S.: Stateless process enactment. In: 14th Conference on Pattern Languages of Programs (PLOP), p. 15:1. ACM, New York, NY, USA (2007)
27. Haesen, R., Snoeck, M., Lemahieu, W., Poelmans, S.: Existence dependency-based domain modeling for improving stateless process enactment. In: Congress on Services: I, pp. 515–521 (2009)
28. Harmon, P., Wolf, C.: State of Business Process Management. Business Process Trends (2016)
29. Havey, M.: Essential Business Process Modeling. O'Reilly Media, Inc., Sebastopol (2005)
30. Heath III, F.T., Boaz, D., Gupta, M., Vaculín, R., Sun, Y., Hull, R., Limonad, L.: Barcelona: a design and runtime environment for declarative artifact-centric BPM. In: 11th International Conference on Service-Oriented Computing (ICSOC), pp. 705–709. LNCS, Springer (2013)
31. Hill, J.B., Sinur, J., Flint, D., Melenovsky, M.J.: Gartner's Position on Business Process Management. Gartner Research 136533 (2006)
32. Hollingsworth, D.: Workflow Management Coalition: The Workflow Reference Model (1994)
33. Hull, R.: Artifact-centric business process models: brief survey of research results and challenges. In: On the Move to Meaningful Internet Systems: Confederated Int'l Conferences: CoopIS, DOA-SVI, and ODBASE. LNCS, vol. 5332, pp. 1152–1163. Springer (2008)
34. Hull, R., Damaggio, E., de Masellis, R., Fournier, F., Gupta, M., Heath III, F.T., Hobson, S., Linehan, M., Maradugu, S., Nigam, A., Sukaviriya, P.N., Vaculín, R.: Business artifacts with guard-stage-milestone lifecycles: managing artifact interactions with conditions and events. In: 5th ACM International Conference on Distributed Event-based Systems (DEBS), pp. 51–62. ACM, New York, NY, USA (2011)

35. Hull, R., Koschmider, A., Reijers, H.A., Wong, W.: Fresh approaches to business process modeling (Dagstuhl Seminar 16191). Dagstuhl Rep. **6**(5), 1–30 (2016)
36. Kitchenham, B.A.: Procedures for Performing Systematic Reviews. Keele, UK, Keele University, vol. 33, pp. 1–26 (2004)
37. Kossak, F., Illibauer, C., Geist, V., Kubovy, J., Natschläger, C., Ziebermayr, T., Kopetzky, T., Freudenthaler, B., Schewe, K.D.: A rigorous semantics for BPMN 2.0 process diagrams. In: A Rigorous Semantics for BPMN 2.0 Process Diagrams, pp. 29–152. Springer, Cham (2014)
38. Kumaran, S., Liu, R., Wu, F.Y.: On the duality of information-centric and activity-centric models of business processes. In: 20th International Conference on Advanced Information Systems Engineering (CAiSE). LNCS, vol. 5074, pp. 32–47. Springer (2008)
39. Künzle, V.: Object-Aware Process Management. Ph.D. Thesis, University of Ulm (2013)
40. Künzle, V., Reichert, M.: PHILharmonicFlows: towards a framework for object-aware process management. J. Softw. Maint. Evol. Res. Pract. **23**(4), 205–244 (2011)
41. Künzle, V., Weber, B., Reichert, M.: Object-aware business processes: fundamental requirements and their support in existing approaches. Int. J. Inf. Syst. Model. Des. **2**(2), 19–46 (2011)
42. Kurz, M.: Taming diversity: a distributed acm-based approach for cross-enterprise knowledge work. In: IEEE/WIC/ACM International Joint Conference on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), vol. **3**, pp. 87–91 (2013)
43. Kurz, M., Schmidt, W., Fleischmann, A., Lederer, M.: Leveraging CMMN for ACM: examining the applicability of a new OMG standard for adaptive case management. In: 7th International Conference on Subject-Oriented Business Process Management (S-BPM ONE), pp. 4:1–4:9. ACM, New York, NY, USA (2015)
44. Küster, J.M., Ryndina, K., Gall, H.: Generation of business process models for object life cycle compliance. In: 5th International Conference on Business Process Management (BPM). LNCS, vol. 4714, pp. 165–181. Springer (2007)
45. Liu, R., Bhattacharya, K., Wu, F.Y.: Modeling business contexture and behavior using business artifacts. In: 19th International Conference on Advanced Information Systems Engineering (CAiSE). LNCS, vol. 4495, pp. 324–339. Springer (2007)
46. London Bridge Group: Vectus Technical Architecture. London Bridge Group, Wellesbourne, Warwick, UK (2001)
47. Marrella, A., Mecella, M., Russo, A., Steinau, S., Andrews, K., Reichert, M.: Data in business process models, a preliminary empirical study (short paper). In: 8th IEEE International Conference on Service-Oriented Computing and Applications (SOCA), pp. 116–122 (2015)
48. Meyer, A., Pufahl, L., Fahland, D., Weske, M.: 11th International Conference on Business Process Management (BPM). LNCS, vol. 8094, pp. 171–186. Springer (2013)
49. Meyer, A., Smirnov, S., Weske, M.: Data in Business Processes. Universitätsverlag Potsdam (2011)
50. Müller, D., Reichert, M., Herbst, J.: Data-driven modeling and coordination of large process structures. In: 15th International Conference on Cooperative Information Systems (CoopIS), pp. 131–149. LNCS, Springer (2007)
51. Mutschler, B., Weber, B., Reichert, M.: Workflow management versus case handling: results from a controlled software experiment. In: ACM Symposium on Applied Computing (SAC), pp. 82–89. ACM, New York, NY, USA (2008)
52. Neumann, C.P., Lenz, R.: Alpha-flow: a document-based approach to inter-institutional process support in healthcare. In: Business Process Management Workshops. LNBIP, vol. 43, pp. 569–580. Springer (2010)
53. Ngai, E.W.T., Xiu, L., Chau, D.C.K.: Application of data mining techniques in customer relationship management: a literature review and classification. Expert Syst. Appl. **36**(2), 2592–2602 (2009)
54. Nigam, A., Caswell, N.S.: Business artifacts: an approach to operational specification. IBM Syst. J. **42**(3), 428–445 (2003)
55. Object Management Group: Case Management Model and Notation, Version 1.0 (2014)
56. Pallas Athena: Case Handling with FLOWer: Beyond Workflow. Pallas Athena BV, Apeldoorn, The Netherlands (2002)
57. Pesic, M., Schonenberg, H., van der Aalst, W.M.P.: DECLARE: full support for loosely-structured processes. In: 11th IEEE International Conference on Enterprise Distributed Object Computing (EDOC), p. 287 (2007)
58. Pryss, R., Tiedeken, J., Kreher, U., Reichert, M.: Towards flexible process support on mobile devices. In: Information Systems Evolution: CAiSE Forum 2010, pp. 150–165. Springer (2011)
59. Pufahl, L., Weske, M.: Batch activities in process modeling and execution. In: 11th International Conference on Service-Oriented Computing (ICSOC), pp. 283–297. Springer (2013)
60. Reichert, M.: Process and data: two sides of the same coin? In: On the Move to Meaningful Internet Systems: Confederated International Conferences: CoopIS, DOA-SVI, and ODBASE, pp. 2–19. Springer (2012)
61. Reichert, M., Rinderle, S., Dadam, P.: On the common support of workflow type and instance changes under correctness constraints. In: On the Move to Meaningful Internet Systems: Confederated International Conferences: CoopIS, DOA, and ODBASE, pp. 407–425. Springer (2003)
62. Reichert, M., Weber, B.: Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies. Springer, Berlin (2012)
63. Reijers, H.A., Rigter, J.H.M., van der Aalst, W.M.P.: The case handling case. Int. J. Coop. Inf. Syst. **12**(03), 365–391 (2003)
64. Reijers, H.A., Vanderfeesten, I., Plomp, M.G.A., Gorp, P., Fahland, D., Crommert, W.L.M., Garcia, H.D.D.: Evaluating data-centric process approaches: Does the human factor factor in? Software and Systems Modeling, pp. 1–14 (2016)
65. Russo, A., Mecella, M., Montali, M., Patrizi, F.: Towards a reference implementation for data centric dynamic systems. In: Business Process Management Workshops. LNBIP, vol. 171, pp. 141–154. Springer (2014)
66. Russo, A., Mecella, M., Patrizi, F., Montali, M.: Implementing and running data-centric dynamic systems. In: 6th IEEE Conference on International Service-Oriented Computing and Applications (SOCA), pp. 225–232. IEEE Computer Society Press (2013)
67. Ryndina, K., Küster, J.M., Gall, H.: A Tool for integrating object life cycle and business process modeling. In: 5th International Conference on Business Process Management (BPM), BPM Demonstration Program. LNCS, Springer (2007)
68. Software-Ley GmbH: COSA Activity Manager. Software Ley GmbH, Pullheim, Germany (2002)
69. Solomakhin, D., Montali, M., Tessaris, S., de Masellis, R.: Verification of artifact-centric systems: decidability and modeling issues. In: 11th International Conference on Service-Oriented Computing (ICSOC). LNCS, vol. 8274, pp. 252–266. Springer (2013)
70. Staffware: Staffware Case Handler—White Paper. Staffware PLC, Berkshire, UK (2000)
71. Steinau, S., Andrews, K., Reichert, M.: A modeling tool for PHILharmonicFlows objects and lifecycle processes. In: Proceedings of the BPM Demo Track and BPM Dissertation Award co-located with 15th International Conference on Business Process Modeling (BPM 2017), Barcelona, Spain, September 13, 2017 (2017)
72. Vaculín, R., Hull, R., Heath III, F.T., Cochran, C., Nigam, A., Sukaviriya, P.N.: Declarative business artifact-centric modeling of decision and knowledge-intensive business processes. In: 15th IEEE International Conference on Enterprise Distributed Object

Computing Conference (EDOC), pp. 151–160. Helsinki, Finland (2011)
73. van der Aalst, W.M.P.: Business process management: a comprehensive survey. ISRN Softw. Eng. **2013**, 37 (2013)
74. van der Aalst, W.M.P., Barthelmess, P., Ellis, C.A., Wainer, J.: Proclets: a framework for lightweight interacting workflow processes. Int. J. Coop. Inf. Syst. **10**(04), 443–481 (2001)
75. van der Aalst, W.M.P., Weske, M., Grünbauer, D.: Case handling: a new paradigm for business process support. Data Knowl. Eng. **53**(2), 129–162 (2005)
76. Vanderfeesten, I., Reijers, H.A., van der Aalst, W.M.P.: Product-based workflow support. Inf. Syst. **36**(2), 517–535 (2011)
77. Wahler, K., Küster, J.M.: Predicting coupling of object-centric business process implementations. In: 6th International Conference on Business Process Management (BPM). LNCS, vol. 5240, pp. 148–163. Springer (2008)
78. Weber, B., Reichert, M., Rinderle-Ma, S.: Change patterns and change support features—enhancing flexibility in process-aware information systems. Data Knowl. Eng. **66**(3), 438–466 (2008)
79. Weske, M.: Business process management architectures. In: Business Process Management: Concepts, Languages, Architectures, pp. 333–371. Springer (2012)
80. Westergaard, M., Slaats, T.: Mixing paradigms for more comprehensible models. In: 11th International Conference on Business Process Management (BPM). LNCS, vol. 8094, pp. 283–290. Springer (2013)
81. Xu, W., Su, J., Yan, Z., Yang, J., Zhang, L.: An artifact-centric approach to dynamic modification of workflow execution. In: On the Move to Meaningful Internet Systems: Confederated International Conferences: CoopIS, DOA-SVI, and ODBASE. LNCS, vol. 7044, pp. 256–273. Springer (2011)
82. Zhang, P., Wang, G., Liu, C.: A data-driven approach to internet-based business collaboration. In: 9th International Conference on Grid and Cooperative Computing (GCC), pp. 377–382 (2010)

Sebastian Steinau is a Ph.D. student at the Institute of Databases and Information Systems at Ulm University. His research interests include data-centric process management systems and the coordination of business processes. He is part of a project in which PHILharmonicFlows, a new data-centric process management system, is being developed. An essential part of the overall PHILharmonicFlows concept is the coordination of the interactions of multiple objects and their lifecycles.



Andrea Marrella is Research Fellow at Sapienza Università di Roma and Information Director of the ACM Journal of Data and Information Quality. His research interests include business process management, process mining, automated planning and reasoning about action in AI, human–computer interaction and cyber-security. He has published over 50 research papers on the above topics.

Kevin Andrews is a Ph.D. student at the Institute of Databases and Information Systems of Ulm University. He works on the PHILharmonicFlows project, with the goal of creating a data-centric business process management system. His research interests include data-centric process management systems, in particular applying the concepts of variability and ad hoc changes to data-centric processes.

Francesco Leotta got his M.Sc. in Engineering in Computer Science from Sapienza Università di Roma cum laude in 2010 and his Ph.D. in Engineering in Computer Science from Sapienza Università di Roma in 2014. His research interests include the application of machine learning and business process mining to smart spaces, indoor localization techniques and brain-computer interfaces.

Massimo Mecella is Associate Professor with Sapienza Università di Roma. His research focuses on service oriented computing, business process management, cyber-physical systems and Internet-of-Things, advanced interfaces and human-computer interaction. He published more than 150 research papers and chaired different conferences in the above areas.

Manfred Reichert is a full professor at Ulm University, where he is director of the Institute of Databases and Information Systems. His research interests include business process management, information systems, and mobile services. Manfred was PC Co-chair of the BPM'08, CoopIS'11 and EDOC'13 conferences. Furthermore, he served as General Chair of the BPM'09 and EDOC'14 conferences as well as the BPM'15 workshops. Recently, he co-authored a Springer book on process flexibility and obtained the BPM Test of Time Award at the BPM 2013 conference.