



HAL
open science

Data-awareness in business process: a systematic literature review

Ali Nour Eldin, Benjamin Dalmas, Walid Gaaloul, Jonathon Baudot

► **To cite this version:**

Ali Nour Eldin, Benjamin Dalmas, Walid Gaaloul, Jonathon Baudot. Data-awareness in business process: a systematic literature review. 2026. <hal-05480289>

HAL Id: hal-05480289

<https://hal.science/hal-05480289v1>

Preprint submitted on 27 Jan 2026

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Data-awareness in Business Process: A Systematic Literature Review

Ali Nour Eldin^{1,2*}, Benjamin Dalmas², Walid Gaaloul¹, and Jonathon Baudot²

¹Telecom SudParis, Institut Polytechnique de Paris, France,
email: {firstName}.{last_name}@telecom-sudparis.eu,

²Bonitasoft, France, emails:{firstName}.{last-name}@telecom-sudparis.eu

*Corresponding author: Ali Nour Eldin, E-mail: ali.nour_eldin@telecom-sudparis.eu

Abstract

Business Process Management (BPM) is a strategic development that enables application leaders to leverage digital transformation, enhancing business performance through the management, orchestration, and automation of end-to-end business processes. Both data and processes form critical elements of business operations, prompting Business Process Management Systems (BPMSs) to incorporate additional information that empowers users to automate their business process models by connecting data flows to respective tasks. Traditional business process modeling languages have focused primarily on the control-flow perspective, often neglecting the data-flow aspect. This oversight has given rise to data-centric approaches that regard data as a first-class citizen. In this paper, we present a systematic literature review on the integration of data and processes within process models, comprising an in-depth analysis of 18 primary studies. Particular emphasis is placed on the design phase of BPM, specifically the representation of data, behavior, and interaction, as well as the tooling associated with each approach.

Keywords: Business Process Model, Data-Aware Process Modeling, Data-Centric, Systematic Literature Review.

1. Introduction

The advent of the digital revolution has escalated the competitive landscape for organizations worldwide. To expedite goal attainment and augment performance, organizations increasingly depend on agile methodologies that facilitate rapid prototyping of innovative ideas. Among these methodologies,

Business Process Management (BPM) emerges as a pivotal development, enabling application leaders to harness digital transformation to enhance business performance through the management, orchestration, and automation of end-to-end business processes (Dumas et al. (2018)). Defined as a set of systematically arranged activities aimed at achieving a specific organizational output, business processes are crucial for the sustained success of enterprises. Process models, which offer a detailed depiction of these activities, are essential for understanding and optimizing business operations (Aguilar-Savén (2004)).

Business Process Management Systems (BPMSs) are instrumental in providing development environments that facilitate the creation of business systems through visual interfaces. These systems employ process modeling languages to delineate business processes and automate associated tasks, including data modification, scheduling, and the integration of data flows with external services (Sahay et al. (2023)). The automation of business processes via BPMSs is intended to minimize the costs and temporal expenditure associated with the implementation and maintenance of software systems, thereby catering to the needs of diverse user groups, both technical developers and non-technical stakeholders (Waszkowski (2019)).

However, traditional BPM approaches often segregate data management from process management, typically categorizing process modeling languages into two types: **activity-centric** and **data-centric**. Activity-centric approaches, such as Business Process Modeling and Notation (BPMN) (von Rosing et al. (2015)) and Petri Nets (Murata (1989)), prioritize task sequences and control flows but tend to marginalize the role of data, thereby creating a significant gap in achieving a holistic view of business operations.

BPMN, for example, is a standard that uses specific notations for defining business processes, which are understandable to both technical and non-technical users. The challenge, however, lies in the execution semantics, which must be uniformly and correctly implemented across different platforms, a process often hindered by the lack of standard enforcement and the complexity of the specifications (zur Muehlen and Recker (2013)).

In contrast, data-centric approaches recognize data as a "first-class citizen," integrating it thoroughly within the process models to ensure a more cohesive and functional BPM environment (van der Aalst et al. (2013), van der Aalst et al. (2005)). Despite the advantages of data-centric methodologies, many such approaches are overly focused on the initial modeling phase, neglecting the full lifecycle of BPM which includes implementation, execution, diagnosis, and optimization. This often results in models that are not only challenging to use but also lack intuitiveness and clarity, complicating adoption among both business users and IT specialists (Steinau et al. (2019)).

This paper systematically reviews the integration of processes and data within BPM, assessing the full lifecycle of these methodologies, including modeling, verification, implementation, execution, and optimization. The goal is to evaluate whether these integrative approaches provide effective enhancements to BPM practices.

The remainder of this paper is organized as follows: Section 2 delineates the research methodology employed during the literature review. The results are presented in Section 3. Finally, the paper concludes with Section 4.

2. Research methodologies

We conducted a Systematic Literature Review (SLR) to analyze how data and processes contribute to business process modeling, assessing their impacts on BPMSs and their integration capabilities. To date, there is one systematic review that focuses on existing data-centric approaches and their capabilities up until 2017 (Steinau et al. (2019)). To conduct this review, we followed a rigorous methodology for identifying, evaluating, and interpreting the available research. According to Kitchenham (Kitchenham (2004)), an SLR has three main phases: planning the review, conducting the review, and reporting the review. Each phase will be detailed in the following sub-sections.

2.1. First phase: Planning the review

This phase involves developing a review protocol in terms of objective, research questions, search strategy, inclusion criteria, and exclusion criteria.

Objective and Research questions.

The present research aims at exploring and investigating the contributions that integrate process and data into the process modeling language for an easy adaptation of BPMS. To achieve this, we analyze the selected publications using a set of carefully derived research questions (RQs), aimed at gaining a comprehensive understanding of data-awareness solutions within BPM:

(RQ1 - process & data) How are the process and data perspectives presented? Understanding how process and data perspectives are presented will clarify the extent to which current models integrate these elements, informing the development of more cohesive BPMSs.

(RQ2 - interaction) How is the interaction between the data and the processes represented? Exploring this interaction will highlight the effectiveness of current integration strategies and pinpoint areas for improvement.

(RQ3 - enactment) How does the proposed approach enact the process model? Investigating how models enact the process will shed light on the practical applicability of theoretical frameworks in real-world BPM environments.

(RQ4 - data privilege) Is there an authorization concept for restricting or granting access to the data? Assessing the authorization concepts used for data access will help evaluate the security and compliance aspects of BPM models.

(RQ5 - model verification) Is there any verification process for the data in terms of correctness criteria, such as the absence of deadlocks and livelocks? This question aims to determine the robustness of data verification processes, essential for ensuring the reliability and integrity of BPMSs.

(RQ6 - granularity) Is there assistance for controlling the granularity of the model? Does it enforce granularity levels? By examining model granularity, this research will explore the adaptability of BPM systems to different organizational needs and detail levels.

(RQ7 - tools) Is there a tool available that assists with modeling? Which stage of the process lifecycle does it support? Assessing the availability and functionality of tools supports understanding the maturity and user-friendliness of BPM solutions.

Search strategy. We carefully selected search terms that relate specifically to data-centric processes within the domain of business process management, avoiding terms with broader, unrelated applications. We have selected the same search string of (Steinau et al. (2019)): ("data-aware process" OR "data-driven process" OR "data-oriented process" OR "data-centric process" OR "artifact-centric process" OR "artifact-based process" OR "knowledge based process" OR "knowledge-driven process" OR "knowledge-intensive process") AND workflow AND business. We then queried the Google Scholar, IEEE explorer, Science Direct and ACM Digital databases. Subsequently, we apply the inclusion and exclusion criteria to determine the eligibility of each paper for inclusion in our study.

Inclusion and exclusion criteria. The review methodology also specifies the inclusion and exclusion criteria for each main study. To restrict the research, the criteria were carefully specified before doing the search. The following were the criteria:

- Only full and long publications in English from peer-reviewed journals and conferences, published after 2017, were considered.
- The publications that do not refer at least to one of the search chains in the title or the abstract were automatically excluded.
- All the publications between 2017 and 2020 should have at least 5 citations to be included.
- The publications proposing new or extension of an existing approach are included.

2.2. Second phase: Conducting the review

Search for studies. The search strategy defined above was applied in May 2024. This search yielded a total of 1088 publications¹ after removing the duplicated title of the whole the search engine.

Study selection. Here, we determine which publications met the inclusion criteria by analyzing their titles, abstracts and keywords. Initially, 44 publications referring to defined inclusion criteria were accepted. Therefore, only 18 publications were definitely selected

¹Raw search string results: <https://docs.google.com/spreadsheets/d/1M-V5R1I-0gm00Q3Zekc-JykcOjTe9rag>

after a full text reading. Each publication was analyzed to extract the relevant information regarding the eight research questions.

2.3. Third phase: Reporting the review

RQ1 explores how different BPM models represent both data and processes. This question is vital for understanding how to manage data alongside process flows. Findings from this question help highlight models that provide clear and comprehensive representations, which are crucial for effective data governance and process optimization. Moreover, RQ1 investigates how the data lifecycle is represented. In activity-centric approaches, the lifecycle of data structures is implicitly expressed through a lifecycle process. However, in data-centric approaches, the processing states of a single data object are described. If a data structure is not associated with a lifecycle process, the term 'data lifecycle' generally refers to the methods of data acquisition. RQ2 investigates the nature of interactions between data elements and process. Understanding these interactions is essential for assessing the ease of integrating or transforming models to become a functional part of BPMS. This question helps identify which BPM systems facilitate smoother integration and modification, thereby enhancing system flexibility and responsiveness to change. RQ3 explores the enactment mechanisms of these approaches. Unlike activity-centric methods, where enactment is primarily driven by control flow, these approaches are driven by data acquisition. RQ4 addresses the ability of a process model to distinguish between read-only and updatable data impacts, especially regarding data security and process reliability. RQ5 seeks to ensure that process models are verifiable for soundness, completeness, and termination, which is critical before the execution phase. It aims to analyze the integration of theoretical and practical models within a BPMS. RQ6 addresses process granularity, which refers to the level of detail or depth at which a process is defined or viewed. This question explores how granularity is managed within BPM systems. RQ7 examines the availability of tools that support various phases of the process lifecycle, assessing their applicability and maturity. For instance, we check whether there is support for modeling or monitoring processes through available tools.

3. Results

This section illustrates the results extracted from several approaches to answer the research questions mentioned in Section 2.1.

3.1. Process & Data Representation

This section presents the results of RQ1, focusing on the presentation of the process and data perspectives.

Table 1 provides an overview of how data and processes are presented in various approaches. The reviewed papers are divided into three categories based on their representation of process perspectives. The process perspectives can be presented by: (i) BPMN, (ii) Petri net, and (iii) a data-centric approach that treats the data perspective while neglecting or ignoring the process perspective. For instance, Combi et al. (2018), Estañol et al. (2022), Ghilardi et al. (2021), Giacomo et al. (2017), Haarmann et al. (2021), and Hahn et al. (2018) represent the process perspective using the standard BPMN. Other works Liu et al. (2022), Montali and Rivkin (2017), Polyvyanyy et al. (2019), and Wang et al. (2022) represent the process perspective using Petri nets, while other works focus on the data perspective or extend existing process and data models, like Ghilardi et al. (2020), which extends the Colored Petri Net.

Moreover, the data perspective is totally different from approach to other. Where each technique proposed and present the data in a specific presentation. Indeed the data lifecycle handle the relevant states in a data representation's life, from creation through final disposition and archiving. The data lifecycle can be presented implicitly within the process perspective, or explicitly by define the state of the data in each process steps. We choose 'data lifecycle' term to be generalize, which can be the object lifecycle, the presentation of lifecycle implicitly within the process perspective, the customization of state machine, and tokenization within Petri net.

DF-BPMN in (Nour Eldin et al. (2023)) represent the data as graphical notation within the activity, these data has types to represent process variable, user data, and database informations. Data-aware BPMN (Calvanese, Ghilardi, et al. (2019)) is a formal definition of BPMN enriched by data, where the authors proposed to define the process variables and database information formally. The authors in (Giacomo et al. (2017)) represent the process variable as an object. This object is a new class object called an artifact, presented using a UML class diagram. The process variables are data that are widely used during execution, and can be referred to as volatile data. The data lifecycle is not specified and is implicitly presented in the process perspective. The authors in (Combi et al. (2018)) focus on the database data by presenting it using a tabular format atop each activity, where this annotation represents operations on the database through the integration of BPMN and UML class diagrams. The data lifecycle is not specified

and is implicitly presented in the tabular presentation of each operation. In (Ghilardi et al. (2021)), the authors propose PDMML (Process and Data Modeling and Manipulation Language), which presents volatile (process/case variable) and persistent data (database data as catalog and repository). Similar to previous works, the data lifecycle is implicitly presented in this work. Looking at (Estañol et al. (2022) and Haarmann et al. (2021)), the data object is defined using a UML class diagram. Then, (Hahn et al. (2018)) presents data as a cross-partner data object that has a unique identifier and contains one or more data elements, each with a unique name. (Estañol et al. (2022) and Hahn et al. (2018)) also implicitly represent the data lifecycle. In (Haarmann et al. (2021)), the authors present the object lifecycle using state transition system.

Moreover, the works that present the process perspective in Petri nets also customize the data within their extensions. In (Montali and Rivkin (2017)), a new formal model balances these two pillars through the marriage of colored Petri nets and relational databases. They have a persistence layer that captures the relational database used to store data and the data carried by such tokens. Extended Data Petri Net (DPNE) Wang et al. (2022) integrates the data interaction and behavior of the entire process. It is based on Petri net with data operations and data Petri nets. Data-aware Workflow Net (DWF-net)Liu et al. (2022)extends the classical workflow net with data reading and writing semantics. Specifically, it is an extension of a Petri net with data reading and writing semantics. They define two kind of places, data place and logic place, to represent within the same model the control-flow and the data-flow. Information Systems Modeling Language (ISML)(Polyvyanyy et al. (2019))has three components: an information model to describe the domain, a Petri net with identifiers to describe dynamic processes, and a specification defining how the processes manipulate information. The lifecycle of these works is presented by the tokens within each Petri net, allowing tracking of each data element.

The rest of the works are not based on activity-centric approaches. For instance, Catalog-nets (CLog-nets) Ghilardi et al. (2020)are an extension of colored Petri nets (CPNs). The data are injected into tokens by extracting relevant values from the database. The data lifecycle is included in the CPNs by tokens, which allow tracking of each data item. In DAPHNE (Calvanese, Montali, et al. (2019)), the authors propose the dapSL language, a declarative language that is fully grounded in the SQL standard. It encapsulates process tasks into SQL-based, parameterized actions that update persistent data, possibly injecting values

Approach	Process perspective	Data perspective	Data lifecycle
Calvanese, Ghilardi, et al. (2019), Combi et al. (2018), Estañol et al. (2022), Ghilardi et al. (2021), Giacomo et al. (2017), Haarmann et al. (2021), Hahn et al. (2018), and Nour Eldin et al. (2023)	BPMN	Object	BPMN
Fritsch et al. (2023), Liu et al. (2022), Montali and Rivkin (2017), Polyvyanyy et al. (2019), and Wang et al. (2022)	Petri-net	Tokens	Petri net or Petri nets Extensions
Ghilardi et al. (2020)	-	CPN tokens	Colored Petri Nets
Steinau et al. (2018)	-	Object	Lifecycle: Custom
Calvanese, Montali, et al. (2019)	-	Tuples	Control flow
Gonzalez-Lopez et al. (2021) and Snoeck et al. (2021)	-	Data/Business object	Lifecycle: State Machines

Table 1: Overview of the process and data representations employed by the selected approaches.

obtained from external inputs (such as user forms, web services, and external applications). The data lifecycle depends on the control flow connected to this data flow system. In (Steinau et al. (2018)), the authors present a relational process instance structure that inherently supports many-to-many relationships between processes and considers cardinality constraints. In this work, an abstraction of data objects is illustrated without showing how the attributes are treated. The data lifecycle depends on the control flow connected to this data flow system. In (Gonzalez-Lopez et al. (2021)), the authors extend the fragment-based Case Management language (Hewelt and Weske (2016)). The data are presented as data objects, each data type holding a particular state in which it is consumed or produced by a process fragment. Finally, (Snoeck et al. (2021)) presents the data as business objects, where these business objects are defined by a UML class diagram. The last two works use a State Machine to illustrate the lifecycle of the objects.

3.2. Interaction between Processes and Data

This section addresses research question RQ2 (see section 2.1). Table 2 summarizes the interaction between process and data and determines whether they

are distinct entities.

The authors in (Estañol et al. (2022) and Giacomo et al. (2017)) use the Object Constraint Language (OCL) Cabot and Gogolla (2012), a textual language for defining queries over a UML schema. Using OCL allows the authors to relate each task to the corresponding elements in the UML class diagram representing the data. In (Combi et al. (2018)), after defining the tabular view for each activity, the interactions between the data are presented directly within the tabular view. This view contains tuples of relations, each consisting of a class set describing the necessary classes to access from the database, an association set representing the relationships between the selected classes, and access types detailing the CRUD operations (Create, Read, Update, Delete). It also specifies when these operations should be executed (at the start of the activity, at the end, or during execution), as well as the cardinality of the data. In DF-BPMN (Nour Eldin et al. (2023)), the authors propose a data-flow sequence that can connect different data types. In Data-aware BPMN (Calvanese, Ghilardi, et al. (2019)), the authors formally define all the data operations (CRUD operations) on top of each activity.

In Delta-BPMN (Ghilardi et al. (2021)), the authors

Approach	Interactions	Separation of behavior/interactions
Nour Eldin et al. (2023)	data-flow relation (graphical representation)	No
Calvanese, Ghilardi, et al. (2019)	precondition, effect	No
Estañol et al. (2022) and Giacomo et al. (2017)	OCL	No
Combi et al. (2018)	Activity View	No
Ghilardi et al. (2021)	precondition, effect, declare variable	No
Haarmann et al. (2021)	FCM and BPMN	Yes
Hahn et al. (2018)	Declarative, Cross-partner data-flow	No
Montali and Rivkin (2017)	Data logic	No
Wang et al. (2022)	Petri Net, Read and Write and Delete operations	No
Liu et al. (2022)	Read and write operations	No
Fritsch et al. (2023) and Polyvyanyy et al. (2019)	Perti Nets, CRUD	No
Ghilardi et al. (2020)	Colored Petri Nets	No
Calvanese, Montali, et al. (2019)	CRUD and external function call	No
Steinau et al. (2018)	Semantic Relationship	No
Gonzalez-Lopez et al. (2021)	CMMN	Yes
Snoeck et al. (2021)	Custom services	Yes

Table 2: Behavior and Interaction descriptions of the different approaches

extend BPMN to incorporate relations between different data types as properties for each activity. These properties declare user input as "var" in the key of the properties and use "precondition" for the conditions that must be executed before the activity begins. Finally, the "effect" describes the interaction between the variables and both volatile and persistent data, which also can be a SQL query. In (Haarmann et al. (2021)), the integration between the data and the process are the same with the standard BPMN, with a stronger focus on data objects. This data can help to defining the goal of a case, also to determine execution semantic of the case model. Cross-partner data flow (Hahn et al. (2018)) allows for the decoupling of data exchange from message exchange and from choreography control flow. It also supports the specification of transformation logic, selective queries, and mappings, e.g., to read or write only parts of a data element, cross-partner data objects,

or multiple data elements simultaneously.

In DB-net (Montali and Rivkin (2017)), the authors describe three layers: control-flow, data logic, and persistence. The data logic layer exposes a set of queries and a set of actions that the control layer can use to obtain data from and induce updates on the persistence layer. The authors of (Wang et al. (2022)) proposed adding three new relations (arcs) to the Petri net to represent the read, write, and delete operations. In DWF-net (Liu et al. (2022)), it has two kinds of flow relations, one is used to represent the control-flow, whereas the other is used to represent data-flow. The data-flow relations, to represent read and write operations on a specific data place.

In ISML (Polyvyanyy et al. (2019)), each transition within the Petri nets is linked to specific operations on the data elements defined in the information models. These operations are the CRUD operations. In CLog-net

(Ghilardi et al. (2020)), the authors use colored Petri nets for the interaction between the control-flow and the data-flow. The queries are defined and specified in terms of first-order logic extended with variable types. Within Daphne (Calvanese, Montali, et al. (2019)), the CRUD operations are defined within the function created on top of each activity that describes the data interaction with the database.

The authors in (Gonzalez-Lopez et al. (2021)) integrate the data and processes by creating detailed case models that adapt to the complexity of knowledge work. In (Snoeck et al. (2021)), data and process interaction occurs through a model-driven architecture where domain models provide a stable data foundation, and process models add dynamic, user-driven behaviors.

The separation of the behavior (control-flow) and data-flow can help in tracking each operation on the data. Only three works from the selected works track each operation.

3.3. Enactment, Data privilege, Verification, and Granularity

This section illustrates the results of research questions RQ3, RQ4, RQ5, and RQ6. Table 3 focuses on: (i) the enactment of the process model, including the data; (ii) data privilege, which refers to the permission access for data objects within the process model, which can help to ensure the ACID database operations, and add privilege one top the process model; (iii) verification of the model, there are different verification techniques, these technique allow to guaranty the enactment of the process model without any error (e.g. deadlocks, live-locks, termination, ...); and (iv) the level of granularity.

To put the model into action, the enactment phase is an important phase to simulate and obtain results. For instance, in (Ghilardi et al. (2020), Haarmann et al. (2021), Montali and Rivkin (2017), and Polyvyanyy et al. (2019)), the authors transfer the model to Colored Petri Nets (CPN) and then use the existing CPN simulations. In (Giacomo et al. (2017)), there is a transformation of BPMN to Petri nets, and OCL to Logic Derivation Rules to allow the model to be executable. In (Calvanese, Montali, et al. (2019) and Snoeck et al. (2021)), the authors provide dedicated APIs for integration with BPM engines and DBMS. In (Hahn et al. (2018)), the authors design TraDE, which is aimed at enhancing the modeling and execution of data-aware service choreographies by facilitating the decoupling of data flow from control flow through middleware that manages cross-partner data objects and enables dynamic data exchanges across service

interactions.

Moreover, few of the approaches support data access permissions. Three approaches (Calvanese, Ghilardi, et al. (2019) and Ghilardi et al. (2020, 2021)) proposed a new model for data access called a catalog data store. In this case, during the design phase, both read-only and updateable data specifications are clearly defined. Another study, (Estañol et al. (2022)), used $UCON_{ABC}$ security policies defined in OCL for data access.

Seven approaches implemented a verification technique. Some, such as those found in (Haarmann et al. (2021) and Montali and Rivkin (2017)), used formal definitions, while other approaches like those in (Calvanese, Ghilardi, et al. (2019) and Ghilardi et al. (2020, 2021)) encoded their models using Model Checker Modulo Theories (MCMT) (Ghilardi and Ranise (2010)), one of the most established model checkers for the verification of infinite-state dynamic systems. This checker helps verify the soundness, termination, and completeness of models. In (Estañol et al. (2022)), the authors transferred BPMN and UML class diagrams to BAUML, a framework that ensures the semantic correctness of the model. Meanwhile, in (Calvanese, Montali, et al. (2019) and Polyvyanyy et al. (2019)), the verification is based on state-space exploration for the logic verification of the model.

Finally, seven approaches enforce one level of granularity, particularly those that have extended the Petri net. However, six approaches choose not to enforce any restrictions regarding the level of process abstraction.

3.4. Tools

The use of tools for modeling and implementing processes is essential in the context of data-aware techniques. In addressing research question RQ7, we examined whether phases of the process lifecycle are supported by tools such as modeling or runtime environments. This ability to distinguish between lifecycle phases is crucial for a more accurate assessment of tool maturity.

Eight approaches (Calvanese, Montali, et al. (2019), Ghilardi et al. (2020), Giacomo et al. (2017), Haarmann et al. (2021), Hahn et al. (2018), Montali and Rivkin (2017), Polyvyanyy et al. (2019), and Snoeck et al. (2021)) provide support for 'Design', 'Implementation', and 'Execution & Monitoring' phases, while another approach (Fritsch et al. (2023) and Nour Eldin et al. (2023)) supports only the 'Design' phase of the business process lifecycle. It is immediately evident that data-aware techniques lack tool support for the final phase, namely 'Optimization'. The absence of support

Approach	Enactment	Data privilege	Verification	Granularity
Giacomo et al. (2017)	Petri net and Logic Derivation Rules	-	-	Unspecified
Ghilardi et al. (2020)	Colored Petri Net	read-only data	MCMT	One level
Haarmann et al. (2021)	Colored Petri Net	-	Formal semantics	Unspecified
Montali and Rivkin (2017)	Colored Petri Net	-	Formal semantics	One level
Polyvyanyy et al. (2019)	Colored Petri Net	-	State-space exploration	One level
Hahn et al. (2018)	TraDE	-	-	Unknown
Calvanese, Montali, et al. (2019)	APIs	-	State-space exploration	Unknown
Snoeck et al. (2021)	APIs	-	-	Two level
Calvanese, Ghilardi, et al. (2019) and Ghilardi et al. (2021)	-	read-only data	MCMT	Unspecified
Estañol et al. (2022)	-	$UCON_{ABC}$	BAUML framework	Unspecified
Liu et al. (2022), Steinau et al. (2018), and Wang et al. (2022)	-	-	-	One level
Combi et al. (2018), Fritsch et al. (2023), Gonzalez-Lopez et al. (2021), and Nour Eldin et al. (2023)	-	-	-	Unspecified

Table 3: Process enactment mechanisms, Data privilege , Verification and Granularity of the different approaches

in all approaches during the execution phase might indicate a general lack of maturity in data-aware techniques. Furthermore, eight methods are currently conceptual only, as no evidence of tool support has been found.

4. Discussion & Conclusion

Business Process Management Systems (BPMSs) commonly use BPMN to automate business processes. Integrating data and process models is increasingly necessary to provide a more holistic view of business processes and reduce implementation times. BPMN, however, has limitations in representing data-related concepts. As a result, researchers are focusing on integrating processes and data into a single modeling language.

In this Systematic Literature Review (SLR), we

presented several approaches to integrate data and process models within the same modeling languages. After analyzing these approaches, it became clear that most remain conceptual, posing challenges for seamless integration into a BPMS for enterprise use. Approaches still in the design phase should be integrated into BPMS to facilitate automation and utilization within enterprises. Nonetheless, a few promising approaches support both the implementation and execution phases.

Looking at the latest SLR that addresses data-centric approaches, only one approach extends BPMN, while all others focus on proposing new data-centric approaches. However, in this work, we find that the extensions of BPMN are increasing, indicating that there is a growing need for a model that balances data and processes. This is also necessary for vendors to integrate within their systems.

Additionally, the verification of BPMN with data

has become more challenging. Therefore, several approaches verify BPMN enriched by data, which highlights a limitation of BPMSs in detecting errors without simulation or theoretical transformation. This verification also becomes necessary for vendors to ensure the execution of a BPMN model without errors.

We recommend developing a modeling language that is user-friendly for businesses, designers, and developers. One effective strategy could be to follow the widely accepted BPMN standard used in BPMSs. This way, researchers could provide a robust, standard-based solution. The proposed language should be compatible with a BPMS or supported by a tool to meet enterprise needs.

Additionally, enterprises often enhance their existing BPMS environments by adding user permissions to integrate data on top of BPMN, and for monitoring purposes. As researchers, we must find ways to make BPMN easier for developers to reduce their workload. We suggest focusing on making BPMN more intuitive for developers and proposing a tool for designing the language that can be converted to any BPMS. The researchers' focus should be on the design phase, while enactment and execution could be managed by an open-source BPMS. Also, the data privileges supported by most BPMSs should not be overlooked. BPMN supports a holistic view of the control flow, making it a good starting point.

In conclusion, the escalating growth of information in organizations necessitates a research focus on integrating data and processes into BPM. In this review, we provide a comprehensive overview of data integration with processes based on eight research questions about approaches and frameworks. This analysis draws on 18 papers published between 2017 and May 2024. The findings of this review illuminate the current state of treating data as a first-class citizen in BPM.

References

- Aguilar-Savén, R. S. (2004). Business process modelling: Review and framework. *International Journal of production economics*, 90(2), 129–149.
- Cabot, J., & Gogolla, M. (2012). Object constraint language (OCL): A definitive guide. *Formal Methods for Model-Driven Engineering - 12th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM 2012. Advanced Lectures*, 7320, 58–90.
- Calvanese, D., Ghilardi, S., Gianola, A., Montali, M., & Rivkin, A. (2019). Formal modeling and smt-based parameterized verification of data-aware BPMN. *Business Process Management - 17th International Conference, BPM 2019*, 11675, 157–175.
- Calvanese, D., Montali, M., Patrizi, F., & Rivkin, A. (2019). Modeling and in-database management of relational, data-aware processes. *Advanced Information Systems Engineering - 31st International Conference, CAiSE 2019*, 11483, 328–345.
- Combi, C., Oliboni, B., Weske, M., & Zerbato, F. (2018). Conceptual modeling of processes and data: Connecting different perspectives. *Conceptual Modeling - 37th International Conference, ER 2018*, 11157, 236–250.
- Dumas, M., Rosa, M. L., Mendling, J., & Reijers, H. A. (2018). *Fundamentals of business process management, second edition*. Springer.
- Estañol, M., Varela-Vaca, Á. J., López, M. T. G., Teniente, E., & Gasca, R. M. (2022). Reasoning on the usage control security policies over data artifact business process models. *Comput. Sci. Inf. Syst.*, 19(2), 547–572.
- Fritsch, A., Schüler, S., Forell, M., & Oberweis, A. (2023). Modelling and execution of data-driven processes with json-nets. *Enterprise, Business-Process and Information Systems Modeling - 24th International Conference, BPMDS 2023, and 28th International Conference, EMMSAD 2023*, 479, 29–43.
- Ghilardi, S., Gianola, A., Montali, M., & Rivkin, A. (2020). Petri nets with parameterised data - modelling and verification. *Business Process Management - 18th International Conference, BPM 2020*, 12168, 55–74.
- Ghilardi, S., Gianola, A., Montali, M., & Rivkin, A. (2021). Delta-bpmn: A concrete language and verifier for data-aware BPMN. *Business Process Management - 19th International Conference, BPM 2021*, 12875, 179–196.
- Ghilardi, S., & Ranise, S. (2010). MCMT: A model checker modulo theories. *Automated Reasoning, 5th International Joint Conference, IJCAR 2010*, 6173, 22–29.
- Giacomo, G. D., Oriol, X., Estañol, M., & Teniente, E. (2017). Linking data and BPMN processes to achieve executable models. *Advanced Information Systems Engineering - 29th International Conference, CAiSE 2017*, 10253, 612–628.

- Gonzalez-Lopez, F., Pufahl, L., Munoz-Gama, J., Herskovic, V., & Sepúlveda, M. (2021). Case model landscapes: Toward an improved representation of knowledge-intensive processes using the fcm-language. *Softw. Syst. Model.*, 20(5), 1353–1377.
- Haarmann, S., Montali, M., & Weske, M. (2021). Refining case models using cardinality constraints. *Advanced Information Systems Engineering - 33rd International Conference, CAiSE 2021, 12751*, 296–310.
- Hahn, M., Breitenbücher, U., Kopp, O., & Leymann, F. (2018). Modeling and execution of data-aware choreographies: An overview. *Comput. Sci. Res. Dev.*, 33(3-4), 329–340.
- Hewelt, M., & Weske, M. (2016). A hybrid approach for flexible case modeling and execution. *Business Process Management Forum - BPM Forum 2016, 260*, 38–54.
- Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004), 1–26.
- Liu, C., Zeng, Q., Cheng, L., Duan, H., & Cheng, J. (2022). Measuring similarity for data-aware business processes. *IEEE Trans Autom. Sci. Eng.*, 19(2), 1070–1082.
- Montali, M., & Rivkin, A. (2017). Db-nets: On the marriage of colored petri nets and relational databases. *Trans. Petri Nets Other Model. Concurr.*, 12, 91–118.
- Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proc. IEEE*, 77(4), 541–580.
- Nour Eldin, A., Baudot, J., & Gaaloul, W. (2023). Zooming in for clarity: Towards low-code modeling for activity data flow. *Business Process Management Forum - BPM 2023 Forum, 490*, 267–282.
- Polyvyanyy, A., van der Werf, J. M. E. M., Overbeek, S., & Brouwers, R. (2019). Information systems modeling: Language, verification, and tool support. *Advanced Information Systems Engineering - 31st International Conference, 11483*, 194–212.
- Sahay, A., Ruscio, D. D., Iovino, L., & Pierantonio, A. (2023). Analyzing business process management capabilities of low-code development platforms. *Softw. Pract. Exp.*, 53(4), 1036–1060.
- Snoeck, M., Smedt, J. D., & Weerd, J. D. (2021). Supporting data-aware processes with MERODE. *Enterprise, Business-Process and Information Systems Modeling - 22nd International Conference, BPMDS 2021, and 26th International Conference, EMMSAD 2021, Held at CAiSE 2021, 421*, 131–146.
- Steinau, S., Andrews, K., & Reichert, M. (2018). The relational process structure. *Advanced Information Systems Engineering - 30th International Conference, CAiSE 2018, 10816*, 53–67.
- Steinau, S., Marrella, A., Andrews, K., Leotta, F., Mecella, M., & Reichert, M. (2019). DALEC: a framework for the systematic evaluation of data-centric approaches to process management software. *Softw. Syst. Model.*, 18(4), 2679–2716.
- van der Aalst, W. M. P., Stahl, C., & Westergaard, M. (2013). Strategies for modeling complex processes using colored petri nets. *Trans. Petri Nets Other Model. Concurr.*, 7, 6–55.
- van der Aalst, W. M. P., Weske, M., & Grünbauer, D. (2005). Case handling: A new paradigm for business process support. *Data Knowl. Eng.*, 53(2), 129–162.
- von Rosing, M., White, S., Cummins, F., & de Man, H. (2015). Business process model and notation - BPMN. *The complete business process modeling to bpm, volume 1* (pp. 429–453). Morgan Kaufmann/Elsevier.
- Wang, Y., Yu, W., Teng, P., Liu, G., & Xiang, D. (2022). A detection method for abnormal transactions in e-commerce based on extended data flow conformance checking. *Wirel. Commun. Mob. Comput.*, 2022, 4434714:1–4434714:14.
- Waszkowski, R. (2019). Low-code platform for automating business processes in manufacturing. *IFAC-PapersOnLine*, 52(10), 376–381.
- zur Muehlen, M., & Recker, J. (2013). How much language is enough? theoretical and practical use of the business process modeling notation. *Seminal contributions to information systems engineering, 25 years of caise* (pp. 429–443). Springer.