On the Declarative Paradigm in Hybrid Business Process Representations: A Conceptual Framework and a Systematic Literature Study

Amine Abbad Andaloussi[a,*], Andrea Burattin[a], Tijs Slaats[b], Ekkart Kindler[a], Barbara Weber[a,c]

[a] *Software and Process Engineering, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark*
[b] *Department of Computer Science, University of Copenhagen, 2100 København, Denmark*
[c] *Institute of Computer Science, University of St. Gallen, 9000 St. Gallen, Switzerland*

Abstract

Process modeling plays a central role in the development of today's process-aware information systems both on the management level (e.g., providing input for requirements elicitation and fostering communication) and on the enactment level (providing a blue-print for process execution and enabling simulation). The literature comprises a variety of process modeling approaches proposing different modeling languages (i.e., imperative and declarative languages) and different types of process artifact support (i.e., process models, textual process descriptions, and guided simulations). However, the use of an individual modeling language or a single type of process artifact is usually not enough to provide a clear and concise understanding of the process. To overcome this limitation, a set of so-called "hybrid" approaches combining languages and artifacts have been proposed, but no common grounds have been set to define and categorize them. This work aims at providing a fundamental understanding of these hybrid approaches by defining a unified terminology, providing a conceptual framework and proposing an overarching overview to identify and analyze them. Since no common terminology has been used in the literature, we combined existing concepts and ontologies to define a "Hybrid Business Process Representation" (HBPR). Afterward, we conducted a Systematic Literature Review (SLR) to identify and investigate the characteristics of HBPRs combining imperative and declarative languages or artifacts. The SLR resulted in 30 articles which were analyzed. The results indicate the presence of two distinct research lines and show common motivations driving the emergence of HBPRs, a limited maturity of existing approaches, and diverse application domains. Moreover, the results are synthesized into a taxonomy classifying different types of representations. Finally, the outcome of the study is used to provide a research agenda delineating the directions for future work.

Keywords:
Hybrid process model, Understandability of process models, Process flexibility, Declarative process modeling, Business process modeling

1. Introduction

In the development of today's Process-Aware Information Systems (PAIS), process modeling has become an important instrument to cope with the complexity of both the *management* and the *enactment* of business processes [1]. On the management level, process modeling provides input for requirements elicitation and allows concretizing business processes while ensuring a common understanding for both domain experts and IT specialists [2]. By deploying a variety of artifacts, process modeling provides a means for communication and collaborative design and enables benchmarking, optimization and process re-engineering [3, 4]. The impact of process modeling goes beyond the management level to cover also the enactment level. Process modeling provides a blue-print for process execution, which in turn, facilitates system support and enables process enactment [5]. Furthermore, the outcome of process modeling enables a wide range of model analysis and verification techniques and allows simulating the model behavior under different execution scenarios [4].

The literature proposes a variety of approaches to graphically represent business processes as a process model. These approaches deploy different modeling languages (e.g., BPMN [6], Petri nets [7] , Declare [8], DCR [9]) and different types of process artifacts (e.g., process models, textual descriptions, animations and guided simulations). Depending on the kind of behavior implied in the process specifications, a business process can be most concisely described using a language from the *imperative–declarative* paradigm spectrum [10, 11]. Imperative languages allow describing explicitly the exact course of actions governing the execution of the business process which often makes them understandable to both domain experts and IT specialists. The use of imperative languages is suitable to model business processes where the execution alternatives are explicitly described in the process specifications. However,

⋆Work supported by the Innovation Fund Denmark project *EcoKnow* (7050-00034A).

∗Corresponding author

Email addresses: amab@dtu.dk (Amine Abbad Andaloussi), andbur@dtu.dk (Andrea Burattin), slaats@di.ku.dk (Tijs Slaats), ekki@dtu.dk (Ekkart Kindler), barbara.weber@unisg.ch (Barbara Weber)

some process specifications tend to abstract from describing the different execution alternatives and rather define a set of constraints guiding the overall process. These specifications can be naturally modeled using declarative languages, which allow using constraints to describe flexible business processes concisely. This way, it becomes possible to overcome the rigidity imposed by imperative languages and describe highly dynamic environments [12].

Previous research has provided evidence for the existence of business processes comprising both rigid and flexible parts [13]. Hence, restricting the modeling of business processes to declarative or imperative languages would imply an unnecessary complexity when modeling the rigid or the flexible parts of the business process. Since declarative languages take a constraint-based approach to describe the control-flow of business processes, representing the rigid parts using a declarative language, would require a high amount of constraints to impose a very specific behavior. Likewise, using an imperative language to model the flexible parts would require specifying all the possible execution alternatives, which would most likely result into a "spaghetti-like" model. (cf. Section 2.4.1 for a concrete example highlighting this inconvenience). In order to enable the modeling of both rigid and flexible parts of business processes concisely, a set of so-called "hybrid" approaches has emerged in the literature. While some approaches address the limitations of declarative notations and propose hybrid languages to combine declarative and imperative languages, other approaches address the separation of concerns between imperative processes and business rules and propose hybrid languages and hybrid process artifacts combining imperative process models with declarative artifacts.

The proposed hybrid approaches have not only the potential for providing concise process representations, but they can be also used to address the notorious limitations of declarative notations associated with their understandability and maintainability [14, 15, 16]. One of the key challenges in that regard is the inability of users to cope with process models with too many constraints [14]. Considering the rich and complex semantics of declarative languages (e.g., Declare) and all the possible ways in which constraints can interact, the understandability of declarative process models gets quickly hampered when dealing with complex processes [14]. The cognitive dimensions framework [17, 18] provides a reasonable explanation to that. Indeed, the interpretation of declarative process models is associated with an increased mental effort as the user is required to keep track of the states of all interrelated constraints while striving to interpret a declarative model. This task gets more complicated, when considering indirect constraints between activities (or so-called "hidden dependencies" [16]). Therefore, it is necessary to interpret the model as a whole rather than specific constraints in isolation. Given the limited capacity of humans' working memory [19] and the small amount of items a human memory can hold (i.e., 7 ± 2) [20], the interpretation of such models becomes very difficult. Hidden dependencies are also among the issues affecting the maintainability of declarative process models. Due to the complex entanglement of constraints, it becomes hard to determine which constraints are affected by a change of the specifications and to check the consistency of new changes with existing constraints [15]. Hence, the maintainability of declarative process models becomes easily prone to misalignment between the process specifications and the actual process model. To overcome the understandability and maintainability limitations of declarative languages, and to offer better support for the human cognitive processes associated with the modeling and the maintenance of declarative process models, several hybrid process artifacts supporting declarative artifacts with imperative ones have been proposed in the literature. These approaches address several issues associated with the understandability of declarative languages such as the complex semantics of declarative languages and the implications of hidden dependencies on the comprehension of declarative process models. Using hybrid process artifacts, extending declarative process models with imperative artifacts, the literature proposes several approaches to clarify the semantics of declarative process models and to track the implications of hidden dependencies on the interplay between the model activities [21, 22]. With regards to the maintainability of declarative process models, hybrid process artifacts can be used to address several challenges rising due to the continuous change of specifications. For instance, hybrid process artifacts (e.g., guided simulations supporting declarative process models) can be used to check the consistency of declarative processes after introducing new constraints in the model, and to keep track of the hidden dependencies rising from altering the constraints in the model [23].

In the following, we use the terminology "Hybrid Business Process Representation" (HBPR) to refer to *(1)* hybrid languages combining existing declarative and imperative languages and *(2)* hybrid process artifacts combining declarative and imperative artifacts.

1.1. Problem Statement

Hybrid approaches cover a wide range of representations addressing different aspects of process modeling. Although, these approaches share similar characteristics, the authors in the literature deploy a mix of terms to designate them, thus no common terminology exists. In addition, the literature lacks the basic foundations needed to define HBPRs. Besides a handful of publications (e.g., [24, 25]) describing HBPRs in an adhoc context, no framework allowing to structure and discern the characteristics of HBPRs has been proposed yet. As a result, the term *"hybrid"* becomes ambiguous and is sometimes used inconsistently in the literature. Furthermore, while several HBPRs have been surveyed in the context of supporting data intensive processes through data-centric approaches [26], little has been done to study the existing hybrid approaches taking a control-flow perspective to look into the declarative paradigm in hybrid representations. In the process of identifying the HBPRs proposed in this context, the need for a unified terminology and a conceptual framework providing a clear distinction of the different HBPRs proposed in the literature becomes a must.

1.2. Contributions

In this paper, we propose a conceptual framework for process artifacts, provide a unified terminology for HBPRs, perform a Systematic Literature Review (SLR) to investigate the existing HBPRs with a declarative language or artifact, and suggest an agenda for future research. Our contributions can be described as follows:

- C1: Propose a conceptual framework to discern the interactions between the different concepts defining a process artifact (cf. Section 2).
- C2: Instantiate the proposed conceptual framework to provide a unified terminology allowing to conceive the different types of HBPRs (cf. Section 3).
- C3: Perform an SLR to scrutinize HBPRs and organize them into a comprehensive taxonomy (cf. Sections 4 and 5). The study will cover hybrid languages combining declarative and imperative languages and hybrid process artifacts combining imperative process models with declarative process artifacts to present business processes concisely. Additionally, the study will focus on hybrid process artifacts extending declarative process models with imperative artifacts to overcome the challenges of declarative modeling languages.
- C4: Delineate a research agenda for future research (cf. Section 6).

Considering the lack of a unified terminology and a clear conceptual framework allowing to define HBPRs (cf. Section 1.1), Contributions C1 and C2 can be generalized to any type of HBPRs, while C3 and C4 focus on hybrid approaches taking a control-flow perspective to look into the declarative paradigm in hybrid representations.

1.3. Overview and Paper Structure

This section provides an overview on the different concepts discussed throughout this study. The aim is to familiarize the reader with the important notions and outline the structure of the paper. Section 2 discusses three important concepts i.e., *business process*, *language* and *process artifact*. A Business process is concretized as a process artifact using a language. A comprehensive definition of a business process and its underlying core *concepts* and *aspects* is presented in Section 2.1, while a set of relevant language characteristics (i.e., *syntax*, *semantics* and *language paradigm*) is defined and discussed in Section 2.2. These concepts provide the building blocks for a framework defining the general scope of a process artifact (cf. Section 2.3).

The proposed framework is instantiated in Section 3 to denominate the two types of HBPRs i.e., *hybrid languages* and *hybrid process artifacts*. As briefly outlined in the beginning of this section, a hybrid language allows expressing a process artifact using a combination of languages (usually from the imperative-declarative paradigm spectrum, cf. Section 2.2.2), whereas a hybrid process artifact allows concretizing a business process using more than one process artifact. This distinction provides a unified terminology (previously mixed in the literature) allowing to designate HBPRs consistently. Once established, a literature search is conducted following the research method presented in Section 4. The findings are scrutinized in Section 5, where the existing HBPRs are analyzed and compared on different levels. The results of the analysis come to support but also to enrich the proposed conceptual framework through a taxonomy discerning the characteristics of both hybrid languages and hybrid process artifacts. In Section 6, the main findings of this work are discussed and a research agenda is presented to guide the direction for the upcoming research. Last but not least, the threats to validity are discussed in Section 7, before concluding the paper in Section 8.

2. Conceptual Framework

This section introduces a conceptual framework defining the general scope for a process artifact. Following the existing terminology used within the BPM field, Sections 2.1 and 2.2 present the concepts associated with business processes and languages respectively. The interactions between these concepts are explained in Section 2.3, where the process artifact framework is presented. Finally, the different types of process artifacts are illustrated in Section 2.4.

2.1. Business Process

A *business process* is defined as *"a set of activities that are performed in coordination in an organizational and technical environment. These activities jointly realize a business goal."* [27]. The way a business process operates in the real world is captured by a process modeler as a set of *abstractions*, each emphasizing a given portion of reality. These abstractions are used to compose a subjective perception of the real world in the form of a mental model [28]. A *mental model* incorporates all the abstractions captured by the modeler about the way the business process operates in the real world. The shape of the mental model is affected by the *concepts* acquired by the modeler from the different ontologies proposed within the BPM field [28]. These concepts help the modeler to aggregate and structure the abstractions about the business process domain more efficiently [29]. Previous research [3] has classified these concepts into *(a) core concepts* which refer to the set of concepts defining the core elements of a business process i.e., process and tasks which instantiate into cases and activities [3] and *(b) aspects* which provide different lenses to look at the business process. The BPM literature discusses three main aspects: the *control aspect*, *organization aspect* and *information aspect*. In addition, several other aspects can be captured from a business process, for instance the *assignment aspect*, *security aspect* and *transaction aspect* [3].

This study focuses on the *control aspect* (also called *control-flow*), which is regarded as the most salient aspect in the literature [3]. *The control-flow represents information about the order of the activities or the constraints for their execution* [12].

2.2. *Language*

This section presents the characteristics of a language in terms of syntax and semantics (Section 2.2.1) and paradigm (Section 2.2.2).

2.2.1. *Language Syntax and Semantics*

A *language* is used to represent the business process. It is seen by Morris [30] as *"a system of interconnected signs, has a syntactical structure of such a sort that among its permissible sign combinations some can function as statements, and sign vehicles of such a sort that they can be common to a number of interpreters"*. A language can be a *natural language* (e.g., English) or an *artificial language* (e.g., programming language or conceptual modeling language). Among the features categorizing a language, Morris introduced *syntax* and *semantics*. Syntax is defined as *"the formal relation of signs to one another"* [30] i.e., the relations between finite meaningful elements of the language which allow deriving grammatically correct expressions. Semantics is defined as *"the relation of signs to real world entities they represent"* [30] i.e., the mapping between the language elements and the real world entities which allows conveying meaning [31].

Languages differ in terms of their *formality*, both regarding the syntax and the semantics. The syntax can be evaluated in terms of its grammatical structure and the completeness of its vocabulary, while the semantics can be evaluated in relation to the extent to which the semantic domain of the language is known [32]. Thereby, both syntax and semantics can be categorized as being *informal*, *semi-formal*, or *formal*.

2.2.2. *Language Paradigm*

The *language paradigm* can be seen as the style in which the language is written. Languages can be differentiated according to the *imperative–declarative* paradigm spectrum. This paradigm takes origins from the field of computer programming. As specified by Winograd [33], imperative programming is based on the idea that the *"the knowledge of a subject is intimately bound with the procedures for its use"*. In other words, imperative programming aims at specifying explicitly the set of commands leading to an output. Conversely, declarative programming as defined by Lloyd [34] aims at *"stating what is to be computed, but not necessarily how it is to be computed"*. Simply put, declarative programming aims at specifying the requirements to be achieved and letting the system determine the way to achieve them.

Roy and Haridi [35] use the notion of *state* to discriminate the two paradigms from a technical perspective. They defined a state as *"a sequence of values in time that contains the intermediate results of a desired computation"*. According to the authors a state can be either *implicit* or *explicit*. An implicit state is a state which neither the computational model nor the program are aware of, so it exists just in the mind of the programmer [35]. Conversely, an *explicit* state is a state which can be explicitly traced in the computational model and by the programmer. Following this distinction, declarative languages encode states *implicitly*, whereas imperative languages encode states *explicitly*. For instance, consider the two implementations of the factorial function shown in Figure 1. In the Prolog code (read as "the factorial of *Number* is *Factorial* if *Number*>0 and *Number1* is *Number*-1 and the factorial of *Number1* is *Factorial1* and *Factorial* is *Number**Factorial1*"), the notion of state is implicit. Indeed, although it would be possible for the programmer to trace down the sequence of states following a certain input value during the execution, there is no explicit predicate in the code keeping track of the computation result (i.e., state) after each recursive call. Alternatively, in the imperative Java implementation of the factorial function, the variable *value* is used in the computation model to explicitly keep track of the computation result after each iteration.

Listing (1) A program computing the factorial of *n* in Prolog

```
factorial(0,1).
factorial(Number, Factorial) :-
    Number > 0,
    Number1 is Number - 1,
    factorial(Number1,Factorial1),
    Factorial is Number * Factorial1.
```

Listing (2) A program computing the factorial of *n* in Java

```
public static int fact(int n)  {
    int value = 1;
    for(int i=n; i>0; i--)
        value *= i;
    return value ;
}
```

Figure 1: Declarative and imperative implementations of the factorial function in Prolog and Java.

The explicitness and the implicitness of states is bound to the representation layer of languages. This is because, at the execution layer, all languages are executed as a set of deterministic procedures which can be represented as states and transitions. This fact allows for a more fine-grained distinction between imperative and declarative languages. Hereby, we define imperative languages as *languages where states are explicit in both representation and execution layers*, and declarative languages as *languages where states are implicit in the representation layer and explicit in the execution layer*.

The same notion of *state* can be used when comparing imperative and declarative modeling languages. At the representation layer, imperative languages as defined by Pesic [14] allow modeling processes where *"all execution alternatives are explicitly specified"*, which in turn, enable representing the different states of a process explicitly. When modeled graphically, imperative languages provide a continuous trajectory (i.e., a sequence of states and transitions) allowing to reach any possible outcome allowed by the model [36]. The use of imperative languages implies the modeling of all possible courses of actions, which is only possible with a complete and well-detailed knowledge about all the alternative paths a business process execution might undergo. However, this is not always possible as the execution path of some business processes might depend on specifications that are only available at run-time and might also be unique to each process instance [37]. Alternatively, *declarative languages* as defined by Pesic [14] are language allowing to express models where *"constraints implicitly specify* [the] *execution alternatives as all alternatives that satisfy the constraints"*. In other words, declarative languages use constraints to describe the overall interplay of actions without explicitly describing the sequence of states and transitions leading to each particular outcome. Herein, states and transitions are implicit at the representation layer of the language and are only constructed at the execution layer, where the predicates and the formulas are interpreted [36]. This characteristic gives the capability to represent highly dynamic business processes concisely without having to explicitly specify the path of each single possible process execution.

Imperative and declarative languages allow representing different types of behaviors (i.e., forbidden, common and exceptional behaviors) [8], as shown in Figure 2, imperative languages are suitable to describe the common behavior, whereas declarative languages extend to both common and exceptional behaviors.

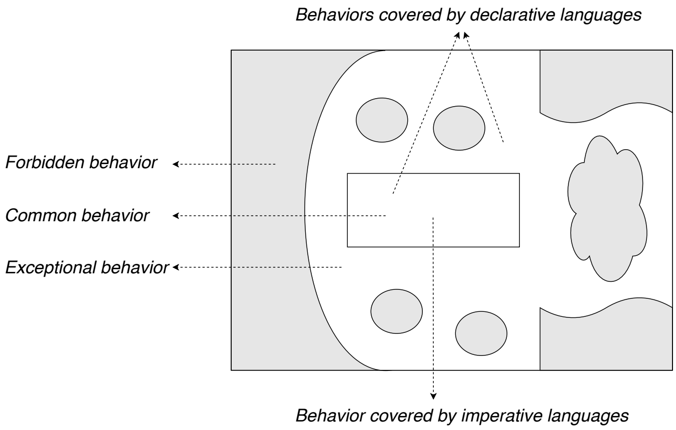


Figure 2: The different behaviors of a business process and the suitability of imperative and declarative languages to cover these behaviors. Adapted from [8].

Figure 3 depicts a three-dimensional framework describing the common languages used to model the control-flow aspect of business processes in terms of formality (both of syntax and semantics) and language paradigm. All these languages describe the control-flow as the order between the different process activities. In the figure, languages sharing more or less the same level of formality and language paradigm are grouped together. For example, Declare, DCR (Dynamic Conditional Response), CMMN (Case Management Model and Notation) [38] and XTT2 (Extended Tabular Tree version 2) [39] have a formal syntax, formal semantics and belong to the declarative language paradigm. Another group of languages comprises Petri nets, BPMN (Business Process Modeling Notation) and YAWL (Yet Another Workflow Language) [40] which all share a formal syntax, formal semantics and belong to the imperative language paradigm. Other languages with distinct level of formality and language paradigm are depicted individually. For instance, R2ML[1] (REWERSE Rule Markup Language) is characterized by a formal syntax, semi-formal semantics and belongs to the declarative language paradigm. Conversely, the original EPC (Event-driven Process Chains) language [41] has a formal syntax, informal semantics and belongs to the imperative language paradigm, whereas SBVR (Semantics of Business Vocabulary and Business Rules) [42] has a semi-formal syntax, semi-formal semantics and belongs to the declarative language paradigm.

Natural language in turn has an informal syntax and informal semantics. However, depending on the used grammatical structure and deployed language vocabulary, it can serve for expressing both declarative and imperative process specifications (cf. Figures 5a and 5b – the first part describes the interplay of actions using constraints, thus it is written in a declarative style, whereas the second part describes the explicit courses of actions, thus it is written in an imperative style). This is exactly why natural language is divided into *Imperative Natural Language* (I-NL) and *Declarative Natural Language* (D-NL). Although natural language has been used for a long time to describe business processes (e.g., paper-based documentations, regulatory documents), it has not been deployed in the BPM literature as a single artifact to represent a business process. However, with the emergence of hybrid process artifacts, natural language has been often combined with other imperative and declarative process representations. (e.g., [43, 44, 45, 46]).

2.3. Process Artifact

This section introduces the process artifact framework. Figure 4 combines the different pieces discussed in Sections 2.1 and 2.2 to illustrate our conceptual framework. The emergence of this framework can be seen as the result of putting together the existing concepts, which have been so far discussed in isolation in the literature. The relationships between the business process, the process artifact and the conceptualization entities are derived from the work of Axenath et al. [3]. The interactions between the process artifact, the mental model and the modeling concepts (represented within the conceptualization entity) are extracted from work of Soffer et al. [28] and the PhD thesis of Zugal [29]. Finally, the role of the language in bridging the gap between conceptualization and process artifacts is inspired by the ontological foundations proposed in Guizzardi's PhD thesis [32].

A process artifact is *the concretization of a business process using a language*. It is an *external* representation describing the way a business process operates in the real world in a formal or informal way [3]. It reflects the modeler's mental model which is an *internal* representation of the business process [28]. The core concepts and the different aspects (cf. Section 2.1) introduced within the BPM field constitute the conceptualization

[1]See http://www.macs.hw.ac.uk/bisel/rewerse/I1/oxygen.informatik.tu-cottbus.de/rewerse-i1%40q%3dr2ml.html

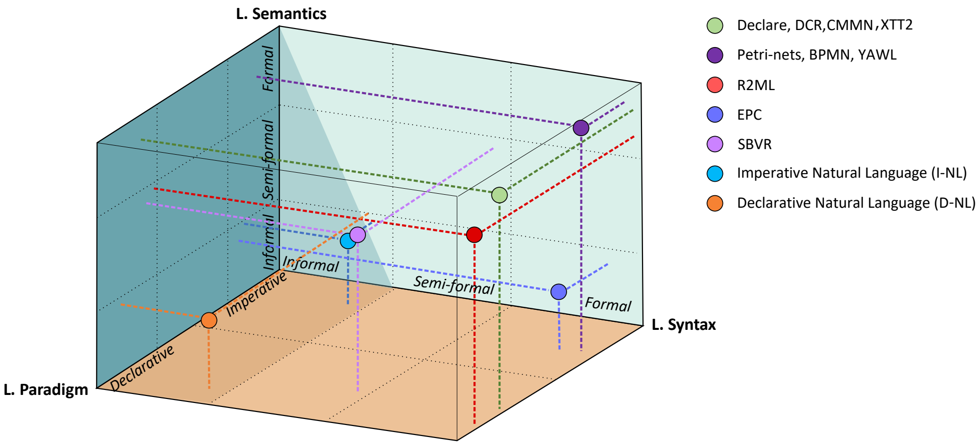


Figure 3: Categorization of some languages according to the formality of their syntax, the formality of their semantics and their language paradigm.

entity, which, in turn, allows structuring both the mental model and the process artifact. The former is structured by providing a schema supporting the aggregation of knowledge about the business process more efficiently [29, 28]. The latter is structured through the concepts and ontologies governing the modeling of business processes [3]. The language has a central role within the process artifact framework. Indeed, by choosing an appropriate syntax, semantics and language paradigm, a language allows expressing a process artifact. In addition, the language enables expressing the different notions of the conceptualization entity [32], which offers a means to transfer the BPM knowledge and make it attainable to the modeler.

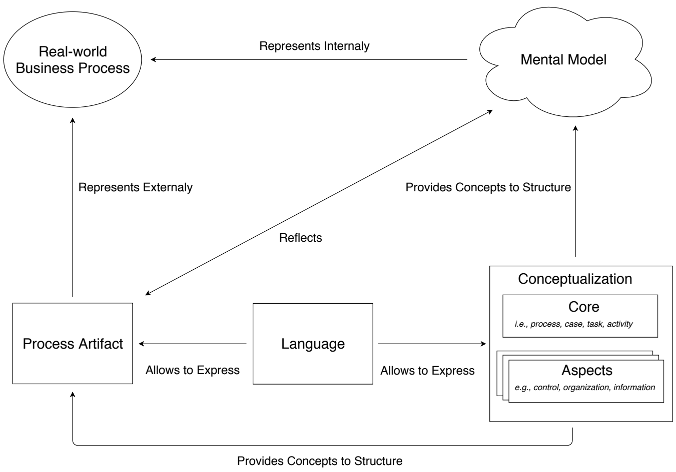


Figure 4: The process artifact framework.

The language provides the process artifact with a set of *inherent features* (i.e., syntax, semantics, paradigm, cf. Section 2.2). For instance, one can say that a process artifact is described in an imperative language characterized by a formal syntax and formal semantics (e.g., Petri nets). Besides the inherent features, a process artifact has a *visual feature*. Building upon the process visual representations in [47], a process artifact can be *static*, *dynamic*, or *interactive* (cf. Section 2.4 for examples). A static process artifact is characterized by a visual representation that remains static over time (i.e., textual process description, process model). A dynamic process artifact is characterized by an animated visual representation (i.e., dynamic over time) replaying previous executions of a business process (e.g., replay of event log traces). An interactive process artifact is characterized by an animated and interactive visual representation that changes depending on the way the user interacts with it (i.e., a guided simulation of a business process).

2.4. *Examples of Process Artifacts*

This section illustrates examples of the different process artifacts introduced in Section 2.3. Section 2.4.1 illustrates static artifacts. Section 2.4.2 illustrates dynamic artifacts. Section 2.4.3 illustrates interactive artifacts.

2.4.1. *Static Artifacts*

Textual process descriptions and process models are both examples of static artifacts, which means that their visual representation does not change over over time (compared to dynamic and interactive artifacts). Figure 5 shows two fragments describing the process of editing and handling a project proposal. Although both fragments are written in natural language, the interactions between the process activities are expressed differently. Indeed, while Fragment 1 describes the general interplay of actions in a loosely-coupled manner (using a D-NL), Fragment 2 specifies the exact course of actions with no room for flexible behavior (using an I-NL, e.g., *"Note that all decisions are final and cannot be reversed"*). This example illustrates a practical scenario where process specifications comprise both flexible and rigid requirements.

The two process models in Figure 6 describe the editing part of the project proposal process (cf. Fragment 1) using imperative (i.e, BPMN) and declarative (i.e, DCR[2]) languages respectively. Although, both models accommodate the same behavior, it is clear that the BPMN model in Figure 6a contains many

[2] The semantics of the DCR relations are summarized in `https://wiki.dcrgraphs.net/connection/`.

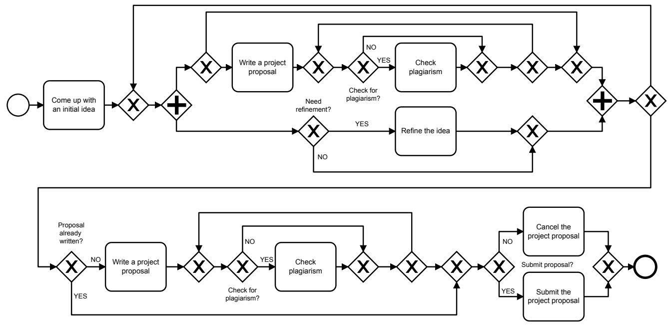
Fragment 1:
The process of writing a project proposal starts when the author comes up with an initial idea. Afterwards, it is possible to write a project proposal and to refine the idea at any time. After having written the project proposal it becomes possible to check for plagiarism. It is possible to cancel the proposal if it turns out that the idea is infeasible. Otherwise, as soon as the project proposal is described sufficiently well, it is possible for the author to submit the proposal. Note that a proposal can be submitted only once.

(a) Declarative process description in D-NL, adapted from [12, 48].

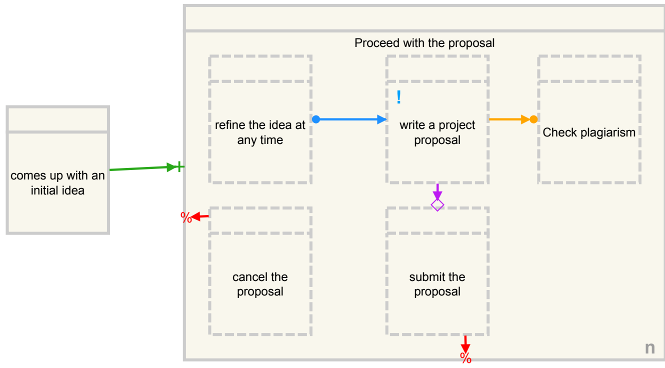
Fragment 2:
When a project proposal is received, a funding officer performs an initial screening of the proposal to check its compliance with the funding requirements of the institute. In case the proposal is not compliant, it is directly rejected. Otherwise, if the proposal complies with the given requirements, the funding officer provides an initial review of the proposal and sends it together with the initial proposal to a competent committee. The committee evaluates the proposal and based on their decision, the project proposal is either approved for funding or rejected. Note that all decisions are final and cannot be reversed.

(b) Imperative process description in I-NL.

Figure 5: Two process fragments describing different parts of the project proposal process.

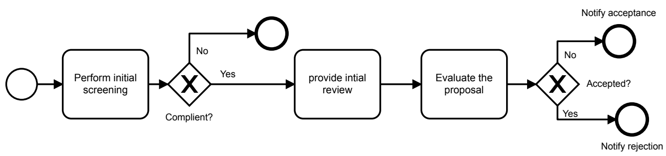


(a) Fragment 1 modeled imperatively.

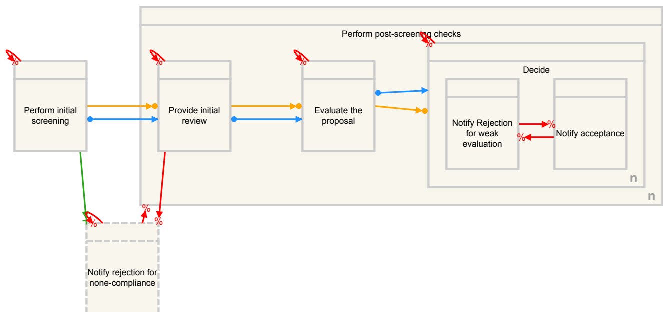


(b) Fragment 1 modeled declaratively.

Figure 6: Comparing the process in Fragment 1 when modeled imperatively using BPMN, and declaratively using DCR. The semantics of the DCR relations are summarized online[2].



(a) Fragment 2 modeled imperatively.



(b) Fragment 2 modeled declaratively.

Figure 7: Comparing the process in Fragment 2 when modeled imperatively using BPMN, and declaratively using DCR. The semantics of the DCR relations are summarized online[2].

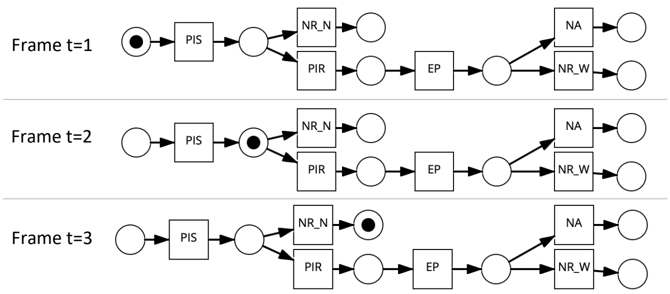


Figure 8: The frames of a Petri nets animation replaying a single event log trace. The letters in the transitions correspond to the initials of the activities extracted from Fragment 2 (e.g., NR_N: Notify rejection for none-compliance).

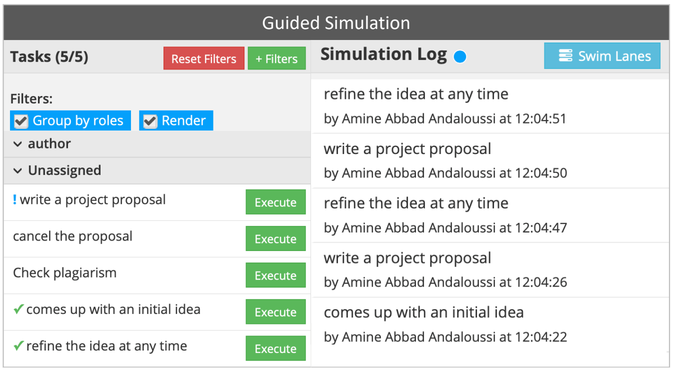


Figure 9: Example of a guided simulation corresponding to Fragment 1.

more elements than the DCR model in Figure 6b, which results in a spaghetti-like process model, making it more visually-complex and thus, hard to understand and maintain. Hereby, it becomes evident that imperative languages are not the ideal candidates for modeling flexible processes. Similarly, the handling part of the project proposal process (cf. Fragment 2) is described using imperative and declarative languages in Figure 7. Since this part of the process is rather rigid, describing it using a declarative language (cf. Figure 7b) would imply extra constraints to restrict the process behavior, which in turn result in a visually-complex process model, making its understandability and maintainability difficult. Alternatively, the use of an imperative language can provide a more concise process model (cf. Figure 7a).

2.4.2. *Dynamic Artifacts*

Dynamic artifacts provide an animated visual representation allowing to perceive how an existing process instance evolve overtime. Figure 8 illustrates the frames of the Petri nets token animation replying a single execution trace. The trace contains the execution of a process model implementing Fragment 2. The Petri nets token animation allows replaying the executed process instances, which in turn provides a visualization of the actual executions of the business process [49]. The choice of Petri nets to illustrate dynamic artifacts is motivated by the concept of token replay [49]. Any other modeling language with a similar concept can be used to express a dynamic artifact.

2.4.3. *Interactive Artifacts*

Interactive process artifacts provide a dynamic visual representation allowing the user to test the different courses of actions allowed by the process model. The example depicted in Figure 9 illustrates a guided simulation of a business process. The process artifact depicted in the figure is an instance of DCR Graphs, however, the same guided simulation could be provided by instantiating any other language. This artifact allows performing a guided simulation based on the user input, which in turn allows to test the possible execution scenarios and to perceive the allowed behavior at any stage of the simulation.

3. Hybrid Business Process Representations

This section provides a unified terminology allowing to conceive the different types of HBPRs. As mentioned in Section 1, the definitions presented in this section are general enough to cover all HBPRs. Sections 3.1 and 3.2 instantiates the conceptual framework presented in Section 2 in order to define the scope of hybrid languages and hybrid process artifacts respectively and to highlight the properties allowing to denominate both of them. Finally, Section 3.3 provides a generic definition of a HBPR.

3.1. *Hybrid Languages*

A hybrid language combines existing languages at the level of their syntax, semantics and language paradigm. Considering the process artifact framework depicted in Figure 4, the language entity can be instantiated into a hybrid language which can be represented as any combination of languages from the imperative and declarative paradigm spectrum. The composition of a hybrid language is restrained by the ability to support the syntax, semantics and language paradigm allowed by all its composing languages. In other words, a hybrid language should remain consistent even when only the vocabulary of a single language is used [50]. This feature allows active users of a composing language to progressively adapt to the new hybrid language without having to acquire it from scratch [24].

The use of hybrid languages brings a number of advantages. Firstly, by combining languages from the imperative–declarative paradigm, hybrid languages allow overcoming the limitations of individual languages and maintaining the balance between understandability and flexibility (e.g., [51]). Secondly, hybrid languages allow delivering an adequate language capable of representing business processes more concisely and precisely (e.g., [52]). Finally, hybrid languages enable the modeling of both rigid and flexible parts of a business process using the same language (e.g., [8]).

3.2. *Hybrid Process Artifacts*

A hybrid process artifact combines a set of interrelated process artifacts describing the same business process. Figure 10

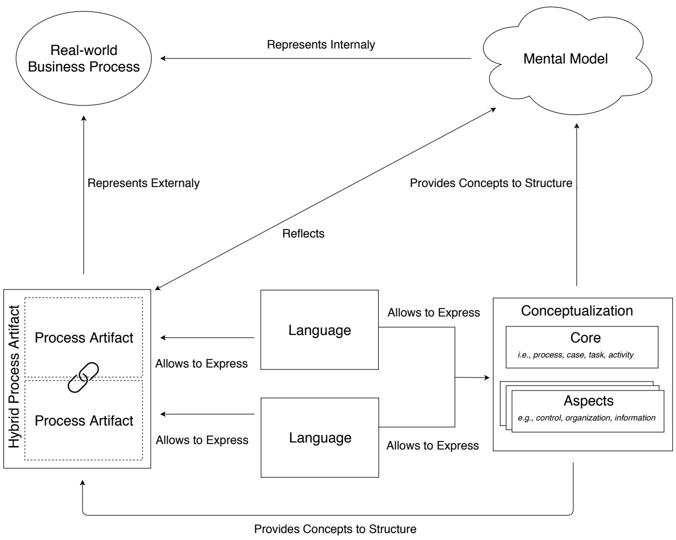


Figure 10: The scope of a hybrid process artifact defined based on the process artifact framework.

illustrates an instance of the process artifact framework capturing the scope of a hybrid process artifact. Here, a set of languages (represented in the figure as separated entities) is used to describe different *but interrelated* process artifacts. The overlap between artifacts is an important characteristic discerning hybrid process artifacts from *multi-perspective* process models (e.g., [53]). While a multi-perspective process model describes each business process aspect in a detached artifact, a hybrid process artifact overlaps in describing the business process aspects, which in turn provides parallel visual representations where equivalent information can be extracted easily.

Hybrid process artifacts have been proposed in the literature to address the separation of concerns between imperative business processes and business rules. Moreover, hybrid process artifacts have been used to improve the understandability of process models as they provide hybrid visual representations allowing to clarify the semantics of the model and to extract equivalent information easily (e.g., [45, 21, 2]). Furthermore, hybrid process artifacts have the potential to improve the maintainability of declarative process models by providing concrete means to track the hidden dependencies (between the activities) introduced due to the entanglement of constraints in the model. For instance, by extending a declarative process model with a guided simulation, it becomes possible to define the desired and prohibited behaviors (through test cases) and to constantly check them with the process model during a maintainability task (e.g., [23]). In addition, hybrid process artifacts can offer an alternative communication channels allowing to alternate between different levels of abstraction and support the knowledge transfer during the Process of Process Modeling (PPM) by combining formal and informal artifacts, which in turn, ensure a seamless communication between domain experts and IT specialists (e.g., [43, 54]).

3.3. Hybrid Business Process Representation

Following the characteristics of hybrid languages and hybrid process artifacts (cf. Sections 3.1 and 3.2 respectively), *a HBPR is defined as a collection of interrelated languages or process artifacts defining overlapping aspects and parts of the same business process.*

4. Literature Search Method

This SLR aims at identifying the existing HBPRs taking a control-flow perspective to look into the declarative paradigm. In addition, it focuses on providing a fundamental understanding of the proposed techniques through a transparent and reproducible approach. The research method deployed in this SLR follows the methodology proposed by Kitchenham [55] and lines up with the guidelines suggested by Budgen and Brereton [56] and Webster and Watson [57].

This section describes the search protocol adopted to conduct this SLR (cf. Figure 11). As a first step, the research problem is investigated by the authors and then formulated as a set of research questions (cf. Section 4.1). Then, following the recommendations of modeling experts (i.e., academics with several years of experience within the BPM field), a pilot search is conducted on the bibliography of notable authors within the field. (cf. Section 4.2). The outcome of this step allowed gathering the most common keywords and refining a comprehensive search string that covers the relevant literature (cf. Section 4.3). In addition, the most common publication venues are identified (cf. Section 4.4) and a set of inclusion and exclusion criteria is defined to filter the search results and select the most relevant articles (cf. Section 4.5). In the following step, the main literature search is performed and then the resulting articles are scrutinized (cf. Section 4.6). Afterwards a forward search and a backward search are performed (cf. Section 4.7). Finally, all the articles are read and relevant data are extracted according to a predefined scheme (cf. Section 4.8).

4.1. Research Questions

The research questions addressed in this study are the result of a series of meetings where the authors discussed the research problem (cf. Section 1.1) and the objectives of this study (cf. Section 1.2). In order to obtain a clear understanding about the HBPRs taking a control-flow perspective to look into the declarative paradigm, it is important to identify and investigate their distribution over time, type (i.e., journal, conference, book chapters) and venues. Therefore, the first research question is formulated as follows:

RQ1: *What publications about HBPRs taking a control-flow perspective to look into the declarative paradigm exist?*

This question is divided into the following sub questions:

RQ1.1: *How are these publications distributed over time?*

RQ1.2: *How are these publications distributed over publication type (i.e., journal, conference)?*

RQ1.3: *How are these publications distributed over publication venues?*

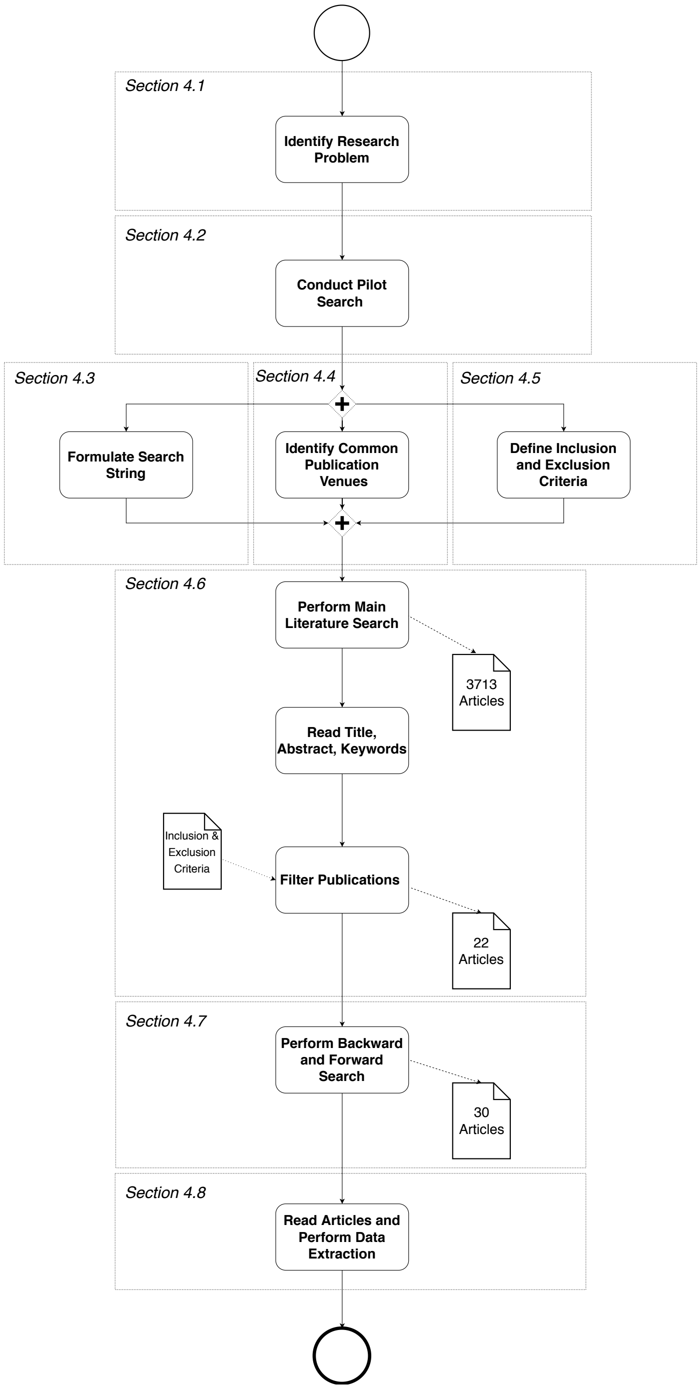


Figure 11: Summary of the protocol deployed to conduct this SLR (in BPMN language).

HBPRs have been deployed in several contexts. Therefore, it is possible that the proposed approaches have emerged within different research lines. The second research question investigates this aspect by addressing the following question:

RQ2: *What are the different research lines where the identified HBPRs were proposed?*

Once the research lines are identified, it is important to scrutinize the motivations behind the existing HBPRs in order to have a clear understanding about the process modeling issues which can be addressed using the proposed HBPRs. To this end, the third research question is formulated as follows:

RQ3: *What are the motivations driving the emergence of the identified HBPRs?*

The identification of the languages and the artifacts used to compose the existing HBPRs allows discerning the ones which were commonly deployed to model the imperative and the declarative process specifications. Hereafter, the fourth research question is formulated as follows:

RQ4: *Which languages and artifacts are combined in the identified HBPRs?*

Among the key contributions aimed by this work is a descriptive taxonomy allowing to categorize the existing HBPRs based on their common inherent and visual features. The fifth research question addresses this contribution as follows:

RQ5: *How can we categorize the identified HBPRs into a descriptive taxonomy?*

The maturity of the proposed hybrid approaches is another important aspect to investigate in order to evaluate the robustness of the proposed HBPRs. In that respect, it is necessary to investigate the extent to which the existing HBPRs have been formalized, and whether they have been implemented and evaluated empirically. These 3 aspects are addressed by the sixth research question as follows:

RQ6: *How mature are the identified HBPRs in terms of formalization, availability of implementation and empirical evaluations?*

Last but not least, the identification of the different application domains where HBPRs have been used allows illustrating the different fields where the use of HBPRs could be beneficial in practice. With this regard, the seventh research question is formulated as follows:

RQ7: *In what application domains can the deployment of the identified HBPRs be beneficial?*

4.2. Pilot Search

Prior to the main literature search, a pilot search has been conducted on the bibliography of notable authors following the recommendations of modeling experts. As a result, the following articles were considered as reference: [46, 58, 50, 24, 16, 59, 21, 60]. These articles use different terminologies. Westergaard and Slaats [50] and Slaats et al. [24] use the terms, "*hybrid model*", "*hybrid process*" and "*mixing paradigms*" to describe HBPRs that combine languages from the imperative–declarative paradigm spectrum. Lu et al. [58] use the term "*flexible workflow*" in the context of an HBPR combining predefined parts with loosely coupled parts of a business process. Wang et al. [46] use the term "*integrated modeling*" referring to a HBPR that combines business rules with business process models. The authors in [16, 21, 59, 60] propose approaches that combine multiple artifacts. Although no clear hybrid terminology was mentioned, the proposed representations can be seen as HBPRs in a way that they combine interrelated artifacts. Therefore, it was necessary to extend the search to cover similar publications (incorporating a combination of declarative representations with other types of representations) where no hybrid terminology was mentioned.

4.3. Search String

Following the keywords extracted from the pilot search (cf. Section 4.2), the search string can be composed from the product of two sets of keywords: *(1)* keywords emphasizing the mixed nature of the proposed representations i.e., *hybrid*, *mixing*, *flexible* and *integrated*. *(2)* keywords emphasizing the concept of a business process i.e., *workflow*, *process*, *model* and *paradigm*. Additionally, as some articles do not use explicit terminologies to designate HBPRs, the keywords *declarative*, *constraint* and *rule-based* were added to Set *1*. These keywords allow covering all subsets of declarative representations including those extending declarative languages and artifacts with other types of representations (e.g., [16]). The use of these keywords leads to more false positive matches, but it helps covering a wider spectrum of the literature. In addition, the false positive matches are filtered-out using the inclusion and exclusion criteria introduced in Section 4.5 and the manual inspection of articles.

As some keywords might be used in different forms (e.g., appending suffixes), all keywords were transformed to their base form, then a wildcard character (i.e., asterisk *) was appended to each one of them to broaden the search by looking for all words starting with the same letters (e.g. *mix* → mix, mixed, mixing*, etc.). Consequently, the following keywords were derived: *hybrid**, *mix**, *flexib**, *integrat**, *declar**, *constraint**, *rule**, *workflow**, *model**, *process** and *paradigm**. To interlink the search keywords, the *"OR"* logical operator was used. Indeed, during the study retrieval process, some literature search engines were unable to provide accurate results using complex search queries (i.e., search strings combining *"OR"*, *"AND"*, *"NOT"* operators). Thus, we have opted for a simple search string to provide a unified search string and maximize the hit rate across all search engines. The final search string is formulated as follows:

> hybrid* workflow* OR hybrid* model* OR hybrid* process* OR hybrid* paradigm* OR mix* workflow* OR mix* model* OR mix* process* OR mix* paradigm* OR flexib* workflow* OR flexib* model* OR flexib* process* OR flexib* paradigm* OR integrat* workflow* OR integrat* model* OR integrat* process* OR integrat* paradigm* OR declar* workflow* OR declar* model* OR declar* process* OR declar* paradigm* OR constraint* workflow* OR constraint* model* OR constraint* process* OR constraint* paradigm* OR rule* workflow* OR rule* model* OR rule* process* OR rule* paradigm*

4.4. Publication Venues

The notion of hybrid representations is widely deployed in several engineering fields. Therefore, conducting a general string look-up would lead to a huge amount of false positive matches. Once again, the recommendations of modeling experts were used to select the most popular data sources as well as the most prominent publication venues. Namely, the following data sources have been covered: Springer Link, IEE Explore Digital Library, ACM Digital Library, Science Direct, and Wiley Inter Science. As some data sources do not enable automated search, the Crossref[3] API was used. Within these data sources, the following journals have been covered: Decision support Systems (DSS), Information Systems (IS), Business & Information Systems Engineering (BISE), Software and

[3]Crossref is Digital Object Identifier (DOI) registration agency indexing publications identified with a DOI from different data sources. See `http://crossref.org/`

Systems Modeling (SoSyM). In addition, the following conference venues have been considered: International Conference on Software and System Processes (ICSSP), Enterprise Distributed Object Computing (EDOC), Business Information Systems (BIS), Business Process Management (BPM), Business Process Modeling, Development and Support (BPMDS), Conference on Advanced information Systems Engineering (CAiSE), Conference on Conceptual Modeling (ER), Fundamental Approaches to Software Engineering (FASE), Formal Methods (FM), Integrated Formal Methods (IFM) and On the Move to Meaningful Internet Systems (OTM). Note that the proceedings of the forums and the workshops organized during each conference were also covered by the search.

4.5. Inclusion and Exclusion Criteria

In order to frame the search and to filter-out false positive matches, a set of inclusion and exclusion criteria has been defined.

4.5.1. Inclusion Criteria

A study is relevant if the following criteria apply:

- IC1: The study emphasizes the modeling of a HBPR.
- IC2: The study proposes a HBPR that includes at least one declarative language or artifact.
- IC3: The proposed HBPR focuses on the control-flow aspect (as defined in Section 2.1).

4.5.2. Exclusion Criteria

A study is excluded in case one of the following criteria apply:

- EC1: The study does not have the main focus on the modeling of HBPRs. (e.g., This excludes studies mining HBPRs i.e., [61, 62, 63].)
- EC2: The study is not published in English.

4.6. Main Literature Search and Selection Process

The main literature search yielded a considerable amount of matching articles. The look-up in the search engines covered the meta-data of the articles i.e., title, abstract and keywords. In total 3713 articles were found. The high number of article retrieved from the search engines is typical for systematic literature reviews (e.g., [64]). In addition, since no unified terminology about HBPRs exists, the search string was formulated to be over-fitting for the purpose to cover a wide range of literature.

The selection process was performed by the corresponding author of this paper who followed a systematic approach to filter the literature articles based on a set of inclusion and exclusion criteria (cf. Section 4.5), which has been formulated and agreed by all the co-authors. Furthermore, in order to reduce any potential bias while selecting articles, the selection process has been constantly checked by the co-authors and borderline papers were discussed before deciding on their inclusion. Prior to the selection process, the meta-data of these articles were organized in a spreadsheet. During the selection process, the title of each retrieved article was scanned first in order to determine its relevance to the literature review (according to the inclusion and exclusion criteria). If the title is prominent, then, the abstract and keywords are inspected to further determine the article relevance. In case, the relevance remains doubtful, the article is fully read before being discussed internally by the co-authors. A spreadsheet with all the found studies and the decision on inclusion and exclusion is available online at `https://doi.org/10.5281/zenodo.3516661`.

As result, 22 relevant articles were selected from the main literature search. The selection also includes articles from the pilot search (cf. Section 4.2). The articles selected in the main literature search are the following: [46, 24, 52, 51, 45, 50, 59, 65, 58, 8, 37, 66, 25, 67, 68, 69, 70, 22, 16, 21, 60, 71] (the articles meta-data are presented after adding the backward search and forward search results, cf. Table 1). In the next step, a backward and forward search are conducted on these articles to gather additional relevant studies.

4.7. Backward and Forward Searches

To cover a wider range of the relevant literature, the initial search process has been expanded with a backward search and forward search. Section 4.7.1 shows the results of the backward search, Section 4.7.2 shows the results of the forward search.

4.7.1. Backward Search

The backward search examines the references cited in the literature to learn more about the foundation of the knowledge in question. In the context of our SLR, the backward search has covered the publications selected from the main literature search. During this process, we came across 4 new publications addressing our research subject. The following publications were then appended to the results of the main literature search: [72, 73, 74, 75].

4.7.2. Forward Search

The forward search examines new articles citing the literature and provides a follow-up on the development of the knowledge in question. In the context of our SLR, Google Scholar[4] was used to conduct the forward search on the publications selected from the main literature search. As a result, 4 new publications were retrieved and appended to the results of the main literature search: [76, 77, 78, 79].

4.8. Data Extraction

This section describes the data extraction process and lists the attributes used to answer each of the research questions introduced in Section 4.1. The data extraction process consists of extracting information relevant to the SLR according to a predefined data extraction scheme. The attributes depicted in Figure 12 summarize the scheme used to organize the data.

The extended article meta-data, including the title, authors, keywords, abstract, references, publication year, publication

[4] See `https://scholar.google.com`

type (i.e., journal or conference) and publication venue, are used to answer RQ1 by first enabling the identification of existing literature and then describing their time distribution, types, and venues. For this analysis, descriptive statistics (i.e., count and distributions in percentage) are used (cf. Section 5.1). To Answer RQ2, all the articles are read and labeled subjectively according to their research line. At the analysis, a backward search is performed on the articles of the final study retrieval list. During this process, an automated tool[5], is used to extract and filter the references cited by each article in order to retain only the citations referring to articles from the final study retrieval list. Afterwards using a graph visualization, the articles are represented as nodes, and the references are represented as directed edges between referring and referred articles. The size of each node is defined based on the number of incoming edges (i.e., the number of cites by the other articles in the graph). As we suspect that the articles belong to different research lines, a color is assigned to each node based on the label of its research line (cf. Section 5.2). For RQ3, the motivations behind each proposed approach are extracted, then used in the analysis phase to discern the different motivations driving the emergence of new HBPRs and to organize them into different categories (cf. Section 5.3). RQ4 is answered by looking at the combinations of languages and artifacts seen in the literature. In this regard, the artifacts used to construct each HBPR (together with their languages) are extracted, then both their inherent and visual features are scrutinized in order to discern their hybrid properties. (cf. Section 5.4). The data extracted in RQ4 is further investigated to answer RQ5 (cf. Section 5.5). To answer RQ6, data about formalization, availability of implementation and availability of evaluation are inferred. On that matter, the articles are classified as having a mathematical formalization, a meta-model formalization or not being formalized at all. Concerning the implementation, in case an article provides an implementation, information about tool name, type (i.e., prototype, plugin or commercial product), parent framework and reference are extracted. Then, for the evaluation, information about the number (#) of participants, evaluation reference, evaluation type (i.e., quantitative, qualitative or both), research aspects, instruments, measurements, and outcome are extracted. The data about formalization, implementation and evaluation are investigated during the analysis to denote the maturity of the proposed approaches (cf. Section 5.6). Finally, to answer RQ7, information about the different application domains used to exemplify the approaches proposed in the literature are extracted, grouped and presented at the analysis (cf. Section 5.7).

5. Analysis of Findings

This section provides an overarching analysis of the literature. Section 5.1 identifies the existing HBPRs and provides descriptive statistics (i.e., time distribution of publications, publication venues, and publication types) emphasizing the general findings obtained from the SLR search. Section 5.2 identifies the existing research lines that propose HBPRs. Section 5.3 highlights the motivations supporting the proposed representations. Section 5.4 distinguishes the different combinations of languages and artifacts proposed in the literature, Section 5.5 introduces a new taxonomy to categorize existing HBPRs, Section 5.6 investigates their maturity in terms of formalization, availability of implementation and empirical evaluation, and finally Section 5.7 reports the different application domains where the proposed HBPRs can be deployed.

5.1. Literature Search Findings

This section reports the results for RQ1. Table 1 shows the final study retrieval list including the title, the venue, the year and the authors of the selected articles. Overall, 30 articles were identified following the search method introduced in Section 4.

As depicted in Figure 13a, the articles addressing HBPRs have emerged since 2001 (answering RQ1.1). The time distribution of publications shows that 2011 and 2016 were the years with the highest number of publications addressing the topic (4 articles each year). By comparing the time distribution between the last two decades (cf. Figure 13b), one can notice that 77% of the articles were published between 2009 and 2018 compared to only 23% between 2001 and 2008. This shows an increasing tendency of articles proposing HBPRs over the last two decades. This tendency is also visible from the trend line depicted in Figure 13a.

Figure 13c shows the distribution of publications over publication type (answering RQ1.2). The search results show that 73% of articles appeared in conference proceedings, 23% appeared in journal proceedings, whereas only 4% where published as book chapters.

Finally, the selected articles were published in different venues (answering RQ1.3). As shown in Figure 13d, besides the initial publication venues considered for the main search, new venues have been covered during the backward and forward search. The distribution of the publication venues shows that EDOC, BPM and CAiSE take the lead with the largest proportions (i.e., 20%, 17%, and 10% respectively) and gather 47% of publications.

5.2. Research Lines

This section discusses the different research lines where HBPRs were introduced (answering RQ2). In the process of examining the articles retained in the final study retrieval list (cf. Table 1), a label was subjectively assigned to each article depending on its corresponding research line (cf. Section 4.8). The results of the labeling indicate that HBPRs have evolved in different contexts. More specifically, we have identified two main research lines within the BPM field. In the first research line (RL1), the authors proposed extending and supporting declarative languages, whereas in the second research line (RL2), the authors proposed integrating business rules with business processes. Section 5.2.1 validates the subjective labeling used to identify the research lines, and then Sections 5.2.2 and 5.2.3 present respectively the articles identified within these research lines.

[5]CERMINE is the tool used to extract references from the articles. Source code available at `https://github.com/CeON/CERMINE`

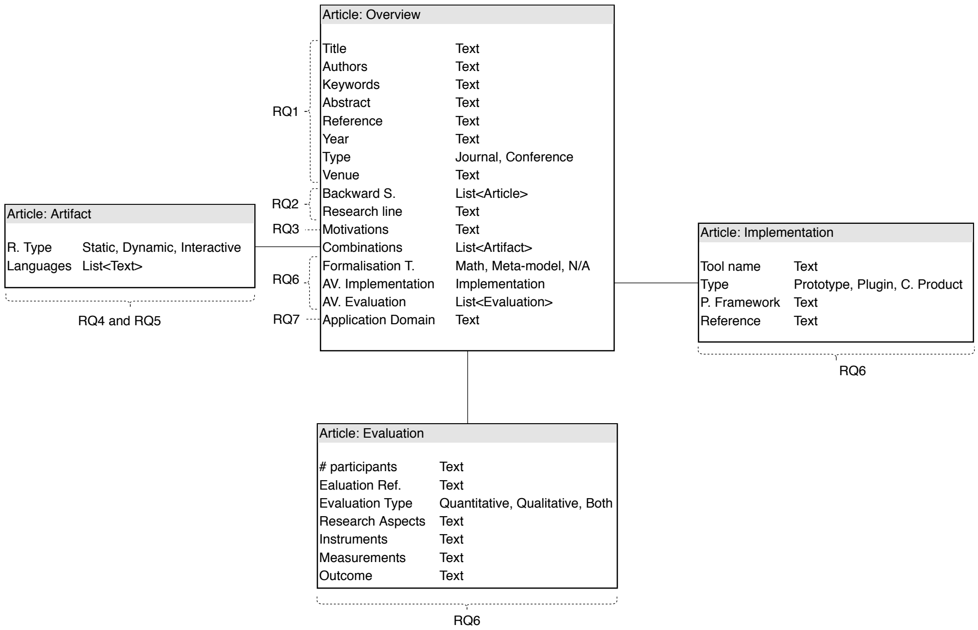


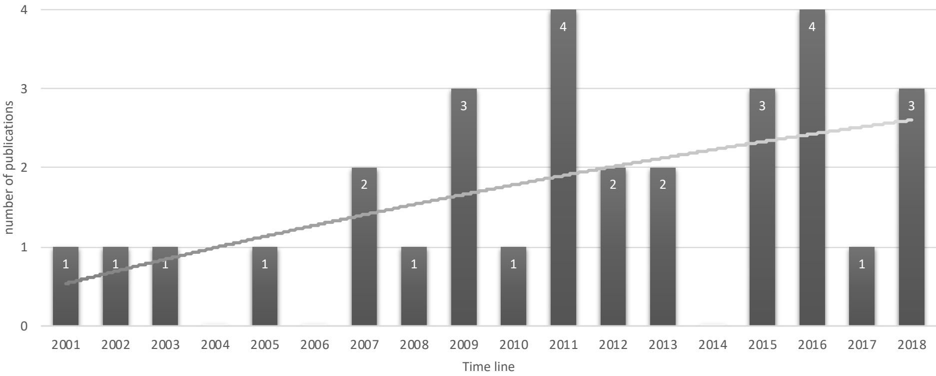
Figure 12: Data extraction scheme.

Titles	Venues	Years	Authors
The Process Highlighter: From Texts to Declarative Processes and Back [21]	BPM	2018	Hugo A. Lopez et al.
Formal Model of Business Processes Integrated with Business Rules [79]	ISF	2018	Kluza, Krzysztof and Nalepa, Grzegorz J.
Discovering hidden dependencies in constraint-based declarative process models for improving understandability [45]	IS	2018	De Smedt, Johannes et al.
Effect of Linked Rules on Business Process Model Understanding [46]	BPM	2017	Wang, Wei et al.
The Semantics of Hybrid Process Models [24]	OTM	2016	Slaats, Tijs et al.
Improving Understandability of Declarative Process Models by Revealing Hidden Dependencies [67]	CAiSE	2016	De Smedt, Johannes et al.
Web-Based Modelling and Collaborative Simulation of Declarative Processes [70]	BPM	2016	Marquard, Morten et al.
Business Process Flexibility and Decision-Aware Modeling—The Knowledge Work Designer [60]	Book	2016	Hinkelmann, Knut
Mixed-paradigm process modeling with intertwined state spaces [52]	BISE	2015	De Smedt, Johannes et al.
Declarative Process Modeling in BPMN [51]	CAiSE	2015	De Giacomo, Giuseppe et al.
Hybrid Process Technologies in the Financial Sector [22]	BPM	2015	Debois, Søren et al.
Mixing Paradigms for More Comprehensible Models [50]	BPM	2013	Westergaard, Michael and Slaats, Tijs
Towards the Combination of BPMN Process Models with SBVR Business Vocabularies and Rules [78]	ICIST	2013	Mickevičiūtė, Eglė and Butleris, Rimantas
Creating Declarative Process Models Using Test Driven Modeling Suite [16]	CAiSE	2012	Zugal, Stefan et al.
Enriching Business Processes with Rules Using the Oryx BPMN Editor [73]	ICAISC	2012	Kluza, Krzysztof et al.
Patterns for Flexible BPMN Workflows [59]	EuroPLoP	2011	Zimmermann, Brigit and Doehring, Markus
Modeling Flexible Business Processes with Business Rule Patterns [65]	EDOC	2011	Milanovic, Milan et al.
Framework for Business Process and Rule Integration: A Case of BPMN and SBVR [68]	BIS	2011	Cheng, Ran et al.
Toward enhanced life-cycle support for declarative processes [69]	SEP	2011	Zugal, Stefan et al.
Exploiting Rules and Processes for Increasing Flexibility in Service Composition [72]	EDOC	2010	Sapkota, Brahmananda and van Sinderen, Marten
Flexibility as a Service [75]	DASFAA	2009	van der Aalst, W. M. P. et al.
On managing business processes variants [58]	DKE	2009	Lu, Ruopeng et al.
Towards a Language for Rule-Enhanced Business Process Modeling [74]	EDOC	2009	Milanovic, Milan and Gasevic, Dragan
Achieving Business Process Flexibility with Business Rules [66]	EDOC	2008	van Eijndhoven, Tim et al.
DECLARE: Full Support for Loosely-Structured Processes [8]	EDOC	2007	Pesic, Maja et al.
Patterns of Business Rules to Enable Agile Business Processes [71]	EDOC	2007	Graml, Tobias et al.
Specification and validation of process constraints for flexible workflows [25]	IS	2005	Sadiq, Shazia W. et al.
A constraint specification approach to building flexible workflows [77]	RPIT	2003	Mangan, Peter and Sadiq, Shazia
On Building Workflow Models for Flexible Processes [76]	ADC	2002	Mangan, Peter and Sadiq, Shazia
Pockets of Flexibility in Workflow Specification [37]	ER	2001	Sadiq, Shazia et al.

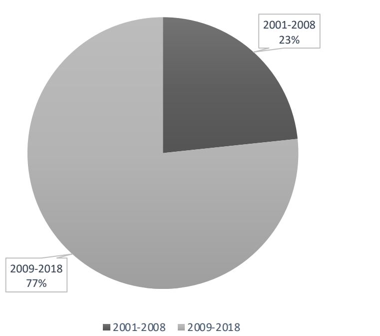
Table 1: Final study retrieval list. The publication venues are abbreviated as mentioned in Section 4.4. The new venues are the following: DASFAA (International Conference on Database Systems for Advanced Applications), DKE (Data and Knowledge Engineering), IS (Information Systems), SEP (Software: Evolution and Process), ICAISC (International Conference on Artificial Intelligence and Soft Computing), ISF (Information Systems Frontiers), Australasian Database Conference (ADC), EuroPLoP (European Conference on Pattern Languages of Programs) and RPIT (Research and Practice in Information Technology).

5.2.1. *Labeling Validation*

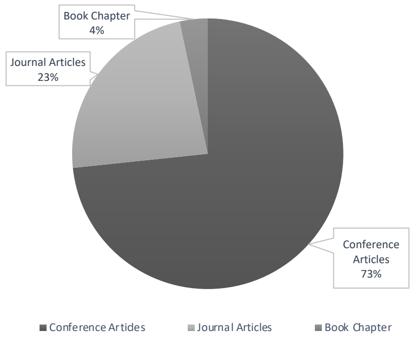
To substantiate our subjective labeling, we have followed the approach introduced in Section 4.8. As shown in Figure 14, two nearly independent clusters of articles can be discerned. Namely, the cluster of articles in RL1 (colored in red) and the cluster of articles in RL2 (colored in green). Except a unique edge (cross-reference) between van Eijndhoven et al. [66] and Sadiq et al. [37], none of the other articles in one cluster has cited articles from the other cluster. By closely inspecting the context where [66] cited [37], we have found a single citation that came up in the context of sharing the same concerns as Sadiq et alẇith regards to process flexibility. However the



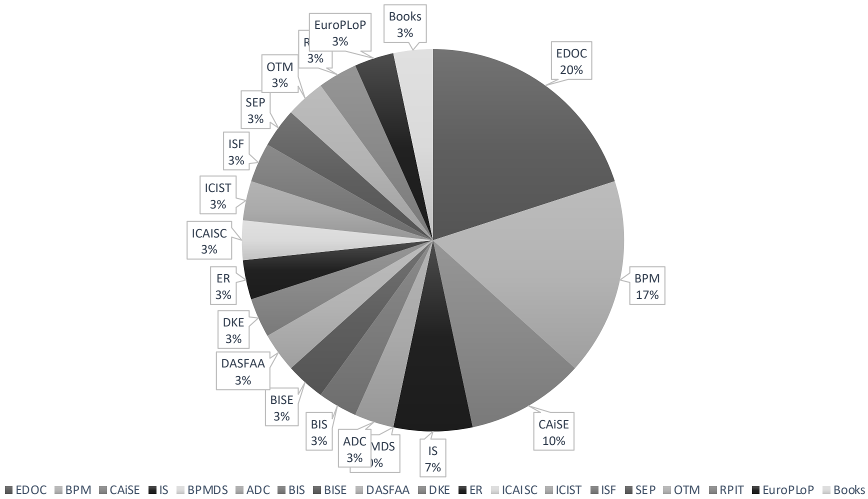
(a) Time distribution of articles organized by their year of publication.



(b) Time distribution of articles in the two last decades.



(c) Distribution of the articles based on their publication venue type.



(d) Distribution of the articles based on their publication venue.

Figure 13: Different distributions of the articles in Table 1.

two approaches use different concepts and techniques to define HBPRs.

Additionally, one article (i.e., Hinkelmann [60]) remained independent from both research lines. Although, the approach proposes a hybrid representation addressing the separation of concerns between process and business logic, no cross-referencing has been identified in relation with the articles of RL2. By scrutinizing the article, we have noticed that the author did not explicitly describe the related work about the similar hybrid representations, which could in turn explain the lack of connections with other similar articles.

The size of the nodes provides insights about the approaches which offer the most widespread traction and were used as a basis to develop other approaches. In the context of the HBPRs covered in this study, [8], [37] and [50] were the most cited

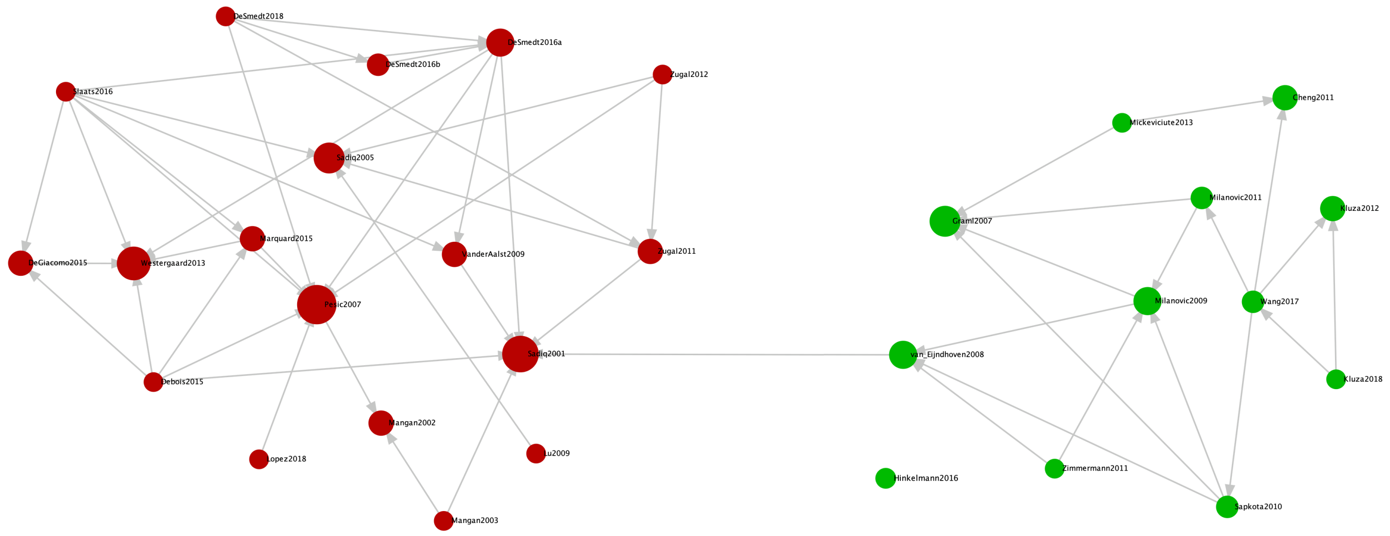


Figure 14: Cross-referencing graph of the articles in Table 1. The colors of the nodes refer to their corresponding research lines. The direction of the arrows indicates the citing between articles.

approaches in RL1, while [71] was the most cited approache in RL2. The impact of these approaches is discussed in Section 6.

In the next sub-sections (cf. Sections 5.2.2 and 5.2.3), the articles proposed within each research line are introduced.

5.2.2. Extending and Supporting Declarative Languages

The use of HBPRs extending declarative languages occupies a large part of the literature. Sadiq et al. [37] propose a hybrid approach to construct process models. Herein, an imperative process model is extended with several "pockets of flexibility" where the relations between the activities in each pocket are modeled in a declarative – constraint-based style. Pockets of flexibility are defined at design-time and explicit workflow executions are constructed at run-time according to the pockets' constraints. Using the same concept, Mangan and Sadiq [76, 77] extend this work and propose a HBPR combining basic imperative modeling constructs with dynamic constraints (cf. Section 5.4). The later is further elaborated in [25]. Similarly, Lu et al. [58] present a HBPR combining pre-defined model parts with loosely coupled model parts using a constraint-based approach.

Pesic et al. [8] introduce *Declare* as a new constraint-based process modeling language for loosely coupled process models. Since Declare is not appropriate for modeling highly structured processes due to its constraint-based nature and the inability of users to make sense of process models with too many constraints [14], the authors suggested to combine it with YAWL (i.e., an imperative language). This in turn, would enable modeling both highly structured parts and loosely structured parts of process models efficiently. An extension for this approach was later proposed by Van der Aalst et al. [75].

Westergaard and Slaats [50] proposed a hybrid language allowing for the use of both the imperative and declarative formalisms within the same process model. This in turn, enables a concise representation for unstructured processes and detailed specifications for structured processes. Similarly, De Giacomo et al. [51] propose a new extension of BPMN entitled BPMN-D. With the same concept in mind, Slaats et al. [24] propose a hybrid framework, allowing to model processes in a hierarchical structure such that each sub-process of the hierarchy can be modeled using either an imperative or declarative language.

De Smedt et al. [52] address the trade-off between understandability and flexibility when using an imperative or a declarative modeling language, and analyze the implications of combining languages from different paradigms (particularly in terms of their syntax, execution semantics, and understandability). On these grounds, the authors introduce a step-wise approach to derive a hybrid language.

Zugal et al. [69, 16] propose a Test Driven Modeling (TDM) approach addressing the understandability and maintainability of declarative process models. The authors base their approach on the concept of computational offloading from CLT [80], which refers to the extent to which a user can extract certain information from a business process model quickly. As declarative process modeling languages have lower computational offloading (compared to imperative languages), enriching them with test cases would increase their computational offloading by providing means to easily locate hidden dependencies and validating specific scenarios against forbidden behaviors. Similarly, Marquad et al. [70] and Debois et al. [22] propose HBPRs to support declarative process models using guided simulations.

Lopez et al. [21] introduce the process highlighter as a means to clarify the semantics of declarative process models. The proposed hybrid representation interlinks the modeling constructs (i.e., roles, activities and constraints) of the model with the corresponding fragments in the textual description. This hybrid approach, in turn, provides a better alignment between the two artifacts and improves the comprehension of the model [48].

In another quest to support declarative process models, De Smedt et al. [67, 45] address the issue of hidden dependencies and its negative impact on the understandability of declare process models (cf. Section 1). In this vein, a hybrid representation revealing hidden dependencies is proposed to avoid all sort of ambiguities while conjoining Declare constraints.

5.2.3. Integrating Business Rules with Business Processes

The second research line evoking HBPRs in the literature is associated with approaches addressing the separation of concerns between imperative processes and business rules. These approaches aim at integrating business rules with process models. In this context, business rules are extracted from the control flow and represented in natural language or following specific rule-based languages (usually using an intuitive syntax). This way, the complexity of the process model is reduced and higher flexibility and adaptability are ensured at run-time. What makes the difference between the approaches proposed in this research line compared to the previous one is that, here, constraints are represented in a more human readable way (i.e., annotations), which in turn, improve the understandability of the business process. Business rules can be appended to a process model as model annotations [68] or as linked rules connected to specific parts of the process model [72]. In both cases, business rules provide a HBPR combining two different artifacts (i.e., business rules with a process model).

The literature proposes a variety of approaches aiming at combining business rules with process models. Cheng et al. [68] address four important aspects to consider while combining both artifacts: *(1)* Business rules and imperative process models have different representations. In other words, business rules tend to be represented textually while imperative process models tend to be represented graphically. Thus, the combination should map the textual annotations to the business process constructs with a minimal information loss. *(2)* The semantics of both artifacts are fundamentally different. *(3)* Each of the artifacts targets different levels of abstraction. While imperative process models elicit the way activities should happen, business rules emphasizes what should happen. *(4)* Nevertheless, an overlap between the specifications provided by both artifacts is possible which should be also taken into consideration when using such combinations. The authors propose an overarching framework to support the combination of imperative models and business rules by introducing two mapping methods allowing to identify the inconsistencies between business rules and imperative business processes. Similarly, Mickeviciute and Butleris [78] investigate the combination capabilities of imperative and declarative languages and address a possible mapping of elements of both languages to infer inconsistencies and overlaps.

Kluza et al. [73] claim that imperative languages are not suitable to model low-level logic of tasks in business processes. Alternatively, the authors propose to use business rules to integrate the low-level logic in business processes. In this work, the authors address two important integration aspects: *(1)* The visual modeling aspect of business rules by introducing an approach to manage them visually. *(2)* The execution aspect of the integrated model by providing an execution environment implementing a rule engine. Kluza and Nalepa [79] extend this work by introducing a formal semantics for integrating business rules in a process model.

Graml et al. [71] argue that existing business processes (modeled in an imperative style) lack adaptability and are unable to cope with changes in real-time. To ensure a high flexibility and better adaptability, the authors propose a set of modeling patterns allowing to extract derivation rules (used for decisions), constraints (used to enforce decisions), and process rules (used to define the dependencies between the process activities) from process models. These rules can be defined separately from the control flow and then integrated as linked rules. Following this approach, the resulting HBPR combines a process model and business rules modeled as linked rules. Similarly van Eijndhoven et al. [66], propose a rule-based approach to separate business rules from process models. The approach consists of of first discerning the static and the changing parts in the process model, then representing the latter as business rules. This way, all modifications on the changing part of the process will involve only editing the business rules.

Milanovic and Gasevic [74] introduce a HBPR allowing to model the different modeling patterns proposed in [71] to integrate business rules in business processes. This work was further elaborated in [65] by Milanovic et al., where the authors review the modeling patterns proposed in [71] and address the lack of a systematic modeling approach that abstracts from the implementation details and rather focus on the modeling itself. Consequently, the authors refined the approach proposed in [71] and abstracted it from any technology dependency.

Similarly, Zimmermann and Doehring [59] introduce a HBPR aiming at extracting the contextual facets from the process model and representing them as business rules. Hence, reducing the complexity of the process model. This way also, dynamic changes are supported and process instances are able to adapt to events and changes of context variables at run-time.

Sapkota and van Sindern [72] address the continuous change in business demands and the inability of existing service composition techniques to cope with flexibility and adaptability of business processes. Consequently, the authors consolidate between declarative and imperative designs by deploying business rules to define constraints and handle service orchestration in a dynamic manner. Similarly to Graml et al. [71], the authors emphasize the importance of extracting rules from process models, then integrating them in a way that business processes can adapt to changing requirements and ensure rules consistency without altering the composition logic.

The understandability of HBPRs combining a process model with business rules was investigated by Wang et al. [46]. In the design of their experiment the authors deploy a layout where a process model and linked rules are displayed side by side. This layout illustrates a hybrid representation that combines a process model with business rules expressed in natural language.

Hinkelmann [60] proposes a hybrid representation allowing to represent process logic imperatively and business logic declaratively in an integrated manner. As mentioned in Section 5.2.1, although no direct connections can be established (in term of cross-referencing) with the article of RL2, the proposed approach is clearly addressing the separation of concerns between process logic and business logic, and thus sharing similar characteristics with the other approaches in RL2.

5.3. Motivations

This section presents the motivations driving the emergence of the proposed HBPRs (answering RQ3). In the context of this study, the motivations are derived based on the functional aspects of the proposed approaches. This information is directly extracted from each of the selected articles after being fully read.

The articles published within the different research lines share several motivations. Most of the authors in the literature motivate their approaches and explicitly describe the motivations behind the proposed HBPRs. Through the literature, the following motivations were identified:

(a) Enhancing process flexibility and allowing adaptability at run-time by combining loosely structured model parts with highly structured parts.

(b) Reducing the complexity of process models by separating business rules from the control flow, then integrating them in a hybrid representation.

(c) Introducing hybrid languages to deliver the most adequate language allowing to represent business processes more concisely and precisely.

(d) Improving the understandability of process models and fostering the communication between the different process stakeholders (i.e., domain experts and IT specialists).

(e) Improving the maintainability of process models and ensuring better process re-usability.

(f) Supporting the modelers in the PPM.

Table 2 summarizes the motivations driving the emergence of the approaches proposed within RL1 and RL2. The table shows that several articles intersect with more than one motivation despite their context or research line. By looking at the distribution of the motivations over the two research lines, one can notice that both research lines share motivations about process flexibility, understandability and maintainability. However, only approaches in RL1 aim at improving the PPM and enhancing the conciseness and preciseness of hybrid languages, whereas only approaches in RL2 aim at separating and integrating business rules.

5.4. Combined Languages and Artifacts

This section identifies the languages and artifacts combined in the proposed HBPRs (answering RQ4). As shown in Table 3, some languages were commonly deployed in several hybrid approaches to model either imperative or declarative process specifications. With respect to RL1, Declare was the most common language to be combined with other languages (i.e., [75, 8, 50, 24, 52, 51, 67, 45, 69, 16]). Besides Declare, DCR was combined with textual annotations (i.e., [21]) and language independent representations (i.e., flow-based representation [22] and guided simulation tool [70]) in order to model declarative specifications. The proposed HBPRs were represented using different types of artifacts. On that matter, two distinct types of artifacts can be discerned, namely, static artifacts and interactive artifacts. Static artifacts were represented as a process model (i.e., [75, 8, 50, 24, 52, 51, 67, 45, 69, 16, 70, 22]) and a textual description (i.e., [67, 45, 21]). Whereas interactive artifacts were represented as a guided simulation (i.e., [69, 16, 70, 22]).

Regarding RL2, BPMN was the only language used to represent the structured parts of business processes, while several other declarative languages were used to represent the unstructured parts (i.e., [68], [78], [73], [79], [65], [59], [60], [46]). The proposed HBPRs in this case were represented using the combination of two static artifacts (i.e., a process model and a textual description).

In addition, several approaches in both RL1 and RL2 abstracted from particular language specifications and rather proposed generic approaches that can be adapted to a wide range of declarative and imperative languages. In RL1, some articles (i.e., [37, 25, 58, 76, 77]) propose hybrid approaches combining generic imperative constructs with declarative constraints. The proposed HBPRs were all represented as static artifacts i.e., process models. Alternatively, in RL2, some articles (i.e., [71, 66, 72]) restricted the modeling of imperative specifications to BPMN , while they still abstracted from choosing a particular language to model declarative specifications. Nevertheless, all the resulting HBPRs assume the combination of two static artifacts i.e., a process model to represent imperative specifications and textual descriptions to represent declarative specifications. Therefore, the use of these approaches is limited to only declarative languages which are conventionally represented in a textual format.

By looking at the different combinations of artifacts shown in Table 3, one can notice that none of the proposed HBPRs comprises a dynamic artifact. On that matter, one can argue that none of the covered approaches has deployed input from event logs to support HBPRs. This limitation is further discussed in Section 6.

5.5. Taxonomy

This section presents a descriptive taxonomy (answering RQ5) based on the outcome of the combinations of the artifacts and the languages presented in Section 5.4. Process artifacts can be categorized according to their inherent and visual features (cf. Section 2.3). The inherent features can be used to discern the formality and the language paradigm characteristics of the languages combined in a HBPR. In terms of formality, some HBPRs in the literature *(1)* combine two languages having a formal syntax and formal semantics, whereas other HBPRs *(2)* combine a language having a formal syntax and semantics with a language having a semi-formal syntax or semantics. Another set of HBPRs *(3)* uses a language having a formal syntax and semantics together with a language having

Motivations	Articles in RL1	Articles in RL2
(a) Process flexibility	[37], [25], [76], [77],[58], [50], [75], [51], [24], [52],	[71], [66], [74], [65], [59], [72]
(b) Separating and integrating business rules		[71], [66], [74], [65], [59], [72], [60]
(c) Conciseness and preciseness of hybrid languages	[50], [51], [24], [52]	
(d) Understandability	[52], [69], [16], [67], [45], [70], [22]	[68], [78], [73], [79], [74], [65], [59], [46], [21], [60]
(e) Maintainability	[69], [16], [37], [25], [70], [22]	[66], [59], [72], [21], [60]
(f) Improving the PPM	[69], [16], [67], [70], [22], [21]	

Table 2: Summary of the motivations behind the HBPRs proposed in the literature.

		Languages														Artifacts		
		Imperative						Declarative								Static		Interactive
	Work	BPMN	YAWL	Petri net	C.Petri net	I-NL	Lang. Indep/Unspecified	Declare	DCR	CMMN	C-NL	R2ML	SBVR	XTT2	Lang. Indep./Unspecified	Text.Des.	P.Model	Guided Sim.
RL1	[75]		x					x									x	
	[8]		x					x									x	
	[50]				x			x	x								x	
	[24]			x				x									x	
	[52]			x				x									x	
	[51]	x						x									x	
	[67]					x		x								x	x	
	[45]					x		x								x	x	
	[69]						x	x									x	x
	[16]						x	x									x	x
	[70]						x		x								x	x
	[22]						x		x								x	x
	[21]					x			x		x					x	x	
	[37]						x								x		x	
	[25]						x								x		x	
	[58]						x								x		x	
	[76]						x								x		x	
	[77]						x								x		x	
RL2	[68]	x											x			x	x	
	[78]	x											x			x	x	
	[73]	x												x		x	x	
	[79]	x												x		x	x	
	[74]	x										x				x	x	
	[65]	x										x				x	x	
	[59]	x										x				x	x	
	[60]	x								x							xx	
	[46]	x									x					x	x	
	[71]	x													x	x	x	
	[66]	x													x	x	x	
	[72]	x													x	x	x	

Table 3: List of the languages and artifacts combined in the HBPRs proposed within RL1 and RL2.

		Formal and Formal		Formal and Semi-formal	Formal and Informal		Generic Language	
		RL1	RL2	RL2	RL1	RL2	RL1	RL2
Hybrid Language	Hierarchical Structure	[8], [75], [24]					[37], [76], [77], [25], [58]	
	Mixed Structure	[50], [51], [52], [60]						
Hybrid Artifacts	P. Model and T. Descriptions		[73], [79]	[68], [78],[74], [65], [59]	[67], [45], [21]	[46]		[71], [66], [72]
	P. Model and Guided Sim.						[69], [16], [70], [22]	

Table 4: Categorization of the literature articles based on their language formality and visual features. *"Formal and Formal"* refers to the combination of two languages having a formal syntax and formal semantics. *"Formal and Semi-formal"* refers to the combination of a language having a formal syntax and semantics with a language having a semi-formal syntax or semantics. *"Formal and Informal"* refers to the combination of a language having a formal syntax and semantics with a language having an informal syntax and semantics.

an informal syntax and semantics. Table 4, categorizes the literature articles according to their language formality[6]. In terms of the language paradigm, HBPRs can be grouped as previously shown in Table 3.

The visual feature can be used to discern the types of the artifacts combined in the proposed HBPRs. As mentioned in Section 2, HBPRs can be divided into *(a)* hybrid languages and *(b)* hybrid process artifacts. In the literature, hybrid languages are composed in a single static artifact, whereas hybrid process artifacts are composed using multiple static and interactive artifacts. By looking closely at the articles describing hybrid languages, two different types of structures emerge: *hierarchical structures* and *mixed structures*. With hierarchical structures, process models are fragmented into sub-processes or so so called "Pockets of Flexibility" [37] where each sub-process or pocket can be modeled using a declarative or an imperative language. Such a decomposition reduces the complexity of the hybrid representation and allows for s better re-usability of the existing model fragments [24]. Alternatively, *mixed structures* allow combining declarative and imperative languages within the same process or sub-process. This way, languages from different ends of the imperative–declarative paradigm spectrum can be fully mixed to represent both imperative and declarative specifications in a compact manner. Although this approach is not very common in the literature some articles (i.e., [50, 51, 52, 60]) have used mixed structures in the design of their hybrid languages. The structures of hybrid

[6] Approaches abstracting from combining particular language specifications are not covered by the formality grouping.

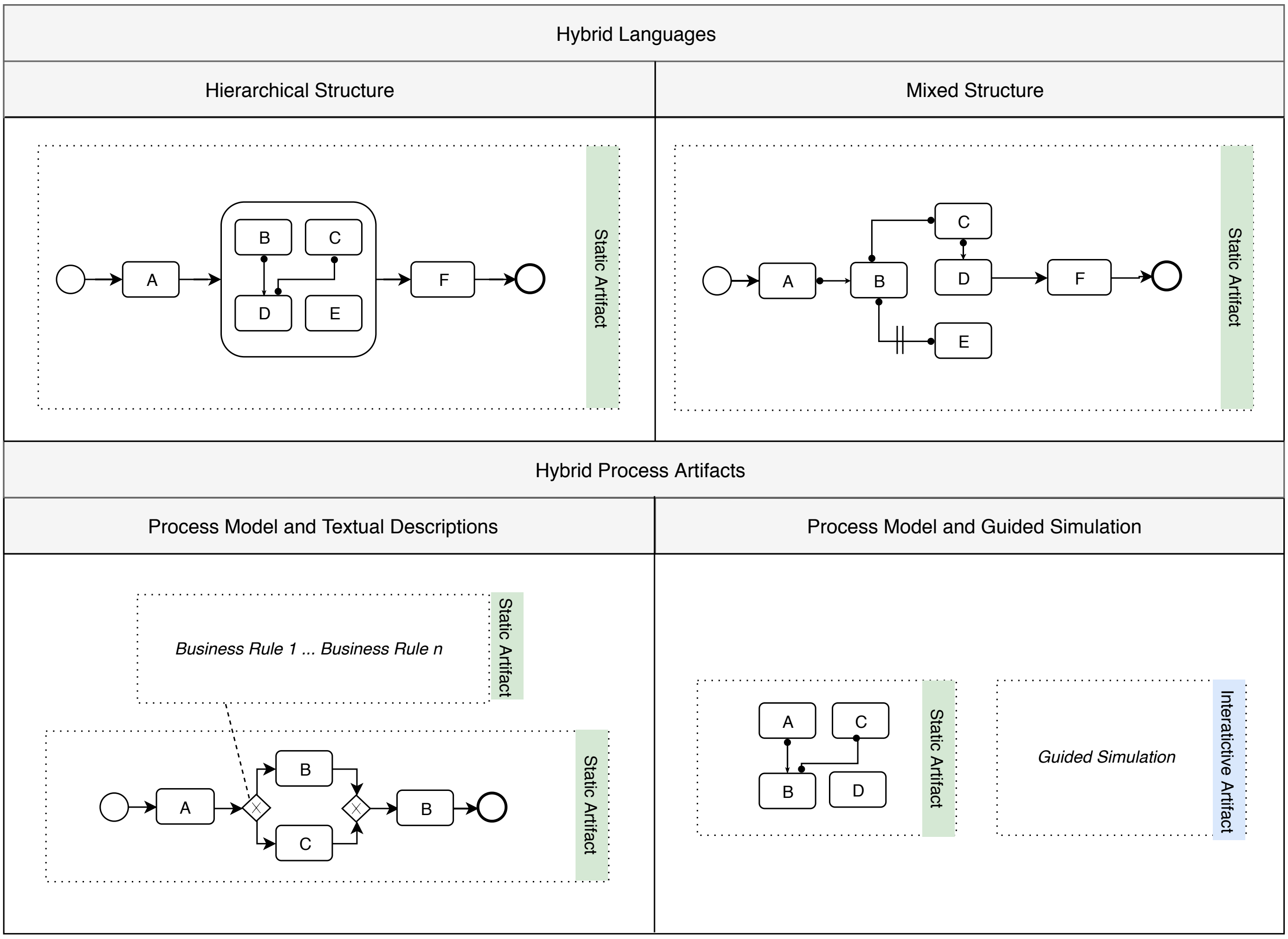


Figure 15: Summary of the descriptive taxonomy. Black edges with arrow head (adapted from BPMN) are used to illustrate the control flow expressed by imperative languages, whereas black edges with bullet head (adapted from Declare) are used to illustrate declarative constraints. Process Models and guided simulations have a formal syntax and formal semantics, whereas business rules can have a formal, semi-formal or informal syntax or semantics.

languages identified in the literature complement the characteristics of hybrid languages introduced in Section 3.1.

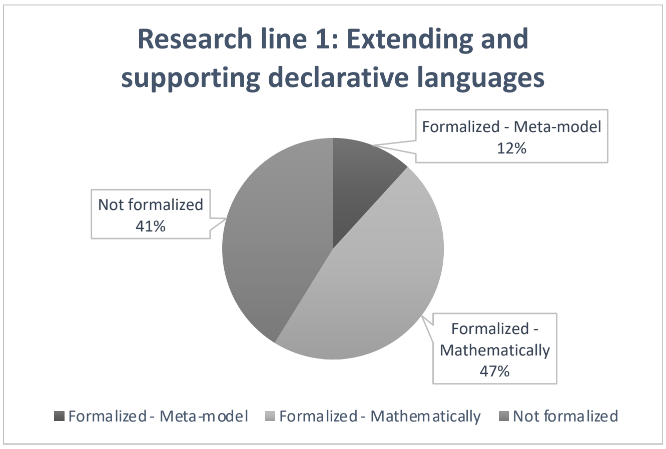
Hybrid process artifacts have been represented differently in the literature. Some articles combine two static process artifacts, which are represented as a combination of a process model with textual descriptions. Other approaches combine a static artifact with an interactive one, namely, a process model with a guide simulation. Table 4 categorizes the literature articles based on their visual features, and Figure 15 summarizes the proposed taxonomy graphically.

5.6. Maturity

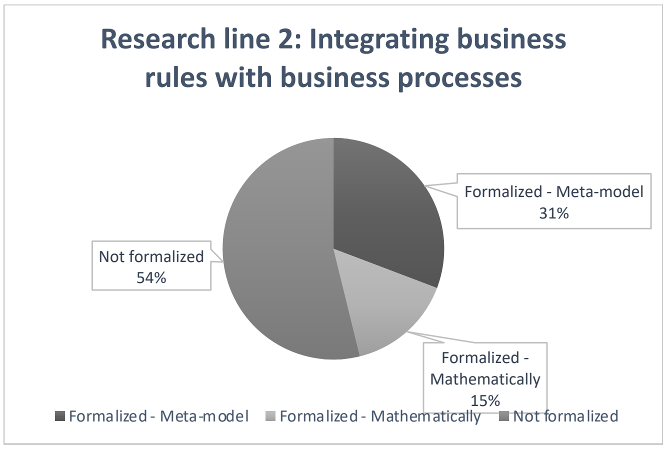
This section evaluates the maturity of the proposed approaches (answering RQ6). To this end, a set of maturity criteria has been defined according to the recommendations of modeling experts. First, the degree of formalization evaluates the extent to which the proposed approaches can be used to design a HBPR consistently. Secondly, the availability of implementation provides indications about how swiftly they can be applied in realistic settings. Finally, empirical evaluations provide an assessments of the proposed approaches under different conditions. Section 5.6.1 compares the articles based on their degree of formalization. Section 5.6.2 provides details about the implementations proposed in the literature and compares them. Section 5.6.3 describes the results of the empirical evaluations conducted in the literature and provides a comparison for them.

5.6.1. Formalization

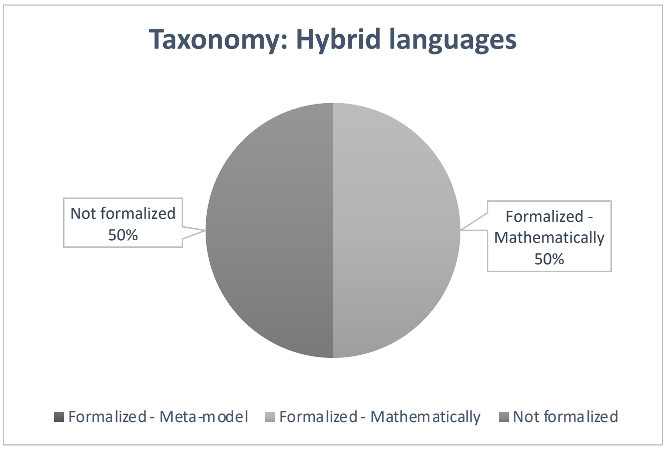
The approaches introduced in the literature are described with varying levels of formalization. Indeed, part of the proposed approaches use meta-models to describe the components of their HBPRs. Although meta-models might not be enough to formally describe the proposed approaches, their abstraction allows conveying the overall idea and provides a moderate understanding to the reader. However, as a mathematical formalization is missing for these approaches, side issues and misinterpretations might be encountered during their deployment. Besides, several authors in the literature have provided a mathe-



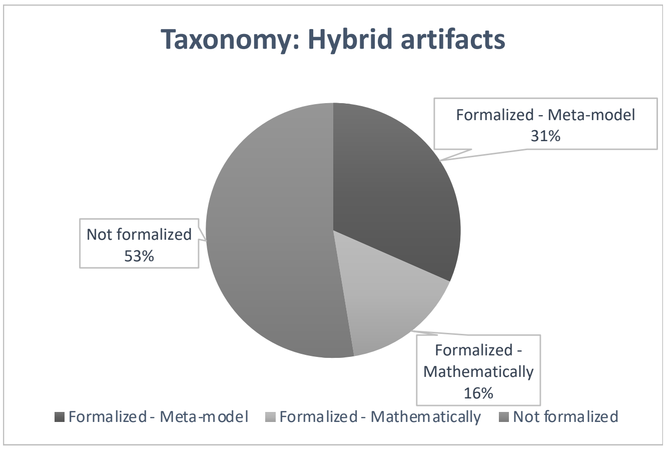
(a) Distribution of articles of RL1 (cf. Section 5.2.2) according to their formalization.



(b) Distribution of articles of RL2 (cf. Section 5.2.3) according to their formalization.



(c) Distribution of articles proposing a hybrid language (cf. Section 5.5) according to their formalization.



(d) Distribution of articles proposing a hybrid process artifact (cf. Section 5.5) according to their formalization.

Figure 16: Distribution of the articles according to their formalization based on different categorizations.

Deg. of Formalization	**Publications Refs.**
Meta-model	[74], [65], [59], [69], [16], [60]
Mathematical	[77], [75], [51], [24], [52], [71], [67], [45], [25], [58]
No Formalization	[37], [76], [8], [50], [68], [78], [73], [79], [66], [72], [46], [22], [70], [21]

Table 5: Degrees of formalization of each of the HBPRs proposed in the literature.

matical formalization to describe their approaches. With this regards, the proposed formalizations provide an overarching understanding of the approaches and mark their readiness for deployment in realistic settings. Another portion of the literature did not provide any formalization to their approaches. By inspecting the articles with none formalized approaches, one can notice that most of them were either part of new initiatives, intermediate work, evaluations or approaches proposed in ad-hoc contexts (i.e., for specific case studies). Table 5 summarizes the degrees of formalization of each of the approaches proposed in the literature.

Figure 16 compares the formalization of the approaches based on the research lines categorization and the taxonomy introduced in Sections 5.2 and 5.5 respectively. By comparing the two research lines, one can notice that nearly half of the approaches proposed within RL1 and RL2 are formalized. Looking at the different types of formalization, one can see a tendency to formalize approaches in RL1 mathematically, whereas more approaches in RL2 are formalized using a meta-model. By comparing the approaches based on the proposed taxonomy (i.e., hybrid languages, hybrid artifacts), one can notice a balanced distribution between none-formalized and formalized approaches in both research lines. However, none of the approaches in RL1 has been formalized using a meta-model, whereas, nearly two-thirds of approaches in RL2 have been formalized using a meta-model. These insights suggest that the approaches in RL1 and the approaches proposing a hybrid language are the candidates to represent HBPRs consistently, which in turn raise the need to provide mathematical formalizations for the approaches in RL2 and for the approaches proposing a hybrid artifact.

5.6.2. Availability of Implementation

The availability of implementation is another important aspect allowing to assess the maturity of the proposed HBPRs. Information about the implemented approaches including the tool name, type, parent framework and a reference to the tool are shown in Table 6. This information provides clear insight into the maturity of the proposed approaches in terms of their implementation characteristics. For instance, the implementation type allows discerning whether the tool is a prototype, a plugin or a commercial product. Furthermore, the implementation type can also provide indications about the tool integration in industry, as commercial products are more likely to be used in industry compared to prototypes or plugins.

Among the 30 articles found in the literature, 18 comprise an implementation. As shown in Table 6, most of the implemented approaches were either prototypes or plugins, whereas only 3 approaches are available as part of a commercial product. Furthermore, as prototypes were mostly used as a proof of concept, some articles have not shared their implementation source code in their publications while other implementations have been discontinued.

By comparing the approaches proposed within the two research lines identified in Section 5.2 (cf. Figures 17a and 17b), it is visible that more approaches have been implemented in RL1 compared to RL2. In addition, by considering the development of commercial products, one can claim that RL1 comprises mature implementations which can be adopted in industrial settings. When comparing the proposed approaches based on the taxonomy in Section 5.5 (cf. Figures 17c and 17d), one can notice that the largest portion of hybrid languages has not yet been implemented and the rest were implemented either as prototypes or plugins. Regarding hybrid artifacts, a large portion has already been implemented as commercial products, prototypes and plugins. This insight denotes the maturity of approaches implementing hybrid artifacts and raises the need for providing a tool-support for the existing hybrid languages since some of them have been already formalized (cf.Section 5.6.1).

5.6.3. Empirical Evaluation

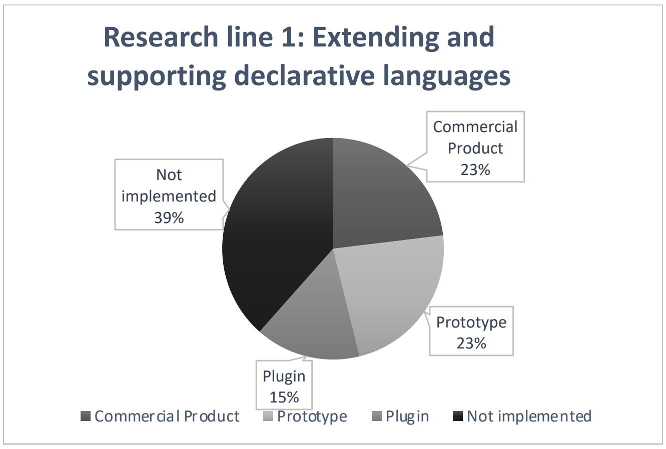
The availability of empirical evaluations is considered as an important factor to evaluate the maturity of the approaches presented in the literature. Among the 30 articles in the literature, only 5 HBPRs developed in 8 articles i.e., [46, 69, 16, 67, 45, 70, 22, 21] were evaluated in the literature. As shown in Table 4, these articles correspond to approaches proposing a hybrid artifact, in particular, approaches combining a process model with textual descriptions and approaches combining a process model with a guided simulation. Table 7 provides an overview about the approaches evaluated in the literature.

The TDM approach [69, 16] was evaluated in [23] and [83], the evaluation has covered two empirical studies. The first study covered 8 participants and investigated the extent to which the proposed HBPR helps to foster the communication with domain experts and IT specialists during the PPM. In this experiment, the communication between the process stakeholders during the PPM was recorded. Then following, the CoPrA approach [84], the verbal data was transcribed and coded according to a specific coding scheme. Additionally, video recordings of the modeling sessions were also collected to perceive the context of the verbal data. A qualitative analysis of the coded data shows that test cases were accepted by the participants as communication channel and the HBPR contributes to a better understandability of declarative process models. The second study has covered 12 participants and investigated the maintainability of the proposed representation. In this study, the TDM framework and the Cheetah experimental platform [85] were used to track the PPM and to assess the quality of the obtained process models. In addition, questionnaires were deployed to measure the cognitive load and the quality of the process models as perceived by the participants by the end of the modeling session. The results of a quantitative analysis demonstrate that the proposed representation improves the maintainability of declarative process models, lowers the cognitive load, and increases the perceived model quality.

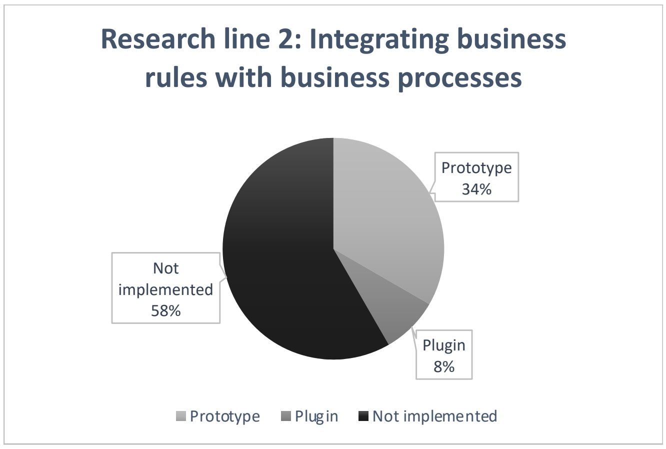
The approach extending a declarative process model with textual descriptions of the hidden dependencies between the Declare relations [67, 45] was evaluated in two studies. A first study was reported in [67] with 95 participants then extended in a second study to cover 146 participants in [45]. In this work, the authors investigate the impact of the proposed HBPR on model understandability by scrutinizing the effect of adding an extra layer of textual descriptions to declarative process models. To this end, the authors used Declare Execution Environment (cf. Table 6) to record important interactions (i.e., opening a dependency graph visualization) and response time. Furthermore, the authors use questionnaires to evaluate the participants comprehension and self-assessment of cognitive load. The results of quantitative and qualitative analyses demonstrate that the use of textual descriptions contributes to an enhanced understandability, reduced mental effort and response time when dealing with a HBPR compared to a declarative process model representation.

The understandability of HBPRs where linked rules are combined with imperative process models was evaluated in [46]. This representation illustrates the common representation of HBPRs proposed in RL2 (cf. Section 5.2.3). The study has covered 58 participants. In order to investigate the impact of integrating business rules in process models, an Eclipse RCP application[7] illustrating a HBPR was developed. Furthermore an eye tracking device was deployed to record the participants gaze data during the experiment, which in turn, were used to derive the total fixation duration (i.e., sum of the duration of all fixations on a specific area of the stimulus [86]). As a performance measure, the response times of participants

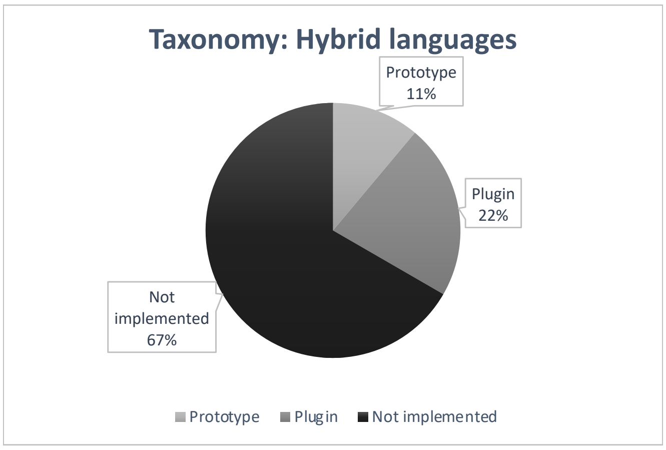
[7]See `https://wiki.eclipse.org/Rich_Client_Platform`



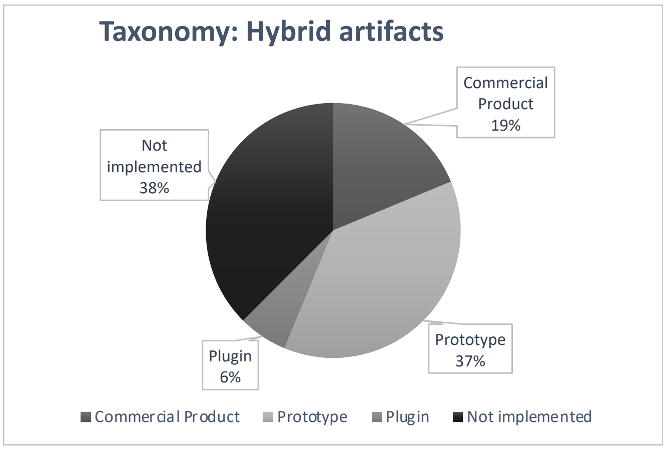
(a) Distribution of articles of RL1 (cf. Section 5.2.2) according to their implementation type.



(b) Distribution of articles of RL2 (cf. Section 5.2.3) according to their implementation type.



(c) Distribution of articles proposing a hybrid language (cf. Section 5.5) according to their implementation type.



(d) Distribution of articles proposing a hybrid process artifact (cf. Section 5.5) according to their implementation type.

Figure 17: Distribution of the articles according to their implementation type based on different categorizations.

Publications	Tools Name	Implementation Type	Parent Frameworks	References
[22]	DCR swimlanes	Commercial Product	DCR Solutions	`http://dcrgraphs.net`
[67, 45]	Declare Execution Environment	Prototype		`http://processmining.be/declareexecutionenvironment`
[59]	vBPMN	Prototype	jBoss Drools 5.1	N/A
[66]	Unamed	Prototype	Aqualogic BPM Studio and ILOG business rules engines	N/A
[71]	Unamed	Prototype	IBM WebSphere Integration Developer/ Process Server	N/A
[73, 79]	Oryx-HQEd	Plugin	Oryx Editor	`https://ai.ia.agh.edu.pl/wiki/hekate:start`
[70]	DCR Simulation Tool	Commercial Product	DCR Solutions	`http://dcrgraphs.net`
[21]	Process Highlighter	Commercial Product	DCR Solutions	`http://dcrgraphs.net`
[60]	The knowledge work designer	Prototype	ADOxx	`http://adoxx.org`
[37, 25]	Chameleon	Prototype	FlowMake	Discontinued
[75, 8]	Declare service	Plugin	YAWL environment	cf. [81]
[50]	Unamed	Plugin	CPN Tools	`http://cpntools.org` cf. [82]
[69, 16]	TDMS	Prototype	Cheetah, Declare Framework	`http://www.zugal.info/tdms`

Table 6: Information about the implementations proposed in the literature.

Articles	Ev. Ref.	# Par.	Ev. type	Instruments	Measurements	Research Aspects
[46]	[46]	58	Quan.	RCP app, eye tracking, questionnaires	Comprehension accuray, response time, total fixation duration	Effect of integrating BR with BP
[69, 16]	[83]	8	Qual.	Audio and video recordings, questionnaires	Coding of transcribed verbal data	Communication during the PPM
		12	Quan.	TDM framework, Cheethah experimental platform	Cognitive load, perceived quality, quality of model	Maintainability and model quality
[67, 45]	[45]	146	Both	Declare Execution Environment, questionnaires	User interactions, response time, self-rating of cognitive load	Impact on comprehension difficulty
[70, 22]	[44, 43]	15	Quan.	Eye tracking, user Interactions	Fixation-based measures, coding of transcribed verbal data	Attention, reading patterns, use of a HBPR artifacts
[21]	[48]	17	Quan.	User Interactions	User Interactions, coding of transcribed verbal data	Benefits of using a HBPR during the PPM

Table 7: Summary of the evaluation findings. The outcomes of the studies are described in Section 5.6.3. New acronyms: Par. (Participants), Ev. (Evaluation), Quan. (Quantitative), Qual. (Qualitative), BR (Business Rules) and BP (Business Processes).

were used. In addition, cognitive load was measured objectively using eye fixation data [87] and subjectively using the participants' self-assessments of the perceived mental effort. The results of a quantitative analysis show that participants had a higher comprehension accuracy and lower mental effort dealing with a HBPR compared to an ordinary representation where rules are separated from the process model. These insights demonstrate that the combination of an imperative process model with linked rules is associated with an enhanced understandability.

The combination of DCR graphs, a flow-based representation of the control-flow and a simulation tool [22, 70] was evaluated by A. Andaloussi et al. [44]. The authors deploy a HBPR combining a DCR process model with a guided simulation and a swimlane illustrating the flow-based representation. The initial study has covered 10 participants (university students and municipal employees). In order to investigate the understandability of the proposed HBPR, the authors examined the distribution of attention on the different artifacts using a set of eye tracking fixation-based measures [88] including fixation count (i.e., number of fixations on specific area of the stimulus [86]) and total fixation duration. Furthermore, the authors identified the common reading patterns of participants following a process mining based approach proposed in [89]. With this regard, fixation data were converted to event logs, then a process discovery technique [49] was used to infer the implied attention maps. The results of a qualitative analysis demonstrate an unbalanced distribution of attention over the different artifacts and denote the presence of different user profiles exhibiting different reading patterns of the proposed representation. A follow up study was conducted in [43]. In this work, the authors triangulated the subjective insights obtained from the retrospective think-aloud sessions with the objective data recorded by the eye tracking device. The follow up study, which has covered 15 participants with different backgrounds, highlighted the benefits and the challenges associated with using each of the HBPR artifacts individually, and investigated the way users with different backgrounds engage with each of the deployed artifacts. In addition, the study explored the different reading patterns associated with different types of tasks. The results show that the deployment of a single artifact is not enough to provide an overarching understanding for domain experts and IT specialists when dealing with different tasks, which in turn motivate the use of HBPRs.

The process highlighter proposed by Lopez et al. [21] was evaluated by A. Andaloussi et al. [48]. The study has covered 17 participants including employees at a Danish municipality and university students. In this work, the authors investigated the potential support offered by a HBPR during the PPM. The triangulation of the subjective insights obtained from the think-aloud data and the objective insights obtained from the user interactions highlights the support provided by the proposed HBPR during the PPM and hints toward an enhanced quality of process models w.r.t to alignment, traceability and documentation of the process specifications.

5.7. Application Domains

This section reports the different application domains of HBPRs referred in the literature (answering RQ7). The approaches proposed in the literature target dynamic business environments characterized by continuous changes in customer's attitudes and regulations. The authors in the literature illustrate the applicability of their approaches in several domains either by investigating specific case studies or by providing realistic examples illustrating the challenges faced in industrial settings.

The literature shares the same requirements for flexible, understandable and maintainable systems able to adjust for changing customers needs in different domains. As shown in Table 8, the literature covers a variety of application domains such as Health Care, Education, Customer Relation Management (CRM), Web Content Management (WCM), Human Resources Management (HRM), Consultancy, Logistics, Insurance, Banking, and Citizens Services. Moreover, the application of some approaches in the literature is not only limited to specific domains but could also be adapted in designing any HBPR sharing similar needs and motivations as the ones discussed in the literature.

Application Domains	**References**
Auctioning Service	[78]
Book Store Management	[65]
Consultancy Service	[54]
Funding Applications	[24]
Governement Citizens Service	[66], [22], [21]
Health Care	[50], [74], [72] [51], [37], [25], [58], [76], [77]
Education	[37], [25], [58], [76], [77], [60]
CRM	[51], [37], [25], [58], [76], [77]
WCM	[76], [77]
Human Ressource Mangement	[75]
Liability Insurrance Processes	[73], [79]
Loan Application	[2]
Order Delivery/Cash Processes	[52], [71]
Ship Engine Maintainance	[59]
Generic HBPR	[8], [68], [46], [69], [16], [67], [45], [70]

Table 8: Application Domains of the approaches proposed in the literature.

6. Discussion

The analysis presented in Section 5 enabled answering the different research questions of this work. This section discusses the results of the analysis with a twofold purpose. On the one hand, it highlights the important findings in the literature (cf. Section 6.1). On the other hand, it provides a research agenda based on the key findings to guide the emergence of HBPRs (cf. Section 6.2).

6.1. Key Findings

The research lines identified in Section 5.2 discern the contexts where hybrid representations have been proposed. Although both research lines combine languages and artifacts and share similar motivations (e.g., flexibility, understandability and maintainability) the underlying approaches have evolved within their own cluster. In the context of the HBPRs covered in this study, the work of Pesic et al. [8] seems to have the most widespread traction. However this traction is mostly due to the specification of the Declare language, which was cited by all the approaches proposing a hybrid representation including Declare. Besides that, the approaches proposed by Sadiq et al. [37] and Westergaard and Slaats [50] are the ones with the most widespread traction within RL1. Indeed, the two structures characterizing hybrid languages (cf. Section 5.5) have been initiated in these two publications. Namely, Sadiq et al. [37] proposed the hierarchical structure to combine hybrid languages, whereas Westergaard and Slaats [50] proposed the mixed structure. Most of the upcoming publications about hybrid languages have taken inspiration from either of the two approaches. In RL2, the cross-referencing between the different articles is lower compared to RL1. Nevertheless, the modeling patterns used to extract business rules from business processes proposed by Graml et al. [71] were the primary source of inspiration for other similar approaches.

Looking at the combined languages and artifacts in Section 5.4, it is clear that most of the existing hybrid languages combine Declare language with imperative languages, whereas, most of the hybrid process artifacts combine declarative artifacts with process models in BPMN. The descriptive taxonomy introduced in Section 5.5 extends the conceptual framework instantiated in Section 3 by discerning the characteristics of the existing HBPRs. The taxonomy shows that none of the proposed HBPRs comprises a dynamic artifact. Indeed none of the covered approaches has deployed traces from event logs to provide a dynamic visualization as part of a HBPR. This is due to the descriptive nature of the proposed taxonomy which emphasizes only the characteristics of the HBPRs covered by the SLR search. In addition, as mentioned in Section 4.5, this study emphasizes only the modeling of HBPRs, thus the approaches mining HBPRs were excluded, which might explain the lack of approaches incorporating a dynamic artifact. Nevertheless, as explained in Section 2.3, the distinction between dynamic and interactive artifacts is rather clear, which, in turn, motivates its placement into the proposed conceptual framework.

The maturity aspects scrutinized in Section 5.6 provide indications about the ability of the proposed HBPRs to be integrated in industrial settings. In terms of formalization, more approaches proposing hybrid languages are formalized compared to those proposing hybrid artifacts. Regarding the availability of implementation, an inverse pattern can be observed: implementations are more common for hybrid artifacts than hybrid languages. We conjecture that this may be the case because research on hybrid languages tends to be of a more theoretical nature and therefore a formal treatment of the language is expected, whereas research on hybrid artifacts tends to focus more on questions of understandability, which are best demonstrated empirically through the implementation of tools. In terms of the maturity of implementation, except for a few commercial tools, most of the proposed approaches within both research lines present prototypes or plugins mainly as a proof of concept. However, in order to reach the industrial market and to ensure a positive impact on the development of HBPRs, more robust implementations are required. In terms of empirical evaluation, few approaches have been evaluated so far and these cover only hybrid process artifacts.

6.2. Research Agenda

This section discusses a research agenda to delineate the directions for the up-coming research. In order to promote the use of HBPRs, it is necessary to consider the entire HBPR development-cycle, which includes the following phases: *(a)* design, *(b)* modeling and *(c)* evaluation.

<u>Design</u>: In the past two decades, a set of approaches and methodologies has been proposed to design HBPRs. Still, little is known about the synergies and overlaps between the languages composing these hybrid representations. Besides a handful of articles looking into the representation capabilities of hybrid process artifacts integrating business rules with business processes (e.g., [90]), most of the other combinations remain unexplored. On that matter, it is necessary to conduct more ontological analyses questioning the overlap between the existing modeling languages and investigating how different languages can semantically complement each other to derive concise representations of business processes.

It is also crucial to consider the human factor during the design of HBPRs as Lindland et al. [91] said “not even the most brilliant solution to a problem would be of any use if no one could understand it”. In the field of process modeling, cognitive psychology has been deployed to compare the visual support offered by different process modeling languages [92]. For instance, Figl et al. [92] compared the control-flow constructs of several languages (e.g., YAWL, BPMN, EPC) according to a subset of the visual design principles introduced by Moody [93] as part of the physics of notations framework. Namely, the covered languages have been compared based on their representational clarity (i.e., the fit between the graphical symbol representing a construct and the semantic concepts referring to it), perceptual discriminability (i.e., the ease to distinguish between the graphical symbols of a language), perceptual immediacy (i.e., the extent to which a graphical symbol can provide a cue to its meaning), visual expressiveness (i.e., the use of visual variables such as shape, size and color in a language) and graphic parsimony (i.e., the graphical complexity of a language). The physics of notations framework provides a comprehensive set of visual design principles allowing to investigate the cognitive effectiveness of single languages – but also hybrid languages and hybrid process artifacts. This in turn could be used to derive hybrid representations with an increased cognitive support.

<u>Modeling</u>: The quality of process models has been extensively investigated in the field of process modeling. Accordingly, several guidelines aiming at enhancing the quality of process models have emerged. The Seven Process Modeling Guidelines (7PMG) [94] and the SEQUAL framework [95] comprise a set of quality aspects defining the criteria for understandable process models. These criteria have emerged as a result of several empirical studies investigating the reading

and the modeling of imperative process models, thus, their applicability remain questionable for declarative process models. Since hybrid representations combine languages from both paradigms, more research is required to *(1)* differentiate the guidelines which can be applicable to languages from both paradigms (e.g., verb-object naming of activities, number of elements in the model) and *(2)* derive new guidelines covering the aspects specific to declarative languages (e.g., the placement of entry-point and exit-point activities in a declarative process model). Afterwards, it is necessary to evaluate the applicability of these guidelines on hybrid representations and refine them accordingly to fit the intended purpose. Considering the variety of combinations of languages and artifacts proposed in the literature, the new guidelines should cover the different classes of HBPRs presented in the descriptive taxonomy (cf. Section 5.5).

The modeling of HBPRs is also constrained by the quality of the tools supporting process modeling. In this context, it is important to guide users toward using what is best in a context specific manner. By learning from users' behavior and the contextual information available at run-time, an adaptive system can be developed to provide a set of recommendations allowing to enhance the interactions with the HBPR. In this direction, initiatives have been made in the field of process modeling to discern the different phases associated with the modeling of BPMN process models (i.e., problem understanding, method finding, modeling and reconciliation) based on eye tracking and user interaction data [96]. Hence, similar approaches could be developed to identify the features defining the different modeling phases of a hybrid representation, which in turn could be used to provide a phase-specific modeling support at run-time.

<u>Evaluation:</u> Process modeling languages have been widely evaluated with regards to their understandability (e.g., [97, 98, 99]), maintainability (e.g., [100]) and modeling (e.g., [101]). However, when it comes to hybrid approaches, there is not much empirical work yet. Indeed most of the existing empirical studies are limited to the understandability of hybrid process artifacts, mainly those combining a process model with a textual annotation or a guided simulation, therefore, it is necessary to extend the evaluations to cover all the other classes of hybrid representations. Empirical evaluations should not be limited to the understandability of hybrid representations but should also cover other perspectives such as the modeling, the maintainability and the communication support offered by hybrid representations. The evaluation of the process highlighter presented in [48] is one among the few studies investigating the modeling using a HBPR and reporting some modeling patterns, even though, the provided insights remain of exploratory nature. To understand the way users engage with modeling tasks using HBPRs, more confirmatory studies are required. In this vein, HBPRs could be compared in terms of the perceived quality of process models and their alignment with the process specifications. In addition, physio-psychological measures can be used to estimate the cognitive load associated with the use of the different representations.

The support for better maintainability is another aspect to be investigated. The empirical evaluation of the maintainability of the TDM [83] is a starting point in this direction. However, due to the limited number of participants and the subjectivity of the used measures, the results cannot be generalized. Besides that, the maintainability of other classes of hybrid representations has to be evaluated as well. Part of the literature (cf. Section 5.3) claims the maintainability support offered by hybrid representations, however, little is known about the proper approach to maintain overlapping artifacts where certain information might be redundant.

The communication support offered by hybrid representations is another aspect to be investigated in the literature especially when dealing with hybrid process artifacts. In that respect, it is important to investigate the ability of hybrid process artifacts to bridge the communication gap between domain experts and IT specialists. An existing study hints towards this kind of support [23]. However, due to the lack of participants, no strong inferential statistics could be made. An in-depth understanding of this aspect would have a strong impact on the development of PAIS systems.

A large portion of existing empirical evaluations use students as subjects to evaluate their approaches, which, in turn, limits the validity of the results to academia. However, in order to be able to generalize the obtained results, it is crucial to go beyond academia and explore the use of HBPRs in industrial and administrative settings. Among the empirical studies to be cited in this direction is the study by [43], where the authors explored the understandability of HBPRs in both academic and administrative settings. In this work, the authors were able to spot considerable differences w.r.t the interactions of students and municipal employees with HBPRs. The results provide preliminary insights which can serve as a basis for future research in this direction.

A great variety of measurements are deployed to assess the interaction of humans with software artifacts. From basic questionnaires to advanced technologies (e.g., eye tracking, electroencephalography) these techniques have proven their efficacy in several user experience studies (e,g., [102]. When considering HBPRs, the deployment of such measurements remains limited, except for a few studies which report eye-tracking related insights (e.g., [46, 44]), the use of physio-psychological measurements to evaluate HBPRs remains very limited. Understanding the human cognitive processes and discerning the different strategies and patterns when reading, modeling and maintaining HBPRs is therefore vital to develop robust hybrid representations.

7. Threats to Validity

The validity of this SLR is subject to some threats particularly related to completeness, selection bias and the reliability of the automated tools used through the data collection and analysis. To ensure the search completeness, generic keywords were appended to the search string after a series of string refinements and search iterations. Moreover, the results of the backward search and forward search allowed identifying the extra articles that were not identified during the main literature search. To avoid any selection bias the articles were selected systematically according to a set of clear inclusion and exclusion criteria. Nevertheless, slight selection bias might be subjectivity introduced during the initial selection process as some articles did not clearly describe and motivate the aim of their approaches. Furthermore, articles published after September, 2018 were not covered by the search as the literature search was completed by that time. The search engines deployed to retrieve relevant literature constitute another potential threat to validity. With this regard, a simple and comprehensive search string was formulated using a single logical operator as some search engines do not support complex queries. In addition, the search process was fragmented to cover each publication venue individually in order to ensure a consistent search through all the relevant venues. Finally, the backward search was performed automatically by extracting and parsing the references cited in each of the selected articles, which in turn, were used to generate the graph visualization depicting the different research lines. A potential risk of this approach is associated with the inability to parse some references which can impact of the validity of the presented visualization.

8. Conclusion

This work proposes a conceptual framework and summarizes the outcome of a SLR about HBPRs. At a first stage a unified terminology is defined and the characteristics discerning hybrid languages and hybrid process artifacts are presented. Afterward a SLR is conducted to explore the existing literature. The analysis of the SLR findings allowed identifying the characteristics of existing HBPRs and the motivations driving their emergence. The data extracted from the literature allowed the identification of two research lines i.e., one aiming at extending and supporting declarative languages and another aiming at integrating business rules in business processes. For both research lines, the underlying publications were scrutinized closely. To this end, the combined representations were analyzed and grouped to derive a descriptive taxonomy. In addition, the maturity of the proposed approaches was profoundly examined. In addition, the common application domains where the use of these approaches is beneficial were identified and presented at the analysis.

The discussion of the findings reveled important insights about the results of the literature research, and provided a comprehensive research agenda tracing out the directions for future work while considering each phase of the HBPR development-cycle.

Finally, The overall contributions of this study allowed developing a deepened understanding of HBPRs. The outcome of this study will contribute to the development of new HBPRs and will serve as a basis to be systematically updated with upcoming research.

References

[1] J. Becker, M. Rosemann, C. von Uthmann, Guidelines of Business Process Modeling (2000) 30–49arXiv:9780201398298, doi:10.1007/3-540-45594-9_3.
URL http://link.springer.com/10.1007/3-540-45594-9_3

[2] J. Pinggera, T. Porcham, S. Zugal, B. Weber, LiProMo-Literate process modeling, CEUR Workshop Proceedings 855 (2012) 163–170.

[3] B. Axenath, E. Kindler, V. Rubin, AMFIBIA: A meta-model for the integration of business process modelling aspects, International Journal on Business Process Integration and Management 2 (2) (2007) 120–131.

[4] M. Dumas, W. M. P. van der Aalst, A. H. M. ter Hofstede (Eds.), Process-Aware Information Systems, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2005. doi:10.1002/0471741442.
URL http://doi.wiley.com/10.1002/0471741442

[5] W. M. P. van der Aalst, Process-Aware Information Systems: Lessons to Be Learned from Process Mining, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 1–26. doi:10.1007/978-3-642-00899-3_1.
URL https://doi.org/10.1007/978-3-642-00899-3_1

[6] O. M. G. OMG, Business process modeling notation v 2.0.
URL https://www.omg.org/spec/BPMN/2.0/

[7] T. Murata, Petri nets: Properties, analysis and applications, Proceedings of the IEEE 77 (4) (1989) 541–580. doi:10.1109/5.24143.

[8] M. Pesic, H. Schonenberg, W. M. van der Aalst, DECLARE: Full Support for Loosely-Structured Processes, in: 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), IEEE, 2007, pp. 287–287. doi:10.1109/EDOC.2007.14.
URL http://ieeexplore.ieee.org/document/4384001/

[9] T. T. Hildebrandt, R. R. Mukkamala, Declarative Event-Based Workflow as Distributed Dynamic Condition Response Graphs, Electronic Proceedings in Theoretical Computer Science 69 (2011) 59–73. arXiv:1110.4161, doi:10.4204/EPTCS.69.5.
URL http://arxiv.org/abs/1110.4161v1

[10] D. Fahland, D. Lübke, J. Mendling, H. Reijers, B. Weber, M. Weidlich, S. Zugal, Declarative versus Imperative Process Modeling Languages: The Issue of Understandability, in: Proceedings of EMMSAD, Springer, 2009, pp. 353–366.

[11] S. Goedertier, J. Vanthienen, F. Caron, Declarative business process modelling: Principles and modelling languages, Enterp. Inf. Syst. 9 (2) (2015) 161–185. doi:10.1080/17517575.2013.830340.
URL http://dx.doi.org/10.1080/17517575.2013.830340

[12] M. Reichert, B. Weber, Enabling Flexibility in Process-Aware Information Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. doi:10.1007/978-3-642-30409-5.
URL http://link.springer.com/10.1007/978-3-642-30409-5

[13] H. A. Reijers, T. Slaats, C. Stahl, Declarative modeling - An academic dream or the future for BPM?, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 8094 LNCS (2013) 307–322. doi:10.1007/978-3-642-40176-3_26.

[14] M. Pesic, Constraint-Based Workflow Management Systems : Shifting Control to Users, phd thesis Edition, Eindhoven University of Technology, 2008.

[15] B. Weber, H. A. Reijers, S. Zugal, W. Wild, The declarative approach to business process execution: An empirical test, in: P. van Eck, J. Gordijn, R. Wieringa (Eds.), Advanced Information Systems Engineering, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 470–485.

[16] S. Zugal, J. Pinggera, B. Weber, Toward enhanced life-cycle support for declarative processes, Journal of Software: Evolution and Process 24 (3) (2012) 285–302. doi:10.1002/smr.554.
URL http://doi.wiley.com/10.1002/smr.554

[17] T. R. Green, Cognitive dimensions of notations, People and computers V (1989) 443–460.

[18] T. R. G. Green, M. Petre, Usability analysis of visual programming environments: a 'cognitive dimensions' framework, Journal of Visual Languages & Computing 7 (2) (1996) 131–174.

[19] F. Chen, J. Zhou, Y. Wang, K. Yu, S. Z. Arshad, A. Khawaji, D. Conway, Robust multimodal cognitive load measurement, Springer, 2016.

[20] G. A. Miller, The magical number seven, plus or minus two: Some limits on our capacity for processing information., Psychological review 63 (2) (1956) 81.

[21] H. A. López, S. Debois, T. T. Hildebrandt, M. Marquard, The process highlighter: From texts to declarative processes and back., in: BPM (Dissertation/Demos/Industry), 2018, pp. 66–70.

[22] S. Debois, T. Hildebrandt, M. Marquard, T. Slaats, Hybrid Process Technologies in the Financial Sector, Proceedings of the Industry Track at the 13th International Conference on Business Process Management 2015 4 (2) (2015) 107–119.

[23] S. Zugal, J. Pinggera, B. Weber, The impact of testcases on the maintainability of declarative process models, in: T. Halpin, S. Nurcan, J. Krogstie, P. Soffer, E. Proper, R. Schmidt, I. Bider (Eds.), Enterprise, Business-Process and Information Systems Modeling, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 163–177.

[24] T. Slaats, D. M. M. Schunselaar, F. M. Maggi, H. A. Reijers, The Semantics of Hybrid Process Models, 2016, pp. 531–551. doi:10.1007/978-3-319-48472-3_32.
URL http://link.springer.com/10.1007/978-3-319-48472-3_32

[25] S. W. Sadiq, M. E. Orlowska, W. Sadiq, Specification and validation of process constraints for flexible workflows, Information Systems 30 (5) (2005) 349–378. doi:10.1016/j.is.2004.05.002.
URL http://linkinghub.elsevier.com/retrieve/pii/S0306437904000456

[26] S. Steinau, A. Marrella, K. Andrews, F. Leotta, M. Mecella, M. Reichert, Dalec: a framework for the systematic evaluation of data-centric approaches to process management software, Software & Systems Modeling 18 (4) (2019) 2679–2716. doi:10.1007/s10270-018-0695-0.
URL https://doi.org/10.1007/s10270-018-0695-0

[27] M. Weske, Business process management–concepts, languages, architectures, verlag (2007).

[28] P. Soffer, M. Kaner, Y. Wand, Towards understanding the process of process modeling: Theoretical and empirical considerations, Lecture Notes in Business Information Processing: Business Process Management Workshops - BPM 2011 99 (2012) 357–369. doi:10.1007/978-3-642-28108-2_35.

[29] S. Zugal, Applying Cognitive Psychology for Improving the Creation, Understanding and Maintenance of Business Process Models.

[30] C. W. Morris, Foundations of the theory of signs, International encyclopedia of unified science, University of Chicago Press, 1938.

[31] V. Fromkin, R. Rodman, N. Hyams, An Introduction to Language, Cengage Learning, 2013.

[32] G. Guizzardi, Ontological foundations for structural conceptual models, Vol. 015, 2005. doi:10.1039/tf9625800697.
URL http://doc.utwente.nl/50826

[33] T. Winograd, Frame representations and the declarative/procedural controversy, in: Representation and understanding, Elsevier, 1975, pp. 185–210.

[34] J. W. Lloyd, Practical advantages of declarative programming., in: M. Alpente, R. Barbuti, I. Ramos (Eds.), GULP-PRODE '94, 1994 Joint Conference on Declarative Programming, Proceedings, Vol. 1, 1994, pp. 3–17.

[35] P. Van-Roy, S. Haridi, Concepts, techniques, and models of computer programming, MIT press, 2004.

[36] D. Fahland, J. Mendling, H. A. Reijers, B. Weber, M. Weidlich, S. Zugal, Declarative versus Imperative Process Modeling Languages: The Issue of Maintainability, in: S. Rinderle-Ma, S. Sadiq, F. Leymann (Eds.), Business Process Management Workshops, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 477–488.

[37] S. Sadiq, W. Sadiq, M. Orlowska, Pockets of Flexibility in Workflow Specification, 2001, pp. 513–526. doi:10.1007/3-540-45581-7_38.
URL http://link.springer.com/10.1007/3-540-45581-7_38

[38] Object Management Group OMG, Case Management Model and Notation Version 1.0.
URL http://www.omg.org/spec/CMMN/1.0/PDF

[39] G. J. Nalepa, A. Lig\keza, K. Kaczor, Overview of Knowledge Formalization with XTT2 Rules, in: N. Bassiliades, G. Governatori, A. Paschke (Eds.), Rule-Based Reasoning, Programming, and Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 329–336.

[40] W. van der Aalst, A. ter Hofstede, YAWL: yet another workflow language, Information Systems 30 (4) (2005) 245–275. doi:10.1016/j.is.2004.02.002.
URL http://linkinghub.elsevier.com/retrieve/pii/S0306437904000304

[41] G. Keller, M. Nüttgens, A. W. Scheer, Semantische Prozessmodellierung auf der Grundlage "ereignisgesteuerter Prozessketten (EPK)", Institut für Wirtschaftsinformatik Saarbrücken: Veröffentlichungen des Instituts für Wirtschaftsinformatik, Inst. für Wirtschaftsinformatik, 1992.

[42] O. Management Group, An OMG ® Semantics of Business Vocabulary and Business Rules TM Publication-Content-Model-for-SBVR.xml (May).
URL https://www.omg.org/spec/SBVR/About-SBVR/{#}specification-metadata

[43] A. Abbad Andaloussi, A. Burattin, T. Slaats, A. C. M. Petersen, T. T. Hildebrandt, B. Weber, Exploring the understandability of a hybrid process design artifact based on dcr graphs, in: I. Reinhartz-Berger, J. Zdravkovic, J. Gulden, R. Schmidt (Eds.), Enterprise, Business-Process and Information Systems Modeling, Springer International Publishing, Cham, 2019, pp. 69–84.

[44] A. A. Andaloussi, T. Slaats, A. Burattin, T. Thomas, B. Weber, Evaluating The Understandability of Hybrid Process Model Representations Using Eye Tracking : First Insights 1–6.

[45] J. De Smedt, J. De Weerdt, E. Serral, J. Vanthienen, Discovering hidden dependencies in constraint-based declarative process models for improving understandability, Information Systems 74 (2018) 40–52. doi:10.1016/j.is.2018.01.001.
URL https://linkinghub.elsevier.com/retrieve/pii/S0306437916304732

[46] W. Wang, M. Indulska, S. Sadiq, B. Weber, Effect of Linked Rules on Business Process Model Understanding, 2017, pp. 200–215. doi:10.1007/978-3-319-65000-5_12.
URL http://link.springer.com/10.1007/978-3-319-65000-5_12

[47] J. Recker, N. Safrudin, M. Rosemann, How Novices Model Business Processes, in: R. Hull, J. Mendling, S. Tai (Eds.), Business Process Management, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 29–44.

[48] A. Abbad Andaloussi, J. Buch-Lorentsen, H. A. López, T. Slaats, B. Weber, Exploring the Modeling of Declarative Processes Using a Hybrid Approach (2019) 1–15.

[49] W. M. Van Der Aalst, Process mining: Discovery, Conformance, and Enhancement of Business processes, Springer International Publishing, 2010.

[50] M. Westergaard, T. Slaats, Mixing Paradigms for More Comprehensible Models, 2013, pp. 283–290. doi:10.1007/978-3-642-40176-3_24.
URL http://link.springer.com/10.1007/978-3-642-40176-3_24

[51] G. De Giacomo, M. Dumas, F. M. Maggi, M. Montali, Declarative Process Modeling in BPMN, 2015, pp. 84–100. doi:10.1007/978-3-319-19069-3_6.
URL http://link.springer.com/10.1007/978-3-319-19069-3_6

[52] J. De Smedt, J. De Weerdt, J. Vanthienen, G. Poels, Mixed-paradigm process modeling with intertwined state spaces, Business and Information Systems Engineering 58 (1) (2016) 19–29. doi:10.1007/s12599-015-0416-y.

[53] U. Frank, Multi-perspective enterprise modeling: foundational concepts, prospects and future research challenges, Software & Systems Modeling 13 (3) (2014) 941–962. doi:10.1007/s10270-012-0273-9.
URL https://doi.org/10.1007/s10270-012-0273-9

[54] F. Dengler, D. Vrandečić, Wiki-Based Maturing of Process Descriptions, 2011, pp. 313–328. doi:10.1007/978-3-642-23059-2_24.
URL http://link.springer.com/10.1007/978-3-642-23059-2_24

[55] B. Kitchenham, Guidelines for performing Systematic Literature Reviews in Software Engineering 33 (Keele, UK, Keele University).

[56] D. Budgen, P. Brereton, Performing systematic literature reviews in software engineering, Proceeding of the 28th international conference on Software engineering - ICSE '06 (2006) 1051arXiv:1304.1186,

doi:10.1145/1134285.1134500.
URL http://portal.acm.org/citation.cfm?doid=1134285.1134500

[57] J. Webster, R. T. Watson, Analyzing the Past to Prepare for the Future: Writing a Literature Review., MIS Quarterly 26 (2) (2002) xiii – xxiii. arXiv:02767783, doi:10.1.1.104.6570.

[58] R. Lu, S. Sadiq, G. Governatori, On managing business processes variants, Data and Knowledge Engineering 68 (7) (2009) 642–664. doi:10.1016/j.datak.2009.02.009.
URL http://dx.doi.org/10.1016/j.datak.2009.02.009

[59] B. Zimmermann, M. Doehring, Patterns for flexible BPMN workflows, Proceedings of the 16th European Conference on Pattern Languages of Programs - EuroPLoP '11 (2011) 1–9doi:10.1145/2396716.2396723.
URL http://dl.acm.org/citation.cfm?doid=2396716.2396723

[60] K. Hinkelmann, Business process flexibility and decision-aware modeling—the knowledge work designer, in: D. Karagiannis, H. C. Mayr, J. Mylopoulos (Eds.), Domain-Specific Conceptual Modeling: Concepts, Methods and Tools, Springer International Publishing, Cham, 2016, pp. 397–414. doi:10.1007/978-3-319-39417-6_18.
URL https://doi.org/10.1007/978-3-319-39417-6_18

[61] D. M. M. Schunselaar, T. Slaats, F. M. Maggi, H. A. Reijers, W. M. P. van der Aalst, Mining Hybrid Business Process Models: A Quest for Better Precision, in: Introducing Information Management, Vol. 1, Springer International Publishing, 2018, pp. 190–205. doi:10.1007/978-3-319-93931-5_14.
URL http://link.springer.com/10.1007/978-3-319-93931-5_14

[62] J. De Smedt, J. De Weerdt, J. Vanthienen, Fusion Miner: Process discovery for mixed-paradigm models, Decision Support Systems 77 (2015) 123–136. doi:10.1016/j.dss.2015.06.002.
URL https://linkinghub.elsevier.com/retrieve/pii/S0167923615001165

[63] J. De Smedt, J. De Weerdt, J. Vanthienen, Multi-Paradigm Process Mining: Retrieving Better Models by Combining Rules and Sequences, Ssrndoi:10.2139/ssrn.2464066.

[64] C. Ayora, V. Torres, B. Weber, M. Reichert, V. Pelechano, Vivace: A framework for the systematic evaluation of variability support in process-aware information systems, Information and Software Technology 57 (2015) 248–276.

[65] M. Milanovic, D. Gasevic, L. Rocha, Modeling Flexible Business Processes with Business Rule Patterns, in: 2011 IEEE 15th International Enterprise Distributed Object Computing Conference, IEEE, 2011, pp. 65–74. doi:10.1109/EDOC.2011.25.
URL http://ieeexplore.ieee.org/document/6037561/

[66] T. van Eijndhoven, M.-E. Iacob, M. L. Ponisio, Achieving Business Process Flexibility with Business Rules, in: 2008 12th International IEEE Enterprise Distributed Object Computing Conference, IEEE, 2008, pp. 95–104. doi:10.1109/EDOC.2008.23.
URL http://ieeexplore.ieee.org/document/4634761/

[67] J. De Smedt, J. De Weerdt, E. Serral, J. Vanthienen, Improving Understandability of Declarative Process Models by Revealing Hidden Dependencies, in: S. Nurcan, P. Soffer, M. Bajec, J. Eder (Eds.), Advanced Information Systems Engineering, Springer International Publishing, Cham, 2016, pp. 83–98.

[68] R. Cheng, S. Sadiq, M. Indulska, Framework for Business Process and Rule Integration: A Case of BPMN and SBVR, 2011, pp. 13–24. doi:10.1007/978-3-642-21863-7_2.
URL http://link.springer.com/10.1007/978-3-642-21863-7_2

[69] S. Zugal, J. Pinggera, B. Weber, Creating Declarative Process Models Using Test Driven Modeling Suite, in: S. Nurcan (Ed.), IS Olympics: Information Systems in a Diverse World, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 16–32.

[70] M. Marquard, M. Shahzad, T. Slaats, Web-Based Modelling and Collaborative Simulation of Declarative Processes, in: H. R. Motahari-Nezhad, J. Recker, M. Weidlich (Eds.), Business Process Management, Springer International Publishing, Cham, 2016, pp. 209–225.

[71] T. Graml, R. Bracht, M. Spies, Patterns of business rules to enable agile business processes, in: 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), 2007, pp. 365–365. doi:10.1109/EDOC.2007.35.

[72] B. Sapkota, M. v. Sinderen, Exploiting rules and processes for increasing flexibility in service composition, in: 2010 14th IEEE International Enterprise Distributed Object Computing Conference Workshops, 2010, pp. 177–185. doi:10.1109/EDOCW.2010.44.

[73] K. Kluza, K. Kaczor, G. J. Nalepa, Enriching Business Processes with Rules Using the Oryx BPMN Editor, in: L. Rutkowski, M. Korytkowski, R. Scherer Rafałand Tadeusiewicz, L. A. Zadeh, J. M. Zurada (Eds.), Artificial Intelligence and Soft Computing, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 573–581.

[74] M. Milanovic, D. Gasevic, Towards a language for rule-enhanced business process modeling, in: 2009 IEEE International Enterprise Distributed Object Computing Conference, 2009, pp. 64–73. doi:10.1109/EDOC.2009.12.

[75] W. M. P. van der Aalst, M. Adams, A. H. M. ter Hofstede, M. Pesic, H. Schonenberg, Flexibility as a Service, 2009, pp. 319–333. doi:10.1007/978-3-642-04205-8_27.
URL http://link.springer.com/10.1007/978-3-642-04205-8_27

[76] P. Mangan, S. Sadiq, On Building Workflow Models for Flexible Processes, Journal Australian Computer Science Communications 24 (2) (2002) 103–109. doi:http://doi.acm.org/10.1145/563932.563918.

[77] P. J. Mangan, S. Sadiq, A constraint specification approach to building flexible workflows, Journal of Research and Practice in Information Technology 35 (1) (2003) 21–39.

[78] E. Mickevičiūtė, R. Butleris, Towards the Combination of BPMN Process Models with SBVR Business Vocabularies and Rules, 2013, pp. 114–121. doi:10.1007/978-3-642-41947-8_11.
URL http://link.springer.com/10.1007/978-3-642-41947-8_11

[79] K. Kluza, G. J. Nalepa, Formal Model of Business Processes Integrated with Business Rules, Information Systems Frontiersdoi:10.1007/s10796-018-9826-y.
URL http://link.springer.com/10.1007/s10796-018-9826-y

[80] M. Scaife, Y. Rogers, External cognition: how do graphical representations work?, International Journal of Human-Computer Studies 45 (2) (1996) 185–213. doi:10.1006/ijhc.1996.0048.
URL http://linkinghub.elsevier.com/retrieve/pii/S1071581996900488

[81] M. Pesic, H. Schonenberg, W. van der Aalst, The Declare Service, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 327–343. doi:10.1007/978-3-642-03121-2_12.
URL https://doi.org/10.1007/978-3-642-03121-2_12

[82] M. Westergaard, T. Slaats, CPN tools 4: A process modeling tool combining declarative and imperative paradigms, CEUR Workshop Proceedings 1021.

[83] S. Zugal, C. Haisjackl, J. Pinggera, B. Weber, Empirical evaluation of test driven modeling, Int. J. Inf. Syst. Model. Des. 4 (2) (2013) 23–43. doi:10.4018/jismd.2013040102.
URL http://dx.doi.org/10.4018/jismd.2013040102

[84] I. Seeber, R. Maier, B. Weber, CoPrA: A Process Analysis Technique to Investigate Collaboration in Groups, in: 2012 45th Hawaii International Conference on System Sciences, IEEE, 2012, pp. 363–372. doi:10.1109/HICSS.2012.184.
URL http://ieeexplore.ieee.org/document/6148651/

[85] J. Pinggera, S. Zugal, B. Weber, Investigating the process of process modeling with cheetah experimental platform- Tool paper, CEUR Workshop Proceedings 603 (2010) 13–18.

[86] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, J. van de Weijer, Eye Tracking: A comprehensive guide to methods and measures, OUP Oxford, 2011.
URL https://books.google.dk/books?id=5rIDPV1EoLUC

[87] R. N. Meghanathan, C. van Leeuwen, A. R. Nikolaev, Fixation duration surpasses pupil size as a measure of mem-

ory load in free viewing, Frontiers in Human Neuroscience 8. doi:10.3389/fnhum.2014.01063.
URL http://journal.frontiersin.org/article/10.3389/fnhum.2014.01063/abstract

[88] A. Poole, L. J. Ball, Eye tracking in human-computer interaction and usability research: Current status and future, in: Encyclopedia of Human-Computer Interaction. Pennsylvania: Idea Group, Inc, 2005.

[89] J. Gulden, A. Burattin, A. A. Andaloussi, B. Weber, From analytical purposes to data visualizations: a decision process guided by a conceptual framework and eye tracking, Software & Systems Modelingdoi:10.1007/s10270-019-00742-z.
URL https://doi.org/10.1007/s10270-019-00742-z

[90] M. Zur Muehlen, M. Indulska, Modeling languages for business processes and business rules: A representational analysis, Information systems 35 (4) (2010) 379–390.

[91] O. I. Lindland, G. Sindre, A. Solvberg, Understanding quality in conceptual modeling, IEEE software 11 (2) (1994) 42–49.

[92] K. Figl, J. Mendling, M. Strembeck, J. Recker, On the cognitive effectiveness of routing symbols in process modeling languages, in: W. Abramowicz, R. Tolksdorf (Eds.), Business Information Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 230–241.

[93] D. Moody, The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering, IEEE Transactions on Software Engineering 35 (6) (2009) 756–779. doi:10.1109/TSE.2009.67.

[94] J. Mendling, H. A. Reijers, W. M. P. van der Aalst, Seven process modeling guidelines (7pmg), Inf. Softw. Technol. 52 (2) (2010) 127–136. doi:10.1016/j.infsof.2009.08.004.
URL http://dx.doi.org/10.1016/j.infsof.2009.08.004

[95] J. Krogstie, Model-based development and evolution of information systems: A Quality Approach, Springer Science & Business Media, 2012.

[96] A. Burattin, M. Kaiser, M. Neurauter, B. Weber, Learning process modeling phases from modeling interactions and eye tracking data, Data & Knowledge Engineering 121 (2019) 1–17.

[97] C. Haisjackl, I. Barba, S. Zugal, P. Soffer, I. Hadar, M. Reichert, J. Pinggera, B. Weber, Understanding declare models: strategies, pitfalls, empirical results, Software and Systems Modeling 15 (2) (2016) 325–352. doi:10.1007/s10270-014-0435-z.
URL https://doi.org/10.1007/s10270-014-0435-z

[98] C. Haisjackl, P. Soffer, S. Y. Lim, B. Weber, How do humans inspect BPMN models: an exploratory study, Software and Systems Modeling 17 (2) (2018) 655–673. doi:10.1007/s10270-016-0563-8.
URL https://doi.org/10.1007/s10270-016-0563-8

[99] K. Figl, Comprehension of procedural visual business process models, Business & Information Systems Engineering 59 (1) (2017) 41–67. doi:10.1007/s12599-016-0460-2.
URL https://doi.org/10.1007/s12599-016-0460-2

[100] F. Garcia, F. Ruiz, C. A. Visaggio, A proposal and empirical validation of metrics to evaluate the maintainability of software process models, in: 2006 IEEE Instrumentation and Measurement Technology Conference Proceedings, 2006, pp. 1093–1097. doi:10.1109/IMTC.2006.328377.

[101] J. Claes, I. Vanderfeesten, F. Gailly, P. Grefen, G. Poels, The structured process modeling theory (spmt) a cognitive view on why and how modelers benefit from structuring the process of process modeling, Information Systems Frontiers 17 (6) (2015) 1401–1425. doi:10.1007/s10796-015-9585-y.
URL https://doi.org/10.1007/s10796-015-9585-y

[102] A. Bojko, Using eye tracking to compare web page designs: A case study, J. Usability Studies 1 (3) (2006) 112–120.
URL http://dl.acm.org/citation.cfm?id=2835663.2835665