Software and Systems Modeling
https://doi.org/10.1007/s10270-023-01110-8

REGULAR PAPER

Business process modeling language selection for research modelers

Siamak Farshidi[1] · Izaak Beer Kwantes[1] · Slinger Jansen[1,2]

Received: 28 September 2022 / Revised: 18 April 2023 / Accepted: 27 April 2023

Abstract

Business process modeling is a crucial aspect of domains such as Business Process Management and Software Engineering. The availability of various BPM languages in the market makes it challenging for process modelers to select the best-fit BPM language for a specific process modeling task. A decision model is necessary to systematically capture and make scattered knowledge on BPM languages available for reuse by process modelers and academics. This paper presents a decision model for the BPM language selection problem in research projects. The model contains mappings of 72 BPM features to 23 BPM languages. We validated and refined the decision model through 10 expert interviews with domain experts from various organizations. We evaluated the efficiency, validity, and generality of the decision model by conducting four case studies of academic research projects with their original researchers. The results confirmed that the decision model supports process modelers in the selection process by providing more insights into the decision process. Based on the empirical evidence from the case studies and domain expert feedback, we conclude that having the knowledge readily available in the decision model supports academics in making more informed decisions that align with their preferences and prioritized requirements. Furthermore, the captured knowledge provides a comprehensive overview of BPM languages, features, and quality characteristics that other researchers can employ to tackle future research challenges. Our observations indicate that BPMN is a commonly used modeling language for process modeling. Therefore, it is more sensible for academics to explain why they did not select BPMN than to discuss why they chose it for their research project(s).

Keywords Business process modeling language selection · Decision model · Multi-criteria decision-making · Decision support system · Case study research

1 Introduction

Business process modeling (BPM) is an essential aspect of business process management [1], organizational design[2], and the development of information systems [3]. It is a prerequisite for leveraging the benefits of business process improvement efforts [4, 5]. Explicit representations of business processes serve as a starting point for the design of an IT solution in the information system (IS) engineering community [6], as a means to improve organizational performance through business process re-engineering (BPR) [7], or to support their documentation in business process management initiatives[8, 9]. Business process modeling is not limited to specific domains but is widely used across many enterprises [10]. Surveys conducted by *BPTrends* in 2011 [11] and 2020 [12] indicate that business process management and modeling are used in various domains, such as manufacturing, healthcare, financial services, and education, among others. The widespread adoption of business process modeling across various approaches has underscored the significance of this technique [13].

Business process models, also known as process models, are an abstract representation of a process [14, 15]. These models systematically describe (some part of) a business process, typically in the form of a series of activities [16]. The

Communicated by Dragan Milicev.

✉ Siamak Farshidi
s.farshidi@uu.nl

✉ Slinger Jansen
slinger.jansen@uu.nl

Izaak Beer Kwantes
i.b.kwantes@students.uu.nl

[1] Department of Information and Computer Science, Utrecht University, Buys Ballotgebouw, 3584 CC Utrecht, The Netherlands

[2] Lappeenranta University of Technology, Lappeenranta, Finland

Published online: 29 May 2023

practice of business process modeling aims to highlight the structure of a business process and describe process characteristics, resources used, the sequence of activities, and their relationships [17]. In other words, business process modeling is defined as *a means of representing the business activities, the information flow and decision logic in business processes* [18] with the intention of producing a cohesive model of the required behavior[19]. Business process modeling offers many benefits, such as facilitating understanding, simplifying complexity, clear communication to multiple stakeholders, performance improvements, and reuse for other business processes [20].

Business process modeling is a popular practice, leading to a surge in the number of methods and tools available to support it [21]. Among these methods are BPMN [22], UML-AD [23], Petri Net [24], EPC [25], and YAWL [26]. This proliferation of BPM languages poses a significant challenge in the business process modeling domain, as it is difficult to choose the most suitable language for a given modeling task [18, 21, 27–34]. For instance, a Google Scholar search using the “business process modeling” query yields over 2.5 million results.

Each BPM language has its specifications and characteristics [18, 31]. These languages come from different traditions [14], focus on different aspects of business process modeling [19], and are appropriate for various domains and purposes [29]. Compatibility requirements [35], serialization format [36], modeling perspectives covered [37], and the degree of formality [38] are some of the characteristics to be considered when selecting a BPM language. However, no single “best” BPM language is suitable for all scenarios. This situation creates difficulties for academics [17], modelers [39], and decision makers [40] in selecting an appropriate BPM language for their modeling problem(s). It is particularly relevant to provide adequate guidance [41] as process documentation projects rely heavily on novice and expert modelers [42]. Selecting an unsuitable BPM language may lead to obstacles, such as a decrease in understandability, an increase in complexity, the inability to describe processes realistically [29], and increased time consumption [21].

One of the primary challenges in BPM is selecting the most suitable language from a vast array of alternatives, each with numerous features, based on a set of requirements. This selection problem can be framed as a multi-criteria decision-making (MCDM) problem, which involves evaluating a significant number of decision alternatives based on multiple criteria that may conflict with one another [43]. While numerous BPM languages have unique characteristics, strengths, and weaknesses, knowledge about these languages is scattered across various sources, such as literature, documentation, websites, and domain expert experience. Currently, there is no comprehensive and accessible decision model published in the literature for selecting the most suitable BPM language for research projects. Therefore, this study aims to develop a decision model that systematically captures knowledge about BPM languages and concepts in a reusable and extendable format. To achieve this objective, a theoretical framework [44] is employed, and a six-step decision-making process proposed by Majumder [43] is followed to develop the decision model (See Fig. 1 and read more in Sect. 3).

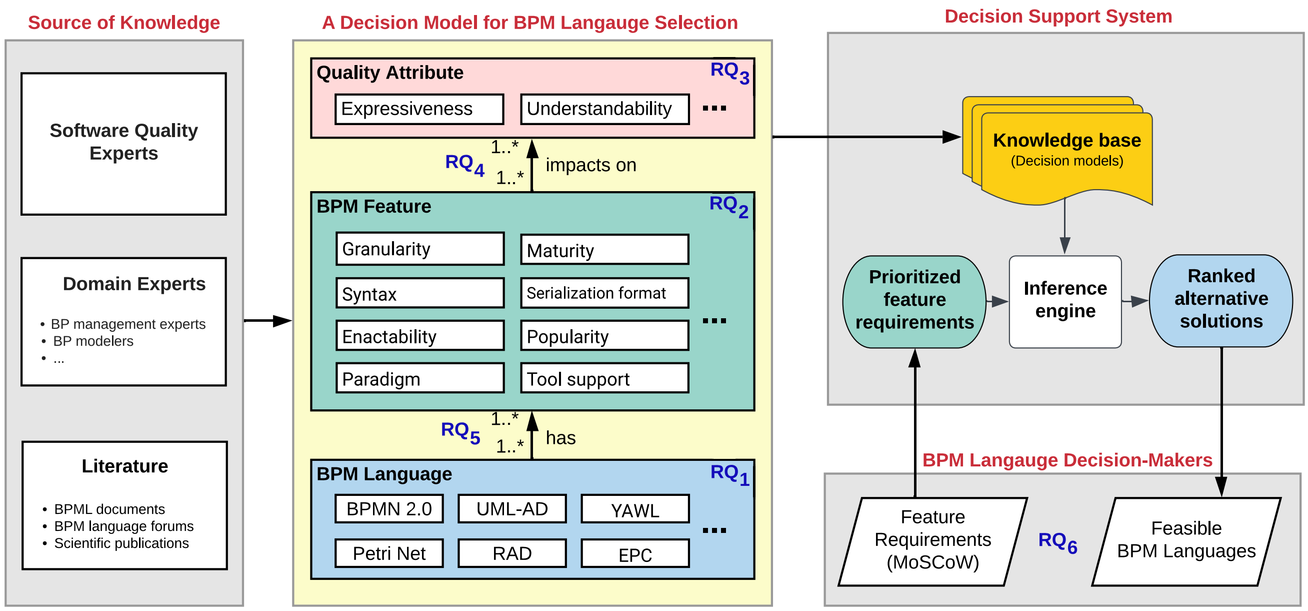


Fig. 1 Illustration of the framework’s main building blocks, adapted from [44, 45]. On the left are the sources of knowledge; in the middle, the proposed decision model for BPM language selection; and on the right, the DSS. Please also note the research questions marked in blue

The rest of the study is structured as follows: Sect. 2 formulates the BPM language selection problem as an MCDM problem and describes the research method based on the design science, expert interviews, snowballing literature review, document analysis, and exploratory theory-testing case studies. This study has the following contributions:

Contribution 1 Sect. 3 explains the integration of the captured tacit knowledge of domain experts through interviews and the explicit knowledge scattered among a wide range of literature, websites, documentation, and reports. The acquired knowledge is presented in the form of reusable knowledge that can support decision makers in understanding which BPM languages are currently available, their characteristics, what attributes can be used to evaluate their quality, and which features are fulfilled by which BPM languages.

Contribution 2 Sect. 4 describes four case studies performed in Germany and Australia to evaluate the effectiveness and usefulness of the decision model.

Contribution 3 Sect. 5 analyzes and compares the DSS results with the case study participants' shortlists of ranked feasible BPM languages. The degree to which the goal dimension of the artifact is met is determined based on the efficacy, validity, and generality metrics. The results show that the DSS recommends solutions the case study participants suggested for their modeling challenges. It is useful for evaluating and validating the BPM language selection decision.

Section 6 highlights the lessons learned, limitations, and threats to validity. Additionally, we argue how we have minimized the threats to the validity of the results. Section 7 positions the proposed approach in this study among other BPM language selection techniques in the literature. Finally, Sect. 8 summarizes the proposed approach, defends its novelty, and offers directions for future studies.

2 Research approach

This section presents the problem definition, indicates the study objective, and formulates the research questions. Furthermore, it elaborates on the mixed research method used in this study, based on design science. Multiple research methods can be combined to understand the studied phenomenon better, connecting complementary findings from the various qualitative and quantitative traditions [46]. The process of capturing, structuring, and organizing knowledge from multiple sources is known as knowledge acquisition [47], done through systematic literature review, expert interviews, document analysis, and case study research. This section elaborates on these knowledge acquisition techniques to answer the research questions and build a decision model for the BPM language selection problem in research projects.

2.1 Problem definition

Decision-making is defined as a process or a set of ordered activities concerning stages of problem identification, data collection alternatives definition, and selecting a shortlist of alternatives as feasible solutions based on the ranked preferences [48, 49]. In the modeling domain, selecting a suitable BPM language is not an uncommon problem (e.g., in [34, 50–52]). As aforementioned, every BPM language has its specificities or characteristics: compatibility requirements, serialization format, perspectives covered, the degree of formality, etc. Besides, a decision problem in the modeling domain is not addressed in the same way by all stakeholders. Information being part of the particular modeling context impacts the design and execution of a business process [53], as well as the motivation for selecting or excluding a certain BPM language [54]. The plethora of BPM languages available makes it difficult for scholars in the BPM area to select one of them [17], while it is often processed modelers with an academic background entrusted with the task of modeling business processes [55]. Furthermore, each BPM language comes from a different tradition [14] and is suitable for different business domains and purposes [29]. Additionally, no unique BPM language is the best option for all scenarios. Consequently, the preferences and judgments of one decision maker are expected to differ. Addressing such issues in building decision models forms the focal point of interest in multiple-criteria decision-making [45].

Multiple-criteria decision-making encompasses an app roach and set of techniques used to rank and evaluate alternative solutions based on a decision maker's perspective, as referenced by [56, 57]. The solutions may differ in how they achieve various objectives, and often no alternative is superior in all criteria, as stated by [58]. Multiple-criteria evaluation problems consist of a finite set of decision alternatives that are explicitly known, according to [43]. The problems of multiple objective mathematical programming, which often involve an infinite or large number of alternatives, are a subclass of MCDM problems, as noted by [59]. Multi-criteria decision-making methods are mathematical models that evaluate alternative solutions over multiple, often conflicting [60] criteria. Such mathematical models can be addressed by building decision models, which show feasible solutions based on the decision makers' preferences, providing a convenient and faster way for the user to make a decision [61]. In this study, we have defined the BPM language selection in research projects problem as a multi-criteria decision-making problem.

Let $Languages$ be a set of BPM languages, represented as $l_1, l_2, ..., l_{|Languages|}$ and $Features$ be a set of BPM features, represented as $f_1, f_2, ..., f_{|Features|}$. Each language $l \in Languages$ supports a subset of the features in $Features$. The objective is to find a set of suitable BPM languages, referred to as *Solutions*, that fulfill a set of BPM feature requirements called *Requirements*, where $Requirements \subseteq Features$. So, $Requirements = \{f_1, f_2, ..., f_n\} \subseteq Features$ and the goal is to find $Solutions = l \in Languages : Requirements \subseteq Features$.

To address the selection problem, an MCDM approach takes *Languages* and their corresponding *Features* as input, applies a weighting method to prioritize the *Features* based on decision makers' preferences to define the *Requirements*, and finally employs an aggregation method to rank the *Languages* and recommend *Solutions*. Therefore, the MCDM approach can be formulated as follows:

MCDM: *Languages*×*Features*×*Requirements* → *Solutions*

MCDM is a well-established approach for handling complex decision-making problems. However, a unique optimal solution for an MCDM problem, such as selecting BPM languages in research projects, often does not exist [62, 63]. In such cases, decision makers must employ their preferences to differentiate between the available solutions. However, this selection process can never be completely objective, as the human decision makers ultimately have to make the decisions [64]. It is, therefore, essential to acknowledge the subjectivity involved in the decision-making process and ensure that the chosen solution aligns with the project's overall goals. To address this challenge, MCDM methods can be used to identify the most preferred solution based on the decision makers' preferences and the criteria used to evaluate the available alternatives [63].

2.2 Research questions

This study's main research question (MRQ) has been defined as **MRQ:** How can knowledge regarding BPM languages be captured and organized systematically to support the academic decision maker with the BPM language selection process in research projects?

We decomposed the main research question into six additional research questions, each one to capture the knowledge required for the creation of a decision model for BPM language selection:

RQ_1 Which BPM languages should be considered in the decision model?

RQ_2 Which BPM concepts should be considered BPM features in the decision model?

RQ_3 Which quality attributes can be used to evaluate the BPM languages?

RQ_4 What are the impacts of the BPM features on the quality attributes of these BPM languages?

RQ_5 Which BPM languages support which BPM features?

RQ_6 Does the decision model effective and efficient to apply in real-world research projects?

2.3 Research methods

Recently, a framework and a decision support system (DSS) were designed and implemented for software engineers and other decision makers facing MCDM problems in software production [44, 65]. The framework is based on knowledge engineering theories. It provides guidance for building decision models for MCDM problems in software production using a six-step decision-making process: (1) identifying objectives, (2) selecting criteria, (3) selecting alternatives, (4) selecting a weighing method, (5) applying an aggregation method, and (6) making a decision based on the aggregated results, as proposed by Majumder [43].

We employed the framework to build multiple decision models [45, 64–68] to assist software engineers with a set of MCDM problems in software production. The framework provides guidelines to systematically capture and organize knowledge from different sources to build decision models for MCDM problems in software production [45]. Furthermore, we introduced a DSS [44] as a platform for building MCDM decision models based on the framework. Then, the decision models can be uploaded to the DSS's knowledge base to facilitate the decision-making process according to the requirements and preferences of the decision maker. Moreover, the DSS can be used over the entire lifecycle and coevolve its advice based on evolving requirements.

In this study, the framework modeled in Fig. 1 was used to build a decision model for the problem of selecting a BPM language for research projects. Initially, 23 BPM languages and 72 BPM features were identified through a literature review and ten semi-structured domain expert interviews. Based on an analysis of BPM language literature, documents, and the experts' tacit knowledge, the mapping between BPM languages and BPM features was identified. Additionally, a systematic literature review mapped BPM features to quality attributes extracted from scientific articles. The impacts of BPM languages on these quality attributes were calculated based on input from three domain experts. Following the guidelines of the framework [44] and the knowledge acquired, a decision model was built for the BPM language selection problem. Finally, the decision model was evaluated through four real-world case studies, and the mapping between research methods and research questions is shown in Table 1.

2.3.1 Design science

Design science is a problem-solving process that has been widely studied [69]. It is an iterative approach [70] that aims to generate generalizable knowledge about design processes

Table 1 Shows which research methods have been selected and employed to address the research questions

RQs	Research methods				
	Design science	Literature study	Expert interview	Document analysis	Case study
RQ_1	✓	✓	✓	✓	
RQ_2	✓	✓	✓	✓	
RQ_3	✓	✓	✓	✓	
RQ_4	✓	✓	✓	✓	
RQ_5	✓	✓	✓	✓	
RQ_6	✓				✓

and decisions [66] by creating and evaluating artifacts that satisfy identified business needs [71, 72]. Similar to a scientific theory, the design process is a set of hypotheses that can be tested by constructing the artifact it describes [62]. However, a scientific theory can only support the feasibility of a design to the extent that the design aligns with the theory's principles. In our work, we utilized domain expert knowledge, the MoSCoW prioritization technique to assess criteria weights and reduce uncertainty, and a six-step decision-making process to apply this framework. Specifically, we addressed the research by constructing a decision model for the BPM language selection problem.

2.3.2 Expert interview

Domain expertise is a fundamental source of knowledge for developing a decision model based on a framework [44]. In qualitative research, *expert interviews* are a critical knowledge acquisition technique [73]. We conducted a series of semi-structured interviews following Meyers' and Newsman's guidelines [74] to explore domain experts' knowledge regarding the BPM language selection problem. To assist in addressing the research questions, we engaged ten academic domain experts from different organizations with extensive experience in the business process modeling domain. It is important to note that these interviews and participants were separate from those involved in the case study research.

To ensure we targeted the appropriate experts, we developed a role description and contacted potential experts via email, including information about the research topic. Experts were selected based on their experience and expertise as reported on their *Google Scholar* and *LinkedIn* profiles, using a set of evaluation criteria that included *expertise, institution, education, publications, citations, H-index*, and *years of experience*.

Each interview followed a semi-structured protocol that lasted between 60 and 90 min. We asked a combination of open- and closed-ended questions to elicit as much information as possible from the experts while minimizing prior bias [75]. The interviews were conducted in person or via a video communication platform such as Google Meet or Zoom, recorded with the interviewees' permission, and transcribed for further analysis. Knowledge acquired during each interview was typically propagated to the following interview to validate captured knowledge incrementally. Additionally, knowledge acquired from the literature review described in Sect. 2.3.3 was propagated to the interviews for validation. For the validity of the results, none of the interviewees or researchers were involved in the case studies. Moreover, the case study participants did not consider domain experts, so we did not interview them to collect data based on their feedback. Accordingly, case study participants did not influence the data collection phases of this study. The interview protocol and full data set are available on Mendeley Data [76].

2.3.3 Literature study

Semi-Systematic Literature Review satisfies several of the criteria of a systematic literature review (SLR) [77], but not all. Specifically, the SLR criteria we have adhered to include the design and documentation of search strings, the use of dedicated databases, the analysis of the number of hits, documenting the included papers for each search string, and the use of explicit inclusion/exclusion criteria. However, we were unable to meet some of the other criteria, such as having multiple authors to evaluate the papers, performing a more rigorous quality assessment of the studies, and conducting data extraction and synthesis. Additionally, we included papers not found via our search strings, which is not typically part of the SLR process. Therefore, we refer to our literature review as a semi-systematic literature review [78] because it accurately reflects the nature of our review.

In contrast to predefined database search strings, we used the snowballing literature review method, as suggested by Wohlin [79], to extract knowledge related to business process modeling and BPM language characteristics, features, and quality attributes. This was because of several reasons: (1) selecting the appropriate databases and constructing effective search strings can be challenging due to the interdisciplinary nature of the studied area [77]; (2) Snowballing has been shown to be comparable to multiple database searches [80]; and (3) Snowballing is a promising approach for identifying relevant work, as new studies almost always cite at least one paper from previous but relevant studies [79].

2.3.4 Document analysis

Document analysis is a knowledge acquisition technique commonly used in qualitative research to investigate and interpret data to derive empirical knowledge [81]. In this study, we used this technique to develop a decision model for the BPM language selection problem and elicit BPM features and quality attributes. We comprehensively reviewed various sources such as web pages, fact sheets, documentation, forums, videos, and product wiki, totaling over 331 unique resources, including grey literature and scientific articles.

The extracted knowledge was then classified into five categories: *BPM languages, BPM features, quality attributes, impacts of BPM features on quality attributes,* and *supportability of BPM features by BPM languages*. This knowledge was employed to build a decision model for the BPM language selection problem in research projects. Finally, the decision model was incorporated into the knowledge base of the DSS.

2.3.5 Case study

Case study research, often referred to as a case study, is an empirical research methodology that investigates a phenomenon within a particular context in the domain of interest [82]. A case study can be employed to test a theory [83], to produce background material for a discussion about a concrete problem in situations where it is hard to find a precise solution [84], to collect data regarding a particular phenomenon, or to apply a tool to evaluate its efficiency and effectiveness [64]. We employed a holistic multiple-case design and followed the guidelines by Yin [82] and Runeson and Holst [85] to conduct and plan the case studies.

Objective This study aims to build a valid decision model for the BPM language selection problem in research projects.

The cases The units of analysis were four BPM language selection problems in research projects that have been published in a journal or conference.

Methods We conducted multiple expert interviews with the case study participants to collect data and identify their preferences and requirements regarding the BPM language selection problem.

Selection strategy We selected the multiple holistic case study method [82] to analyze the data within and across multiple situations.

Theory The proposed decision model is a reference model to support academics with the business process modeling language selection problem in research projects.

Protocol We followed the following protocol to conduct the case studies and evaluate the decision model:

Step 1. Requirements elicitation The case study participants defined their BPM feature requirements and prioritized them according to the MoSCoW prioritization technique [86]. Furthermore, they identified potential solutions and motivated the decision to select a particular BPM language in their research context.

Step 2. Results and recommendations Based on the case studies' requirements and priorities, we defined four separate cases in the knowledge base of the DSS. Then, the DSS recommended a set of ranked feasible BPM languages as alternative solutions for each case individually. Next, we reported the outcomes to the case study participants.

Step 3. Analysis We compared the feasible solutions suggested by the DSS with the BPM language that the case study participants selected for their case. Furthermore, we analyzed the outcomes and observations and reported them to the case study participants. The analysis of the DSS results showed which feature requirements could lead to infeasible solutions (i.e., those not supported by the BPM languages) and which alternative solutions might be used to model the case study projects (see Sect. 5).

3 Decision model

In this study, we utilize an MCDM framework previously introduced and evaluated by our team [87] to build a decision model for BPM language selection. The MCDM framework provides a structured approach to decision-making, which can aid researchers in selecting the most appropriate alternative solution based on their specific project requirements. Including quality attributes and mappings enhances decision-making by allowing for a more comprehensive evaluation of each alternative. The model's components are illustrated in Fig. 1, which also presents the primary building blocks of the DSS.

3.1 BPM languages (RQ_1)

In order to identify a set of BPM languages, we began with a hypothesis and compiled a list of 93 potential BPM languages based on relevant community websites and forums. We then conducted a systematic literature study by reviewing 45 published research articles from well-known sources such as IEEE, Google Scholar, and SpringerLink. In addition, we interviewed ten domain experts to gain more insight into popular and applicable BPM languages. Interestingly, most experts we interviewed were only familiar with a limited number of the BPM languages, as shown in the Domain Alternatives dataset on Mendeley Data [76].

Table 2 Displays a subset of the Boolean features ($Feature^B$), the BPM languages ($Languages$), and the BFL mapping, where BFL: $Feature^B \times Languages \rightarrow \{0, 1\}$. The complete list of all features and mappings is accessible on Mendeley Data [76]. It should be emphasized that identifying these features and their respective mapping to BPM languages was conducted meticulously, reviewing over 168 unique sources of knowledge, such as webpages, manuals, scientific articles, and expert consultations

Boolean Features Business Process Modeling Languages	Coverage	Ordered Checklist	Unordered Checklist	Petri Net	Coloured Petri Net	Flowchart	BPMN 2.0	IDEF0	IDEF3	YAWL	EPC	UML Activity Diagram	UML Use Case Diagram	UML State (Transition) Di	RAD	DFD	VSM (Value Stream Mapp	Gantt Chart	BPEL (2.0)	BPML	EEML	ER/ERD	UML Sequence Diagram	UML Communication Diag
Feature Coverage		26.39%	23.61%	63.89%	63.89%	45.83%	81.94%	40.28%	44.44%	68.06%	55.56%	69.44%	54.17%	52.78%	47.22%	48.61%	36.11%	37.50%	66.67%	48.61%	40.28%	29.17%	52.78%	41.67%
Communication	60.87%	0	0	0	0	1	1	0	1	0	1	1	1	1	1	1	0	1	0	0	1	1	1	1
Analysis	47.83%	0	0	1	1	1	1	1	0	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0
Enaction	43.48%	1	1	1	1	0	1	0	0	1	0	0	0	1	0	1	0	0	1	1	0	0	0	0
Control	39.13%	0	0	1	1	1	1	0	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0
Functional	60.87%	1	1	1	1	1	1	1	0	1	1	1	0	0	0	1	1	1	0	0	1	0	0	0
Dynamic/Behavioural	56.52%	1	0	1	1	0	1	0	1	0	1	1	0	1	1	0	0	0	1	1	0	0	1	1
Informational/Resource	17.39%	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	1	0	0
Organizational	26.09%	0	0	0	0	0	1	0	0	0	1	1	1	0	1	0	0	0	0	0	1	0	0	0
General-purpose ML	69.57%	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	0	0	1	1	1	0	0	0
Textual	26.09%	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	1	0	0
Graphical/Diagrammical	78.26%	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1
Active community	60.87%	0	0	1	1	1	1	0	0	1	0	1	1	1	0	0	1	1	1	0	0	1	1	1
Organizational body	34.78%	0	0	0	0	0	1	0	0	1	0	1	1	1	0	0	0	0	1	0	0	0	1	1
Conformity to standard	43.48%	0	0	1	1	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1	1
Evolutionary	34.78%	0	0	0	0	0	1	0	0	1	0	1	1	1	0	0	0	0	1	0	0	0	1	1
Control flow/Sequence	65.22%	1	0	1	1	1	1	0	1	1	1	1	0	0	1	0	0	1	1	1	1	0	1	0
Information flow/Message flow	65.22%	0	0	1	1	0	1	1	0	1	0	1	1	0	1	1	1	0	1	1	1	0	1	1
Data flow	56.52%	0	0	1	1	0	1	1	1	1	1	0	1	1	1	1	0	0	1	0	0	0	1	0
Inclusive choice (OR)	26.09%	0	0	0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0
Exclusive choice (XOR)	65.22%	0	0	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0	1	1	1	0	1	0
Parallelism (AND)	56.52%	0	0	1	1	0	1	0	1	1	1	1	0	1	1	0	0	0	1	1	1	0	1	0
Input (resource)	47.83%	0	0	1	1	1	1	1	0	0	1	1	0	0	0	1	0	0	1	1	1	0	0	0
Output (deliverable)	47.83%	0	0	1	1	1	1	1	0	0	1	1	0	0	0	1	0	0	1	1	1	0	0	0
External entities	34.78%	0	0	0	0	0	1	0	0	0	1	1	1	0	0	1	1	0	1	0	1	0	0	0
Agent (Roles/Entitites)	60.87%	0	0	0	0	0	1	0	0	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1
Duration	26.09%	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0
Deadline	17.39%	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0
Business rules	21.74%	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0
Business goals	8.70%	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
Annotation	43.48%	1	1	0	0	1	1	0	0	0	0	1	1	0	0	0	1	0	1	1	0	0	1	0
Task/Activity	78.26%	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	1	1	1	1	0	1	0
Event/Trigger	47.83%	0	0	1	1	0	1	0	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0
Mappability	26.09%	0	0	1	0	0	1	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0
Standardized file format	47.83%	0	0	1	1	0	1	0	0	1	0	1	1	1	0	0	0	0	1	1	0	0	1	1
Execution Engine Availability	39.13%	0	0	1	1	1	1	0	1	1	1	0	0	1	0	0	0	0	1	0	0	0	0	0
Simulation support	30.43%	0	0	1	1	0	1	0	0	1	0	0	0	0	1	0	0	0	1	1	0	0	0	0

To prevent potential biases, we only included BPM languages mentioned in at least four sources of knowledge, including published research articles and expert interviews. After analyzing the collected data, we identified 23 BPM languages mentioned in at least four resources. Due to time and resource constraints, this study focused only on imperative BPM languages and excluded languages such as DECLARE [88] and the Case Model Management and Notation (CMMN) [89]. A subset of the BPM languages considered in the decision model is presented in the first row of Table 2. The complete set of languages can be found on Mendeley Data [76].

3.2 BPM features (RQ_2)

Identifying appropriate BPM features primarily relied on domain experts' feedback and an extensive literature review through a snowballing process. An initial hypothesis regard-

Table 3 Displays the mapping among non-Boolean BPM features ($Features^N$) and BPM languages (*Languages*), so that *NFL*: $Features^N \times Languages \rightarrow$ H, M, L, N. The mapping is based on a 4-point Likert scale (**H**igh, **M**edium, **L**ow, **N**one). Note that this table is primarily the result of expert interviews and a review of the sources of knowledge identified in the *source of knowledge* column

non-Boolean Features Business Process Modeling Languages	Ordered Checklist	Unordered Checklist	Petri Net	Coloured Petri Net	Flowchart	BPMN 2.0	IDEF0	IDEF3	YAWL	EPC	UML-AD	UML UCD	UML STD	RAD	DFD	VSM	Gantt Chart	BPEL (2.0)	BPML	EEML	ER/ERD	UML-ID: Sequence	UML-ID: Collaboration	Source of knowledge
Graphical Economy	N	N	L	L	M	H	L	M	H	M	H	L	H	M	L	H	M	N	N	H	N	H	L	*Document analysis*
Formality	L	L	H	H	L	H	H	H	H	L	M	M	M	H	L	L	L	H	H	H	H	L	M	*Literature study*
Perceptual discriminatibility	N	N	H	H	H	M	H	M	H	H	M	H	H	L	H	H	L	N	N	L	N	L	H	*Literature study*
Semantic transparency	N	N	H	H	H	M	H	L	H	H	H	H	H	L	H	H	H	N	N	M	N	M	L	*Literature study*
HR availability	N	N	L	L	H	H	M	L	L	L	H	M	M	L	L	L	M	M	M	N	L	M	M	*https://linkedin.com*
Tool availability	H	H	H	H	H	H	L	M	M	H	H	H	H	L	H	H	H	H	L	L	H	H	H	*https://google.com*
Popularity	N	N	H	M	H	H	L	L	L	M	H	M	M	H	M	L	N	M	L	N	L	M	M	*Domain experts*
# Google hits	N/A	N/A	45100000	424000	4,99E+08	2910000	7440	10300	23300	721000	1300000	8120000	6780000	61800000	929000	609000	14400000	270000	33700	316000	940000	1800000	1800000	*https://google.com*
# Scholar hits	N/A	N/A	7,00E+05	17300	4,00E+05	60500	1820	1730	3540	17300	17900	17400	18000	87300	17400	16000	24200	17600	3540	8350	12300	19200	19200	*https://scholar.google.com*
# LinkedIn jobs, global	N/A	N/A	19	0	956	7740	13	0	2	431	2031	3070	1	609	639	31870	588	840	233	0	133	36	36	*https://linkedin.com*
Domain experts	9	9	10	2	10	10	0	3	8	9	9	1	1	2	7	4	1	2	0	1	0	1	1	*Domain experts*
Literature study	0	0	23	3	16	28	20	15	13	18	36	3	2	14	13	1	3	9	8	4	8	1	1	*Literature study*

ing the BPM feature set was developed using sources such as documentation analysis, web pages, manuals, and forums. Each BPM feature possesses a data type, which can be classified as either *Boolean* or *non-Boolean*. For instance, features like "graphical" and "formality" are categorized as Boolean and non-Boolean data types.

During the research phase, ten domain experts participated in the evaluation of the initial set of BPM features. A potential list of BPM features was developed by identifying new concepts discovered during interviews. To minimize potential biases, we only included BPM features mentioned in at least four sources of knowledge, including published research articles and domain expert interviews. As a result, 72 BPM features were identified, including 64 Boolean features (Table 2) and eight non-Boolean features (Table 3), which were extracted from the outcomes of the literature review and expert interviews.[1]

3.3 Quality attributes (RQ_3)

Quality attributes are inherent non-functional characteristics [45]. The quality of a BPM language, a primary concern for academics when selecting one, is determined by the degree to which it meets the decision maker's requirements, such as expressiveness, understandability, and ease of use. Identifying quality attributes is recommended by other researchers to measure the characteristics of BPM languages.

The results of our literature study, available on Mendeley Data [76], confirm that researchers do not agree on standard criteria or quality attributes for evaluating BPM languages. Furthermore, some suggested quality characteristics were mainly applied to address specific research questions, and different appellations referred to the same concepts and vice versa. Therefore, there is a need for generic, domain-independent, and well-defined criteria to assess BPM languages.

Similar decision models for other MCDM problems in the software engineering domain (e.g., [45, 64, 66]) use the ISO/IEC 25010 [90] and extended ISO/IEC 9126 [91] standards as the foundation of the quality evaluation model. These standards provide consistent terminology for specifying, measuring, and evaluating systems and software [92]. However, most quality criteria defined in these ISO standards do not apply to BPM languages. Therefore, we conducted a concise literature review to gather knowledge about frequently used quality attributes for assessing BPM languages. We merged characteristics with overlapping or largely overlapping definitions and only considered quality characteristics mentioned in at least two sources of knowledge to prevent potential biases. These quality attributes provide a standardized way of measuring a BPM language and form the foundation of the quality evaluation model that determines which quality characteristics should be considered when assessing the properties of a BPM language.

[1] The entire list of BPM features and their mappings against BPM languages can be found on the *BPM language Selection* website (https://dss-mcdm.com).

3.4 Mapping: features and qualities (RQ_4)

The mapping between the sets of *BPM language quality attributes* and *BPM features* has been established with the help of domain experts' knowledge. In this phase of the research, three domain experts participated in mapping the BPM features (*Features*) to the BPM languages' quality attributes (*Qualities*) based on a Boolean adjacency matrix. The final Boolean adjacency matrix can be accessed on Mendeley Data [76] and is publicly available for other researchers to use. The framework allows for a feature to be present in multiple quality attributes. For example, the quality attributes *portability* and *replaceability* are influenced by the *standardized file format* BPM feature.

Granularity refers to the level of detail that a BPM language supports for describing business processes [35]. The domain experts noted that the BPM language positively impacts the granularity of BPM features *graphical economy* and *decomposition*.

Enactability measures the degree to which a BPM language supports automated enaction [27] and dynamic simulation of business processes [19]. It includes the BPM features *true concurrency*, *simulation support*, and *formality*, among others.

Completeness refers to the degree to which all the necessary concepts of the application domain are represented in modeling [21]. In other words, a BPM language is considered *complete* when it can cover all the relevant modeling perspectives required by decision makers [54] and has the appropriate modeling concepts to express processes. This characteristic includes features such as *functional*, *behavioral*, *informational*, and *organizational* modeling language constructs.

Stability refers to how stable and well-accepted a BPM language is in the BPM language community [35]. Factors such as *tool availability*, *conformity to standards*, *active community*, and *popularity* can impact the perceived stability of a BPM language.

Learnability is the ease with which a BPM language can be learned [28, 52]. A BPM language is considered *learnable* when modelers and actors require little effort to master the notation [52]. This characteristic can be supported by BPM features such as *graphical notation*, *perceptual discriminability*, and *semantic transparency*.

Flexibility or **Modularity** refers to the ability of a BPM language to design models that can be easily adapted in response to business process changes [50]. This characteristic includes BPM features such as *graphical economy* and *general-purpose modeling language*.

Innovation Inducer describes the ability of a BPM language to inspire modelers to discover new solutions and modeling practices [31]. The quality attribute is positively influenced by an *active community* and the *availability of documentation*.

Tool Support or *Software Support* refers to the degree of the presence of software that supports a particular process modeling language [35, 50]. This characteristic includes features such as a formal specification for converting process models to an executable BPM language (*mappability*), *standardized file format*, *execution engine availability*, and the availability of *free* and *open-source* tools.

The knowledge acquired on the mapping between BPM features and quality attributes was utilized to determine the impact factors incorporated in the DSS score calculation [44, 45].

3.5 Mapping: features and languages (RQ_5)

A BPM language can have Boolean ($Feature^B$) or non-Boolean ($Feature^N$) features. Boolean features support functional modeling perspectives, while non-Boolean features assign non-Boolean values to BPM languages. For example, the popularity of a BPM language in the market and academic domain can be low, medium, or high. Therefore, this study considers both Boolean and non-Boolean features, such that $Features = Feature^B \cup Feature^N$.

The mapping *BFL*: $Feature^B \times Languages \rightarrow 0{,}1$ defines the supportability of Boolean BPM features by BPM languages. $BFL(f,l) = 0$ indicates that the BPM language does not support the feature or there is no evidence or proof of the feature's supportability. $BFL(f,l) = 1$ means that the BPM language l supports the BPM feature f. The BPM feature mapping is based on scientific publications, grey literature, documentation of BPM languages, and domain expert interviews. Table 2 shows a subset of the Boolean features considered in the decision model.

Based on the knowledge acquired from the systematic literature review and domain expert interviews, we have defined eight non-Boolean BPM features, such as formality, graphical economy, maturity, and popularity in the market. Non-Boolean BPM features have been assigned values based on a 4-point Likert scale (**N**one, **L**ow, **M**edium, **H**igh), where *NFL*: $Feature^N \times Languages \rightarrow N, L, M, H$, based on predefined parameters. For instance, the popularity of a BPM language was defined based on *domain expert opinion*, *LinkedIn (Jobs, Global)*, *the number of SLR occurrences*, *Google search hits*, and *Google Scholar hits*. Table 3 presents a subset of the non-Boolean BPM languages and their assigned values in the decision model.

3.6 Feature requirements

The DSS receives BPM feature requirements based on the MoSCoW prioritization technique, which is a prioritization method used by decision makers to weight their feature requirements. The weights for the MoSCoW prioritization technique are defined as $W_{MoSCoW} = \omega_{Must}, \omega_{Should}, \omega_{Could}, \omega_{Will\ not}$, according to the definition by Clegg et al. [93]. BPM feature requirements with *Must-Have* or *Won't-Have* priorities act as hard constraints, while BPM feature requirements with *Should-Have* and *Could-Have* priorities act as soft constraints. In other words, the DSS excludes all infeasible BPM languages that do not support features with *Must-Have* priorities and support features with *Won't-Have* priorities. The system then assigns nonnegative scores to the feasible BPM languages based on the number of BPM features with *Could-* and *Should-Have* priorities.

For non-Boolean BPM features, decision makers specify their desirable values from their perspective. For instance, a decision maker may be interested in prioritizing BPM languages with a popularity level above the average and a high level of formality. As such, the levels of *popularity* and *formality* are considered *Should-Have* and *Must-Have* features, respectively.

4 Empirical evidence

Four academic case studies in the academic domain have been conducted to answer the last research question (RQ_6) and evaluate and signify the decision model's usefulness and effectiveness in addressing the BPM language selection problem in research projects. To increase diversity in our evaluation, we have selected the research articles that acted as the case study subjects from four different conferences, including the International Conference on Wirtschaftsinformatik (WI), the International Conference on Software Architecture (ICSA), the International Conference in Business Process Management, and International Conference on Advanced Information Systems Engineering (CAiSE), with a CORE Rank[2] of at least A*, A, B, or C. Moreover, the authors of the selected case studies were located in three different countries, Australia, Germany, and Spain (See Table 5).

Before participating in this research, the case study participants identified and selected a BPM language for the modeling problem in their respective contexts. The selected BPM language was their shortlist of ranked feasible solutions (Table 5). Furthermore, the case study participants explained their BPM language selection decisions. Then, the case study participants specified their BPM feature requirements based on the MoSCoW prioritization technique (Table 4), defined and stored as four academic cases in the knowledge base of the DSS. Finally, the inference engine of the DSS-generated feasible solutions for each case is observed. The remainder of this section describes the ranked shortlists and prioritized feature requirements and analyses the DSS outcomes.

4.1 Case study 1: a decision model for choosing patterns in blockchain-based applications

Based on use case characteristics and implicit pattern trade-offs, Xu et al. [94] proposed multiple decision models to assist architects and developers in selecting a suitable (combination of) pattern(s) for blockchain-based applications. The selection has been based on characteristics of the use cases and trade-offs implicit in the pattern. The decision models were then evaluated for correctness and usefulness using six domain expert interviews. The results show that the proposed design model brings structure and rationale to the decision process. The decision models proposed in the study have been modeled using a notation that borrows its elements from the Business Process Model and Notation. Note that the authors introduced various other elements to allow for representing high-level design goals (grey box), complements-relationships (double-headed arrow), and pattern constraints (octagon). According to the case study participant, the principal reason for selecting BPMN, as opposed to, for example, UML-AD, was its high expressiveness—having more graphical constructs and more straightforward ways to model various gateways.

4.1.1 Requirements

The case study participant defined the following subset of requirements for modeling the decision models for pattern selection of blockchain-based applications (for more detail, see Table 4):

- The BPM language must cover the functional and behavioral perspectives to support views of what, when, and how activities are performed (R05 and R06).
- The case study participant indicated that the BPM language had to facilitate understanding, communicating, and describing a process graphically (R01, R10, and R11).
- A high expressiveness adds significant value to the BPM language (R53). The language should have explicit gateways for inclusive choice, the only choice, and parallelism (R19, R20, and R21). Furthermore, the case study participants looked for languages depicting events, triggers, input, and output (R32, R22, and R23).
- The maturity of the BPM language was a quality concern of the expert, so active community, evolutionary, organizational body, and documentation availability were prioritized as Should-Have (R58, R62, R60, and R64).
- Potentially suitable BPM languages must have high tool availability (R70). Moreover, the availability of free tools would add significant value (R71).
- According to the case study participant, the language should be easy to understand. The BPM language's per-

[2] https://www.core.edu.au/.

Table 4 Shows the feature requirements of the case study participants based on the MoSCoW prioritization technique, where Must-Have (M), Should-Have (S), Could-Have (C), and Won't-Have (W) represent different priority levels. The percentage of BPM languages that support each feature is denoted by the value next to the feature (Cov.). It should be noted that the full list of feature requirements is also available on Mendeley Data [76]

	Cov.	Case study 1	Case study 2	Case study 3	Case study 4	
Communication	61%	M	S	C	M	R01
Analysis	48%	M	C		C	R02
Enaction	43%		M	M > S	S	R03
Control	39%		M	S	S	R04
Functional	61%	M	M	M	M	R05
Dynamic/Behavioural	57%	M	M	M	M	R06
Informational/Resource	17%		S		S	R07
Organizational	26%		M	S	C	R08
General-purpose ML	70%	M	M	C	M	R09
Textual	26%	W	W			R10
Graphical/Diagrammical	74%	M	M	C	M	R11
Tabular	4%		W			R12
Mathematical	22%		S	S		R13
Imperative	100%	S	M	M		R14
Declarative	0%		W	W		R15
Control flow/Sequence	65%	M	M	M	S	R16
Information flow/Message flow	65%		M	C	C	R17
Data flow	57%		S	M > S		R18
Inclusive choice (OR)	26%	M	M		C	R19
Exclusive choice (XOR)	65%	M	M	M	M	R20
Parallelism (AND)	57%	M	M		M	R21
Input (resource)	48%	M	S	C	S	R22
Output (deliverable)	48%	M	C			R23
External entities	35%	C	M	M		R24
Agent (Roles/Entitites)	61%	S	M	S		R25
Duration	26%		C			R26
Deadline	17%		C			R27
Business rules	22%	S	S		C	R28
Business goals	9%	S	S		M > S	R29
Annotation	43%	M > S	C		C	R30
Task/Activity	78%	M	M	M	M	R31
Event/Trigger	48%	M	M	M	S	R32
Mappability	26%	C		S		R33
Standardized file format	48%	C	M	M		R34
PNML	9%					R35
XSD	13%		C	M > S		R36
XMI	26%		C	M > S		R37
yawl-XML	4%					R38
EPML	4%					R39
XML	52%			M		R40
Execution Engine Availability	52%		M			R41
Simulation support	30%		C			R42
Verification support	39%		C			R43
True concurrency	26%	S		W		R44
System Interpretation	26%	C	M	M > S		R45
Decomposition	39%	S	C		M > S	R46
High-level support	43%	S			S	R47
Low-level support	83%	S				R48
State-based	52%		M > S			R49
Activity-based	61%	M	C	M	S	R50
Novice	48%	S	S		C	R51
Expert	65%	M > S	M	M	S	R52
Graphical economy	78%	M				R53
Visual emphasis: colour	13%	W > N				R54
Formality	100%	M > S	M	M	S	R55
Perceptual discriminatibility	78%	S	S		C	R56
Semantic transparency	78%	S	S			R57
Active community	61%	M > S	S	M		R58
HR availability	87%	C		C		R59
Organizational body	35%	M > S		C		R60
Conformity to standard	43%		S			R61
Evolutionary	35%	M > S		S		R62
Theoretical foundation	83%	M			C	R63
Documentation availability	57%	M > S	M	M	M	R64
Training material availability	96%	S		C		R65
YouTube availability	83%	C				R66
Course availability	65%	C				R67
Blog/Article availability	91%	S				R68
Popularity	83%	M	M	S		R69
tool availability	100%	M	M	S	C	R70
free tools availability	83%	S	M	M		R71
open-source tools availability	74%	C	M	M		R72

ceptual discriminability (R56) and semantic transparency (R57) had to be above average.

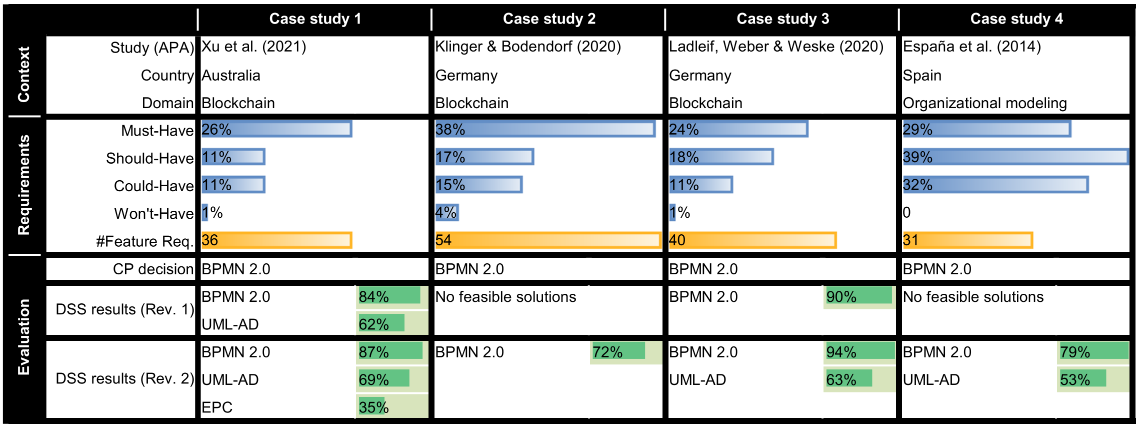
- The BPM language must have a high level of popularity and be covered in the academic domain to have a theoretical foundation (R69 and R63).
- As the models had to be understandable by humans, enactability was of lesser importance to the case study participant (R41, R42, and R43), although the BPM language should allow for true concurrency (R44). Additionally, a standardized file format (R34) and a formal specification of mappability to a business process execution language (R33) would be nice to have.

4.1.2 Results

The case study participant mentioned BPMN and UML-AD as potentially feasible solutions, ultimately selecting BPMN due to its high expressiveness. The expert defined 36 BPM feature requirements and assigned Must-Have priorities to over half of them (53%). Furthermore, 3% of the features have a Won't-Have priority assigned to them. The remaining percentage of feature requirements were prioritized as soft constraints (Should-Have and Could-Have).

The DSS results were similar to the case study participants, suggesting BPMN as the "best-fit" feasible solution. Additionally, albeit with lower scores, the DSS suggests

Table 5 Presents the context of the case study research papers (*Context*), the feature requirements (*Requirements*), and the outcomes of the DSS for the case studies based on the requirements and priorities (*DSS Solutions*). The table also includes the number of feature requirements (*#Feature Req.*), coverage (*Coverage*), and the percentages of the Must have, Should have, Could have, and Won't-Have (MoSCoW) priorities. The percentage next to the solutions signifies the scores calculated by the DSS. Please note that the DSS results have been revised two times (Rev. 1 and Rev. 2) accordingly to the revision of priorities of requirements



		Case study 1	Case study 2	Case study 3	Case study 4
Context	Study (APA)	Xu et al. (2021)	Klinger & Bodendorf (2020)	Ladleif, Weber & Weske (2020)	España et al. (2014)
	Country	Australia	Germany	Germany	Spain
	Domain	Blockchain	Blockchain	Blockchain	Organizational modeling
Requirements	Must-Have	26%	38%	24%	29%
	Should-Have	11%	17%	18%	39%
	Could-Have	11%	15%	11%	32%
	Won't-Have	1%	4%	1%	0
	#Feature Req.	36	54	40	31
Evaluation	CP decision	BPMN 2.0	BPMN 2.0	BPMN 2.0	BPMN 2.0
	DSS results (Rev. 1)	BPMN 2.0 84% UML-AD 62%	No feasible solutions	BPMN 2.0 90%	No feasible solutions
	DSS results (Rev. 2)	BPMN 2.0 87% UML-AD 69% EPC 35%	BPMN 2.0 72%	BPMN 2.0 94% UML-AD 63%	BPMN 2.0 79% UML-AD 53%

UML-AD (69%) and EPC (35%) as feasible solutions. A significantly lower score for the Event-driven Process Chains is expected, as the features impacting the language's maturity and universality quality attributes have lower supportability than BPMN 2.0 and UML-AD. For example, EPC is not (actively) maintained by an organizational body (R60), whereas BPMN and UML are.

4.2 Case study 2: blockchain-based cross-organizational execution framework for dynamic integration of process collaborations

Klinger and Bodendorf [95] proposed an Ethereum-based framework for executing cross-organizational process collaborations, solving integration and trust problems frequently occurring with many participants collaborating on a commonly defined workflow. The framework is exemplified and evaluated using a use case from a large German industrial manufacturing company. The results indicate that the framework provides process transparency to all process participants, *having the potential to reduce costly lay times and quality improvements of strictly enforced workflows*. Contrary to former blockchain-based execution frameworks, which focus on orchestration diagrams as the basis for execution, Klinger and Bodendorf use a transpiler to automatically generate Smart Contracts, resembling the process flow based on BPMN collaboration diagrams. As a result, additional transformation steps between collaboration and single pool (process orchestration) models become obsolete.

4.2.1 Requirements

The case study participant defined the following subset of requirements, reasoning their decision to use BPMN collaboration diagrams to represent the process models (for more detail, see Table 4):

- The BPM language must support the functional, behavioral, and organizational perspectives (R05, R06, and R08).
- As the primary objective of the proposed framework is cross-organizational blockchain-based process execution, and the BPM language must facilitate the execution, managing, and monitoring of a business process and its models (R03 and R04).
- The case study participant highlighted that the BPM language's serialization format must be graphical (R11). Textual and tabular (R10 and R12) notations must be excluded as feasible solutions.
- As aforementioned, the enactability of the process models was a decisive factor for the BPM language selection process of the case study participant. System interpretation and execution engine availability received Must-Have priorities (R41 and R45).
- The expert indicated that the BPM language must have a standardized file format (R34).
- The maturity of the BPM language was not relevant to the case study participant as long as an extensive and well-documented manual was available (R64).
- The case study participants preferred to employ a BPM language with an above-average tool availability (R70). Moreover, free and open-source tool availability are considered hard constraints (R71 and R72).
- The BPM language must allow for modeling control flow and informational flow between internal and external agents (R16, R17, R24, and R25).
- According to the case study participant, *the Ethereum Smart Contracts have been misused as a "State Machine",*

prompting the experts to have the activity-based BPMN adopted to "states". Correspondingly, the BPM language must be activity-based (R50) and include a mechanism to express states in the sense of transaction boundaries (R49).

4.2.2 Results

Before participating in this research, the case study participant selected BPMN as the feasible solution. Next, he identified 54 feature requirements based on the MoSCoW prioritization technique. Due to the complex requirements, more than half of the features prioritized hard constraints with Must-Have and Won't-Have values. The remaining set of feature requirements received Should-Have and Could-Have priorities.

Accordingly, the DSS excluded practically all BPM languages as feasible solutions in the decision model, slightly relaxing the constraints opened up BPMN as a feasible solution. More specifically, we convert the Must-Have requirement state-based (R49) from a hard constraint to a soft constraint (Should-Have), as the specification of mechanisms to access states is outside of the scope of the BPMN standard [96] according to our mappings. Instead, we considered this an implementation detail for process engines, such as Camunda, in which BPMN-Transaction Boundaries can prescribe states. Subsequently, the decision model considers BPMN to be an activity-based BPM language [97–99] rather than a state-based language. To prevent the DSS from excluding BPMN as a feasible solution, we have prioritized state-based as a Should-Have feature requirement in the case definition of the DSS.

Similar to the considerations of the case study participant, the DSS suggests BPMN as the best-fit decision alternative. Combining two or more process modeling languages could also help achieve the combined goal of communicating and enacting a single process model. However, using two or more BPM languages and process models (re)introduces the redundant transformation step the framework tries to avoid. As such, we did not further relax the requirements.

4.3 Case study 3: external data monitoring using oracles in blockchain-based process execution

The conceptual limitations of Blockchain restrict the real-time integration of external data, which could lead to runtime behavior that is non-compliant due to missing data updates or incorrectly evaluated conditional constraints. As such, Ladleif, Weber, and Weske [100] contribute a first assessment of the nature of these issues and develop compliant transaction-driven semantics facilitating common business process patterns within smart contracts. Finally, they extend and propose various Oracle-based implementation strategies on a conceptual, technology-independent level. The work contributes toward holistic support for business processes on Blockchain.

As a running example throughout the work, a ticketing system of a railway company has been modeled using the Business Process Model and Notation. According to the case study participant, the predominant motivation for selecting BPMN collaboration diagrams was his popularity and familiarity with the notation. Moreover, the expert highlighted that BPMN choreography diagrams were excluded from the ranked shortlist of feasible solutions as these diagrams lack objects for modeling data stores and data flows.

4.3.1 Requirements

The case study participant defined the following subset of requirements for modeling the ticketing system of a railway company as the running example in their study (for more detail, see Table 4):

- The BPM language must cover the functional and behavioral perspectives (R05 and R06). Additionally, the ability to depict where and by whom activities are performed would add remarkable value to the BPM language (R08).
- Due to the nature of the work revolving around process execution, the BPM language must support the goal of enaction (R03). If the BPM language were to facilitate control, that would also add significant value (R04).
- As the authors intended to implement a subset of the semantics of the notation to fit the research questions, the availability of an execution engine was not relevant to the decision-making process (R41). For the same reasons, it was not relevant whether the BPM language allowed for the simulation and verification of the process models (R42 and R43).
- The case study participant indicated that the article was written specifically for domain experts. The easy understandability of the BPM language played a trivial role in the decision-making process (R51 and R52).
- According to the case study participant, the availability of free and open-source tools was mandatory to select a suitable BPM language (R71 and R72). Furthermore, an above-average tool available for free and paid solutions was also desirable (R70).
- The market's popularity was the expert's main quality concern. They want to select potential BPM languages with an active community and high popularity (R58 and R69).
- The case study participant required the BPM language to have a high degree of formality with well-defined semantics (R55).

- Due to the nature of blockchain-based process execution, where *in the simplest form, every activity or element would be one blockchain transaction serialized and executed one after the other*, truly concurrent BPM languages had to be excluded (R44). Moreover, the expert preferred imperative BPM languages due to the finite possible process variations (R14 and R15).
- Finally, the case study participant stated that the BPM language must have a standardized file format (R34). The high degree of tool support for the extensible markup language (XML), e.g., in Camunda [Men14], the file format had to be of this type (R40) specifically.

4.3.2 Results

Before participating in this study, the case study participant selected BPMN as the best-fit feasible solution for the modeling task. The expert defined 40 BPM feature requirements and assigned Must-Have priorities to 17. Another 13 have been prioritized as Should-Have, eight as Could-Have, and two as Won't-Have.

Based on the requirements defined by the case study participant, the DSS ranks BPMN as the best-fit feasible solution with 90%. We slightly relaxed the file-format priorities for XSD and XMI from Must-Have to Should-Have, opening up UML-AD as a potentially feasible solution with a score of 63%. The significantly lower score for UML-AD makes sense since the notation supports fewer features related to the experts' goal of process execution, such as system interpretation and a graphical construct for representing the flow of data (R03, R45, and R18). Hence, the results of the DSS are similar to the ranked shortlist of feasible solutions of the case study participant, asserting that BPMN is the best-fit feasible solution for modeling the ticketing system of a railway company as the running example in the study.

4.4 Case study 4: capability-driven development of an SOA platform: a case study

Capability-driven development (CDD) presents a paradigm for organizational modeling and information technology development within enterprises. Espana et al. [101] identified several critical pillars underlying this approach, including capability modeling that encompasses goals, context, and processes, pattern-based design, and a heightened awareness of runtime context, which enables service delivery adjustment. Nevertheless, the authors have observed a dearth of empirical studies on the practical application of CDD in industry. To address this gap, Espana et al. [101] conducted a research study and reported their findings in their paper, which focused on capability modeling in a service-oriented architecture (SOA) development project.

The case study participant (the first author of the paper [101]) explained that they selected BPMN as their modeling language to represent the process model, process variants, and capability delivery patterns of the SOA platform. To model aspects of the capability solution that traditional business process modeling notations did not cover, the authors extended the BPMN notation slightly. Although BPMN met their needs, they acknowledged that identifying the most suitable notation for each model view remains an open research challenge. The author also emphasized the need for specialized guidelines or extensions for different notations and situational guidelines to adapt to project contingencies. In CDD, identifying and modeling variability is crucial to automating service delivery without manually customizing software code. While BPMN was a suitable modeling language for their study, the authors recognized its limitations.

4.4.1 Requirements

The case study participant defined the following subset of requirements for selecting a modeling language for the capability delivery patterns of the SOA platform (more information can be found in Table 4):

- The language for modeling business processes should clearly and concisely represent the process that all stakeholders can easily understand (R01).
- The language needs to facilitate the analysis and improvement of a process. This means that it should enable the identification of inefficiencies and bottlenecks in the process and provide a means of optimizing it (R02).
- The language should enable the translation of the process into an executable form, allowing for automated execution (R03).
- The language should enable tracking of the process as it progresses, allowing for real-time monitoring and management (R04).
- The language selected should provide the ability to define process elements and their functional dependencies, including activities, subprocesses, and other relevant components. This will enable a comprehensive understanding of the process and its constituent parts (R05).
- The language should specify when and how activities are performed within the process. This will enable a clear and concise representation of the sequence of activities that comprise the process (R06).
- The language should enable modeling data entities produced or manipulated by the process and their relationships. This will provide insight into the structure of informational entities within the approach (R07).
- The language should also enable the identification of where and by whom activities are performed, both functionally and physically. This will provide insight into the

role of individuals and teams in the process and the physical location of activities (R08).
- The language should be designed to model the widest variety of application domains, so it should be flexible and adaptable enough to accommodate a broad range of business processes (R09).
- The language notation should be graphical. This will enable a visual representation of the process, making it easier to understand and communicate (R11).
- The language should allow for modeling an order in which tasks are to be executed, including a concept of time. This will enable a clear and concise representation of the sequence of activities within the process (R16).
- The language should enable the modeling of messages and their flow. This will provide insight into the flow of communication within the process (R17).
- The language should allow for modeling activity/task-based models. This will represent the process in terms of discrete activities and tasks (R31).
- The language should allow for decomposition and process nesting. This will enable a hierarchical representation of the process, providing insight into the relationship between activities and subprocesses (R46).
- A comprehensive and well-documented manual or language specification should be available for the selected language. This will guide the use of the language and ensure consistency and accuracy in its application (R64).

4.4.2 Results

The experts involved in the case study opted for BPMN as the most suitable and practical means of modeling language for depicting the process model, process variations, and capability delivery patterns of the SOA platform. The experts in question delineated 31 essential BPM features and designated 9 of these as Must-Haves. Additionally, 12 features were accorded Should-Have priority, whereas 10 were classified as Could-Haves.

Based on the requirements specified by the case study participants, the DSS initially failed to produce any viable results. Consequently, we informed the case study participant of this outcome and adjusted two features of the DSS, namely Business goals and decomposition (R29 and R46). After these adjustments, the DSS recommended BPMN as the most appropriate and feasible solution with a ranking of 79%, followed by UML-AD as a secondary alternative solution with a ranking of 53%. The lower score for UML-AD can be attributed to the notation supporting fewer features pertinent to the experts' objective of process execution, such as Business Rules, Enaction, and Decomposition (R03, R28, R46). Thus, the DSS-generated outcomes aligned with the shortlist of feasible solutions ranked by the case study participants, affirming that BPMN is the best-fitting feasible solution for capability modeling in the SOA development project.

5 Analysis of the results

The degree to which the goal dimension of the artifact is met is determined based on four evaluation metrics: (1) *efficacy*, which refers to the extent to which the artifact achieves its intended effect [102], (2) *validity*, which is the degree to which the artifact operates correctly [103], and (3) *generality*, which is the broader goal addressed by the artifact [104]. In addition, we incorporated the opinions of domain experts and compared the results of the DSS to the BPM language selection decision of case study participants (ranked shortlist). Since, to the best of our knowledge, there are no other accessible decision support systems in the business process modeling domain that allow for case definition, the evaluation is considered to be absolute.

Regarding efficacy, the decision model aims to assist academics with their business process modeling language selection problems in research projects. The domain experts and case study participants stated that the comprehensive overview of BPM languages and features could be useful for exploring implicit feature requirements and discovering alternative solutions for modeling problems in the academic domain. Furthermore, they indicated that decision makers often choose a BPM language based on familiarity rather than the characteristics and quality attributes relevant to the modeling problem. Hence, several domain experts expressed that modelers and scholars with limited knowledge of BPM languages could find the decision model useful for evaluating and justifying their selection decisions. Finally, the case study participants appreciated the validation of the BPM language selection decision provided by the artifact. In conclusion, the decision model appears to perform satisfactorily based on the efficacy metric and could be a valuable tool for academics.

The decision support system recommends *BPMN 2.0* as a feasible solution for all four case studies (see Table 5 for details), indicating that the Business Process Model and Notation 2.0 supports all of the case study participants' features with a Must-Have priority. These results align with previous research showing BPMN 2.0 to be among the most popular process modeling solutions on the market [105, 106], and the de-facto standard for process modeling [107, 108]. Furthermore, BPMN's maturity, universality, expressiveness, and popularity are relatively high, as it supports 82% of the BPM features considered in the decision model (for more detail, see [76]). Similar findings have been reported by Silva [109], and Entringer et al. [17], where BPMN is the only BPM language that meets all evaluative criteria.

Regarding the participant feature requirements, Table 4 shows that the *functional perspective*, *behavioral perspec-*

tive, *Exclusive choice (XOR)*, and *Task/Activity* features were identified as "Must-Have" features by all participants. The selection of these features indicates that participants prioritize the BPM language's ability to express the functional and behavioral aspects of the business process and specify exclusive choices and activities/tasks that form part of the process. Organizations can utilize this information to evaluate BPM languages and select one that best suits their needs for expressing their business processes' functional and behavioral aspects.

The *general-purpose* feature, the *graphical/diagrammatical* representation, the *control flow/sequence*, *Parallelism (AND)*, and *Event/Trigger* were identified as essential features by at least three of the four case study participants. These features are crucial for the design and execution of complex business processes. The *general-purpose* feature enables the BPM language to be used in various contexts, making it a versatile tool for modeling business processes. The *graphical/diagrammatical* representation facilitates the visualization and understanding of the business process, which is essential for communication and collaboration among stakeholders. The *control flow/sequence* and *Parallelism (AND)* features are necessary for the proper execution of business processes, ensuring that activities are performed in the correct order and that parallel activities are executed correctly. Finally, the *Event/Trigger* feature allows for the specification of events that initiate activities in the business process, enabling organizations to automate processes triggered by specific events.

All case study participants identified *tool availability*, *Documentation availability*, *Formality*, *Experts*, *Activity-based*, *Input (source)*, and *Communication* as feature requirements. These features are critical for the successful adoption and use of BPM languages. *Tool availability* refers to the availability of software tools that support the BPM language, which is important for developing, testing, and executing business processes. *Documentation availability* is crucial for facilitating stakeholders' understanding and use of the BPM language. The *Formality* of the BPM language syntax and semantics is important for ensuring that the language is unambiguous and consistent.

Including the *Experts* feature requirement in the selection process emphasizes the importance of the modeling language's ability to support advanced modeling elements. This requirement is critical as it enables users with a high level of expertise in the field to model complex applications or domains that require more advanced constructs. In other words, users with a deeper understanding of the modeling language and its advanced features should be able to leverage their expertise to create models that accurately represent the intricacies of the system being modeled.

The *Activity-based* feature is essential for specifying the activities that constitute the business process, which is important for process automation and optimization. The *Input (source)* feature pertains to the ability to specify the sources of input to the business process, which is important for ensuring that the process is executed correctly. Finally, the *Communication* feature is essential for the specification of communication channels between activities in the business process, which is important for collaboration and coordination among stakeholders.

The decision makers have different perspectives on their feature requirements, primarily influenced by the purpose, or *goal*, for which business process models are to be used. Table 4 shows that the case study participants who indicated the feature requirements more confidently in adherence to their currently selected solution were advised of a more limited set of potentially feasible alternative solutions. In other words, many hard constraints (Must-Have and Won't-Have) on unique BPM features lead to the DSS suggesting fewer alternative solutions.

For instance, finding a BPM language that supports features for depicting and implementing process visualizations using a single diagram (such as in case study two) led to 30 BPM features being defined as hard constraints. Flexibility on the feature requirements leads to a higher number of ranked feasible solutions, illustrated in Table 5 under *Rev. 2*. For example, the relaxation of some of the BPM feature constraints in the second revision led to the *Event-driven Process Chain (EPC)* (35%) and *Unified Modeling Language Activity Diagrams (UML-AD)* (69%, 63%, 53%) being opened up as potentially feasible solutions for case studies one and four, respectively.

Regarding the validity metric, the results of the DSS match with the case study participants ranked shortlist of potentially feasible BPM languages. As such, the decision-model scores are sufficient concerning the validity of the results based on the four case studies.

Generality is an often mentioned criterion for evaluating design theory [110], which refers to the broadness of the goal addressed by the artifact, where the broader the goal, the more general the artifact [104]. As aforementioned, the goal addressed by the decision model is to assist academics in selecting a suitable BPM language for their modeling problems in research projects. The decision model includes the most prominent BPM languages, positively impacting the generality metric. However, the decision model omits *declarative* BPM languages, depriving the model of its purpose when such a language is required—negatively affecting the generality of the artifact. Despite that, we have included *declarative* as a feature so that the decision model can be easily expanded by adding declarative BPM languages in the future. Furthermore, the decision model is suitable for selecting BPM languages in the academic domain. Fortunately, this work is part of a larger research project in which additional decision models are created for other multi-criteria decision-making problems in the software domain (such as [45, 62, 64–66]).

6 Discussion

This section highlights the viewpoints of the domain experts and case study participants on the decision model. Moreover, it explains the lessons learned and our observations while researching, building, and evaluating the decision model. Finally, this study discusses threats to validity, mitigation tactics, and limitations.

6.1 Case studies

We conducted case studies with four domain experts to identify their feature requirements using the MoSCoW prioritization technique. Table 4 shows that the case study participants who defined a higher number of hard constraint features were advised a more limited set of alternative solutions. For example, the second case study participant defined 30 hard constraint features, which limited the number of potentially suitable solutions. In contrast, case study participants one and three defined relatively more soft-constraint features, leading to a broader list of potentially suitable solutions.

The infeasible solutions arose because some participants indicated their requirements as hard constraints without knowing that their desired solutions (that they used in their study before participating in this research) might not support those requirements. The limited set of BPM languages that support the hard constraint features might not have been known to the participants, leading to conflicts between the selected features and available solutions.

To address this issue, we relaxed the constraints by converting a set of hard constraints (Must-Have and Won't-Have) to soft constraints (Should-Have, Could-Have, or None). We iteratively performed this process until at least one feasible solution was found (see Table 5, *Rev. 2*). By doing so, we were able to provide the participants with alternative solutions that better aligned with their preferences and prioritized requirements.

It is important to note that identifying and prioritizing features for a BPM solution can be a complex task, and process modelers may not always be aware of the limitations of their desired solutions. Therefore, the decision model we propose in this paper can help process modelers make more informed decisions by providing a comprehensive overview of BPM languages, features, and quality characteristics.

6.2 Expert interviews

We formally coded the analysis of the interviews and literature study (the full data set is available on Mendeley Data [76]) and leveraged incremental concept development [66] to identify relevant concepts from the literature study and domain expert interviews. Next, candidate qualities and features were defined and fine-tuned during the interviews and confirmed through the literature study results.

Multiple domain experts asserted that modelers and scholars with little experience with and knowledge of BPM languages could find the decision model useful for exploring potentially suitable solutions, determining explicit and implicit feature requirements, and justifying their language selection decision(s).

According to the domain experts, experienced business process modelers usually select a BPM language based on their familiarity with it—rather than comparing a set of languages based on their characteristics. The case study participants shared this view. Nevertheless, the domain experts highlighted that the decision model proves particularly useful for evaluating and validating if the "right" BPM language has been selected. Additionally, the case study participants highlighted the appreciation of the validation of their case.

6.3 Lessons learned

BPM languages that are actively maintained by an organizational body or consortium, such as the Object Management Group (OMG) [111], are typically supported by an active community, which ensures the availability of tools and training materials. Additionally, these BPM languages tend to have a high level of maturity, providing support for BPM features such as documentation availability, evolutionary, and theoretical foundation.

BPMN stands out as a highly comprehensive language that supports most BPM features considered in the decision model (81.94%). These results are consistent with bibliometric analysis by Silva [109], which identifies BPMN as the only BPM language meeting all evaluative criteria. Similarly, Recker et al. [112] observe that BPMN (v1.0) is the only language capable of covering all aspects of business process modeling. Several other studies also describe BPMN as the de-facto standard for business process modeling today, including Entringer et al. [17], Kocbek et al. [113], and Khudori et al. [108].

The remaining graphical BPM languages, such as *EPC*, *UML-AD*, and *YAWL*, are all nearly equal in their level of expressiveness, supporting basic process modeling constructs such as control-flow, agents, and *OR*, *XOR*, and *AND* gateways. These BPM languages differ in the perspectives they cover and the goals they support. For example, YAWL is often used for defining workflows, while UML-AD specifies processes in a way that facilitates software development. In contrast, EPC is a high-level description of business operations [114].

Selecting a suitable BPM language for a large project with complex requirements is a crucial task that largely depends on the level of expressiveness of the languages. In this regard, BPMN stands out as the clear winner. Therefore, it may be more meaningful for academics to explain why they did not select BPMN rather than justifying their decision to use it for their modeling task. Nevertheless, the proposed decision model offers a novel approach that enables researchers to comprehensively identify implicit feature requirements, explore potential alternatives, and evaluate and validate their BPM language selection decision. The decision model synthesizes knowledge from various sources, presenting a reusable and comprehensive overview of BPM languages and their features, which researchers can leverage for future challenges.

Experience in using BPM languages provides valuable insights when selecting suitable BPM languages. However, modelers tend to favor the BPM languages they have used and are familiar with. Adapting to a new BPM language is often time-consuming and costly for decision makers. Therefore, the "best" BPM language for academics is usually the one they have used before. However, relying solely on familiarity poses risks, such as discovering that the selected language does not meet all desired criteria later in the modeling process. To avoid workarounds and language extensions, academics should use the proposed decision model to evaluate and validate their BPM language selection decision for their research projects.

During the development of the decision model in this study, modeling languages such as the Communicative Event Diagram (CED) [115], a component of the broader requirements engineering method named Communication Analysis, were not considered. This decision was based on the fact that modelers do not typically use CED as the primary language for process modeling. Instead, it is commonly employed to capture requirements related to communication between stakeholders, such as identifying the actors involved, messages exchanged, and communication channels used.

While such languages can complement process modeling, they lack the essential features required for process modelings, such as task/activity modeling, control flow/sequence, and event/trigger modeling. These features are critical for effectively capturing the process flow and logic of the modeled system. Therefore, using a language lacking these essential features can hinder the accuracy and reliability of the resulting model. In addition to these factors, the decision not to include CED in the decision model was also influenced by its limited use within the modeling community. Using a language that is not widely adopted by modelers may lead to challenges in communication and collaboration among stakeholders, ultimately impacting the success of the modeling effort.

6.4 Threats to validity

Assessing the validity is essential to any empirical study [45] and case study research [116]. The trustworthiness of the results, bias by the researcher(s), and the extent to which the results are true are all denoted by the validity of a study [85]. Typically, validity discussions involve construct validity, internal validity, external validity, and conclusion validity [62]. Furthermore, We adopt the classification scheme by Yin [82] to distinguish between four aspects of validity, being: (1) construct validity, (2) internal validity, (3) external validity, and (4) reliability. Below we highlight these aspects and the tactics used to mitigate the threats to the validity of this study.

Construct validity refers to the extent to which the test meets its goals [85] and whether an accurate operational measure has been used for the concepts being studied [117]. To mitigate the threats to the construct validity, we followed Majumders' six-step decision-making process [43] and the multi-criteria decision-making theory to build the decision model for the BPM language selection problem. Besides, multiple knowledge acquisition techniques have been employed to capture knowledge regarding the BPM languages: document analysis, snowballing literature review, and multiple domain expert interviews. Moreover, the effectiveness and usefulness of the proposed decision model and DSS have been evaluated using four real-world case studies related to four different articles published in conferences with a CORE rank of at least A*, A, B, or C.

Internal validity demonstrates and proves the assumed causal relationships between observations [82]. In other words, "*it determines whether the study is sound or not*" [45]. We used the following methods to mitigate the threats to the internal validity: first, we define DSS success when it, in part, aligns with the case study participants' shortlist of ranked feasible solutions. However, using the case study participant's opinion as a measurement instrument is risky, as they may have subjective opinions and lack the knowledge required to make a well-grounded judgment. We counter these risks by conducting multiple case studies and using interview protocols, assuming the case study participants handle them in their best interest.

External validity is concerned with the generalizability of the results analytically and reliably [82], referring to how the operations of a study can be repeated with the same results [117]. The decision model's usefulness has been evaluated in the academic domain, capturing knowledge from different knowledge sources without regional limitations to define the constructs and build the decision model. We hypothesize that the decision model can be generalized to other BPM language selection problems in the academic domain.

A challenge for the decision model is that the identified BPM features vary greatly depending on the perception of the expert. For example, some experts indicated that *standardized file format* and *mappability* are features of BPM languages. In contrast, other domain experts argued that they should not be considered features of BPM languages. Hence, it could be argued that a (much) larger number of experts is needed to reach a consensus about the features. We have surveyed over 50 scientific publications to concord and derive additional features and definitions from mitigating this threat. Moreover, we have only included features in the decision model, which occurred in at least four sources of knowledge. Finally, we plan to mitigate the feature consensus problem by adapting the knowledge base in the future based on feedback from decision makers using the DSS. Accordingly, knowledgeable academics can successfully apply the decision model to evaluate their decisions. Less knowledgeable academics can use the DSS to discover feature requirements and potentially feasible solutions for the modeling task.

Conclusion validity verifies whether the methods of a study, such as the data collection method, can be reproduced with similar results. Following the multi-criteria decision-making framework and Majumders' six-step decision-making process, we captured and organized the knowledge from the various sources of knowledge. The accuracy of the extracted knowledge was guaranteed by developing various interview protocols for the domain expert interviews and case studies and defining the knowledge extraction strategy and format. Following the framework and interview protocols allows us to keep the knowledge extraction process consistent and validate if the acquired knowledge has addressed the research questions.

Reliability refers to the extent to which the data and analysis depend on the specific researcher [82]. Typically, research is considered reliable when someone can examine the work and come to a similar conclusion [118]. Common threats to the reliability of a study would be the lack of a comprehensive research plan [119] and not having systematic interview questions. Consequently, these threats have been mitigated by following the MCDM theory, Majumders' six-step decision-making process, and the interview protocols, which have been made available on Mendeley Data [76].

6.5 Limitations

We find it relevant to highlight a few practical and theoretical limitations that may influence the outcomes and conclusions of our research. Due to the initial study being a Bachelor's thesis, the available time and resources were limited. In contrast, the literature review, document analysis, expert selection, expert interviews, case study selection, case definition, and evaluation of the decision model have proven to be time-consuming and exhaustive. As a result, one of the limitations of the work is that the non-Boolean feature *expert availability* is based on a single metric. In contrast, using a weighted average based on multiple metrics would have been beneficial.

Furthermore, finding academic experts that were benevolent toward participating in the study (both for the interviews and the case studies) has proven to be a challenging task due to their: (1) not responding, (2) not having a schedule that allows for such initiatives due to increasing workloads, or (3) having other priorities. As a result, the decision model has been evaluated employing four case studies, meaning the results are not directly generalizable. As per the recommendation of Eisenhardt [120], preferably, at least one more case study was conducted to develop the theory and evaluate the decision model.

Last but not least, the DSS has another problem inherent to its very nature: decision models can quickly get outdated without proper updates to them. For example, new BPM languages and extensions can arise. Moreover, evolutionary techniques could introduce new features or adapt the language to cover a more significant part of the features currently included in the decision model. However, the nature of the decision model is such that it allows for easy addition, removal, or adjustment of the BPM languages and features. Hence, the decision model is both maintainable and evolvable.

7 Related work

This section provides a comprehensive overview of evaluating BPM languages, contextualizing the proposed decision model within the existing literature. Each subsection briefly describes the work, highlighting its unique contributions and relevance to the current study. Snowballing was the primary method in this study to investigate the existing literature regarding business process modeling and strategies that address the BPM language selection problem. Table 6 summarizes a subset of selected studies that address the BPM language selection problem.

7.1 Evaluating BPM languages

The field of business process modeling has witnessed a growing interest in recent years, leading to the development of various techniques and notations for representing business processes. Numerous comparative analyses have been con-

Table 6 Compares selected studies on BPM language selection. The first two columns show the study and publication year. The third column shows the decision-making approach (DMA) used. The fourth column indicates whether the approach is MCDM. The fifth column shows if pairwise comparison (PC) was used. The sixth and seventh columns display the number of criteria and alternatives considered. The next three columns show the common quality attributes, features, and alternatives with the first row (this study). The last column shows the percentage of coverage of the considered alternatives, features, and quality attributes

Study	Year	DMA	MCDM	PC	#C	#A	#CQ	#CF	#CA	Cov. (%)
This study		DSS	Yes	No	97	23	25	72	23	100
[18]	2021	AHP	Yes	Yes	7	5	7	0	5	100
[27]	1999	Benchmarking	No	N/A	10	2	4	6	2	100
[32]	2016	ELECTRE Tri-B	Yes	No	43	3	7	18	3	61
[127]	2006	Benchmarking	No	N/A	8	7	0	8	6	93
[128]	2004	Benchmarking	No	N/A	6	12	0	6	8	78
[17]	2021	Benchmarking	No	N/A	9	4	6	3	4	100
[129]	2006	Benchmarking	No	N/A	43	3	7	18	3	61
[130]	2002	Benchmarking	No	N/A	9	3	0	9	2	92
[35]	2007	AHP	Yes	Yes	22	7	8	10	5	79
[28]	2019	MAP	No	N/A	30	28	16	5	15	62
[19]	2009	Benchmarking	No	N/A	5	7	3	2	7	100
[30]	2016	Benchmarking	No	N/A	10	6	5	2	6	81
[131]	2004	Benchmarking	No	N/A	37	5	6	21	5	76
[50]	2015	Benchmarking	No	N/A	13	3	10	2	3	94

ducted to aid scholars in selecting the most suitable BPM language for their research projects.

For instance, Rad et al. [121] proposed an evaluation framework for service-based process modeling languages in the healthcare sector. Lu and Sadiq [122] presented a critical and comprehensive analysis of two prominent modeling approaches for business processes, focusing on control-flow capabilities.

Several studies have evaluated the available BPM languages over the past few decades. Tangkawarow and Waworuntu [51] presented a comparative analysis of four BPM techniques, explaining their definition, structure, advantages, and disadvantages. Niziol et al. [123] compared and characterized various aspects of three selected modeling notations: UML, BPMN, and EPC, using the 4+1 architectural view model [124]. Abdel-Fattah et al. [29] developed an evaluation framework for the business process modeling techniques based on specific business process needs, emphasizing BPMN, UML-AD, PN, IDEF3, and RAD.

Bork et al. [125] proposed an empirical evaluation technique for evaluating the intuitiveness of domain-specific modeling languages, while Mili et al. [33] presented an overview of seventeen BPM languages and categorized the various languages from different scientific traditions. The field has also seen a growing interest in recent years, leading to the development of various techniques and notations for representing business processes.

Entringer et al. [17] conducted a bibliometric analysis to identify the most commonly portrayed BPM languages in the literature. They found that BPMN, UML, EPC, and IDEF were the most represented in the works surveyed. Aldin and de Cesare [19] conducted a comparative analysis of seven BPM techniques based on five criteria. Awadid et al. [28] proposed a context-aware roadmap with associated methodological guidelines for selecting an appropriate BPM formalism. Hommes and van Reijswoud [21] developed the Q-ME framework to assess the quality of BPM techniques, while Johansson et al. [126] analyzed four primary process-oriented modeling techniques using Moody's quality criterion for what makes a good model.

While these works have significantly contributed to the field, they do not provide a readily usable and accessible DSS for academics to select a potentially feasible BPM language for their modeling problems. Benchmarking and statistical analysis, typically time-consuming, requires a thorough knowledge of BPM languages and concepts. Additionally, decision-making based on such analysis can be complex as decision makers cannot simultaneously assess all their preferences and requirements, particularly when the number of alternatives and requirements is high.

7.2 MCDM approaches

Selecting the most suitable BPM language requires assessing multiple alternatives and criteria. Including information within a specific modeling context impacts the development and implementation of a business process [53], as well as on the decision to select or reject a BPM language [54]. Thus, the decision-making problem is approached differently by different stakeholders. The chosen BPM language(s) should address the preferences, concerns, and priorities of decision makers. In contrast to benchmarking and statistical analysis-based approaches, these approaches offer primarily generic results and comparisons, overlooking the requirements and preferences of individual decision makers.

MCDM tools and techniques are mathematical decision models that aggregate criteria, viewpoints, or features [132]. Support, a fundamental concept in MCDM, indicates that decision models are developed with active participation from the decision maker [133]. Alternatively, an iterative process is utilized to analyze decision makers' priorities and accurately describe them in a suitable decision model. This iterative and interactive modeling procedure is the underlying principle of MCDM's decision support approach, distinguishing it from statistical and optimization decision-making approaches [134].

A variety of MCDM approaches have been recently introduced by different researchers [66], of which a subset is presented below. The analytical hierarchy process (AHP) is a well-known structured method based on mathematics and psychology for organizing and analyzing MCDM problems. For instance, Guizani and Ghannouchi [18] proposed an approach that transforms the BPM language selection problem into an MCDM problem, which is then solved using the AHP multi-criteria analysis method. This framework is based on seven criteria for selecting a BPM language that best satisfies a modeler's requirements. This MCDM approach decomposes complex problems into a hierarchical system, where binary combinations are established at each level of the hierarchy [135].

Scanavachi Moreira Campos and de Almeida [32] present a rigorous and systematic framework for selecting a modeling language based on the semiotic quality framework (SEQUAL) [136] and the Elimination and Choice Translating Reality (ELECTRE) Tri-B method. ELECTRE Tri-B is an outranking method [137] that addresses sorting problems by synthesizing criteria in a unique function, thereby enabling the offset of a disadvantage of one criterion by an advantage it has over another.

Krogstie and Arnesen [138] introduce a generic quality framework that assesses decision makers' requirements and compares their supportability against a set of BPM language characteristics to identify the most suitable solution for the modeling task. Other typical MCDM approaches include The Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) [139] and Boolean Decision Trees, such as Criteria Hierarchy Trees, which allow decision makers to model complex decision problems involving multiple objectives [140].

Despite their practical use, most of these frameworks use pairwise comparisons to assess the quality of BPM languages. However, this approach becomes prohibitively time-consuming for problems with a large number of criteria. For instance, a problem with n criteria requires $\frac{n(n-1)}{2}$ pairwise comparisons [135]. As the number of criteria increases, the complexity of pairwise comparison grows exponentially, making it increasingly challenging to conduct. Furthermore, some of these methods, such as AHP, are not scalable, as modifying the sets of alternatives and criteria requires the evaluation process to be redone completely. These methods are typically only suitable for several decision criteria and alternatives.

In our study, we considered 93 alternatives, 52 quality attributes, and 117 BPM features and identified 23 alternatives, 25 quality attributes, and 72 features for building the decision model. Finally, the results of MCDM approaches are valid for a specified period and get outdated as technology advances and new BPM languages and extensions are introduced. This challenge is also present in our study, and we propose our plans for keeping the knowledge base up to date in Sect. 8.

In contrast to the named approaches, the cost of creating, evaluating, and applying the proposed decision model in this study is not penalized exponentially by the number of decision criteria and alternatives. The decision-making process is split into four maintainable phases, making it an evolvable and expandable approach [87]. Additionally, we introduce several parameters to measure and estimate the values of non-Boolean criteria, such as the popularity and tool availability of the BPM languages. Furthermore, we employed a literature review to establish a standard set of quality attributes as a reference point for determining the quality of the BPM languages. In brief, the proposed decision model addresses the essential aspects of knowledge management, such as knowledge capturing, sharing, and maintenance, to support academics with the MCDM problem of selecting a suitable BPM language for their research project(s).

8 Conclusion and future work

In this study, we modeled selecting a business process modeling language in research projects as a multi-criteria decision-making problem, which evaluates alternatives based on a set of decision criteria [43, 63]. To support academic decision makers, we presented a decision model based on the technology selection framework [44] and deeply embedded requirements engineering concepts such as the MoSCoW prioritization technique. This technique allows for the prioritization of requirements based on their relative importance to the success of a project, ensuring that the most critical requirements are given the highest priority, ultimately leading to the selection of the most suitable BPM language for the project. Additionally, by incorporating established requirements engineering concepts like MoSCoW, we increase the reliability and credibility of our approach, making it easier for other researchers to replicate our work in similar contexts.

To develop the decision model, we conducted a systematic literature review to determine uniform BPM language quality attributes. The current version of the decision model consists of 72 generic BPM features, 23 BPM languages that (partly) support these features, and 25 quality attributes that evaluate the quality of the BPM languages. We gathered feedback from ten domain experts to evaluate the decision model's efficacy, validity, and generality and conducted four academic case studies.

The results indicate that the decision model can sufficiently assist academics during the BPM language selection process in research projects and is particularly beneficial for evaluating and validating the decision maker's decision, exploring potentially feasible alternative solutions, and analyzing an extensive list of features. Our findings also align with those of other researchers, indicating that BPMN 2.0 is a highly expressive and flexible language, covering the largest number of BPM features considered in this study (82%) and considered to be "the de-facto standard for representing processes occurring in virtually every kind of organization one can think of" [107]. Therefore, it makes more sense for academics to argue why they did not select BPMN rather than discussing why they did choose BPMN for the modeling problem(s) in their research project(s).

Finally, the BPM languages and features collected from domain expert interviews, systemic literature review, and document analysis provide a comprehensive overview of the BPM languages and characteristics other researchers and academics can employ for future challenges. We have made these data available as separate datasets on Mendeley Data [76].

Our future work aims to improve the decision model by adapting the knowledge base based on feedback from decision makers using the decision support system. Additionally, we aim to introduce additional metrics for determining the values assigned to the non-Boolean BPM features. Furthermore, future studies could explore the multitude of BPM language extensions available, such as the 52 BPMN extensions uncovered in a literature survey by Zarour et al. [141].

Acknowledgements We would like to express our gratitude to the ten domain experts who participated in and contributed to this research project. We are also thankful to Dr. Xu, Dr. Klinger, Dr. Ladleif, and Dr. España for participating as case study participants and for granting us permission to use their publications [94, 95, 100, 101] as case studies to evaluate the decision model. Finally, we appreciate the efforts of the journal editors and reviewers in providing valuable feedback and suggestions to improve the quality of this manuscript.

References

1. Havey, M.: Essential Business Process Modeling. O'Reilly Media, Inc., Sebastopol (2005)
2. Johansson, L.-O., Wärja, M., Carlsson, S.: An evaluation of business process model techniques, using moody's quality criterion for a good diagram. In: BIR 2012: Emerging Topics in Business Informatics Research 2012, Nizhny Novgorod, Russia, September 24-26, 2012, vol. 963, pp. 54–64 (2012). Rheinisch-Westfaelische Technische Hochschule Aachen, Lehrstuhl Informatik V
3. de Oca, I.M.-M., Snoeck, M., Reijers, H.A., Rodríguez-Morffi, A.: A systematic literature review of studies on business process modeling quality. Inf. Softw. Technol. **58**, 187–205 (2015)
4. Sánchez-González, L., García, F., Ruiz, F., Mendling, J.: Quality indicators for business process models from a gateway complexity perspective. Inf. Softw. Technol. **54**(11), 1159–1174 (2012)
5. Indulska, M., Green, P., Recker, J., Rosemann, M.: Business process modeling: perceived benefits. In: International Conference on Conceptual Modeling, pp. 458–471. Springer (2009)
6. Reijers, H.A.: Business process management: the evolution of a discipline. Comput. Ind. **126**, 103404 (2021)
7. Campos, C.S., Daher, S.F.D., Almeida, A.T.: New patents on business process management information systems and decision support. Recent Patents Comput. Sci. **4**(2), 91–97 (2011)
8. Becker, J., Clever, N., Holler, J., Neumann, M.: Business process management in the manufacturing industry: ERP replacement and ISO 9001 recertification supported by the icebricks method. In: Business Process Management Cases, pp. 413–429. Springer (2018)
9. Davies, I., Green, P., Rosemann, M., Indulska, M., Gallo, S.: How do practitioners use conceptual modeling in practice? Data Knowl. Eng. **58**(3), 358–380 (2006)
10. Ruiz, F., Garcia, F., Calahorra, L., Llorente, C., Gonçalves, L., Daniel, C., Blobel, B.: Business process modeling in healthcare. Stud. Health Technol. Inform. **179**, 75–87 (2012)
11. Harmon, P., Wolf, C.: Business process modeling survey. Bus. Process Trends **36**(1), 1–36 (2011)
12. Harmon, P., Garcia, J.: The bptrends report. The state of business process management (2020)
13. Pinggera, J., Soffer, P., Fahland, D., Weidlich, M., Zugal, S., Weber, B., Reijers, H.A., Mendling, J.: Styles in business process modeling: an exploration and a model. Softw. Syst. Model. **14**(3), 1055–1080 (2015)
14. Curtis, B., Kellner, M.I., Over, J.: Process modeling. Commun. ACM **35**(9), 75–90 (1992)
15. Liu, D.-R., Shen, M.: Business-to-business workflow interoperation based on process-views. Decis. Support Syst. **38**(3), 399–419 (2004)
16. Devillers, M.: Business process modeling as a means to bridge the business-it divide. Master's thesis, Radboud University Nijmegen (2011)

17. Entringer, T.C., Ferreira, A.d.S., Nascimento, D.C.d.O.: Comparative analysis of the main business process modeling methods: a bibliometric study. Gestão & Produção **28** (2021)
18. Guizani, K., Ghannouchi, S.A.: An approach for selecting a business process modeling language that best meets the requirements of a modeler. Procedia Comput. Sci. **181**, 843–851 (2021)
19. Aldin, L., De Cesare, S.: A comparative analysis of business process modelling techniques. In: UK Academy for Information Systems Conference Proceedings (2009)
20. Nagm-Aldeen, Y., Abdel-Fattah, M.A., El-Khedr, A.: A literature review of business process modeling techniques. Int. J. **5**(3), 43–47 (2015)
21. Hommes, B.-J., Van Reijswoud, V.: Assessing the quality of business process modelling techniques. In: Proceedings of the 33rd annual Hawaii international conference on system sciences, p. 10 (2000). IEEE
22. Committee, O.M.G.B.T., et al.: Business Process Model and Notation, Version 2.0. OMG, Object Management Group lnc., Needham (2011)
23. Booch, G., Rumbaugh, J., Jackobson, I.: UML: unified modeling language. Versão (1997)
24. Hee, K.M.v., Sidorova, N., Werf, J.M.v.d.: Business process modeling using petri nets. In: Transactions on Petri Nets and Other Models of Concurrency VII, pp. 116–161. Springer (2013)
25. Scheer, A.-W., Thomas, O., Adam, O.: Process modeling using event-driven process chains. (2005)
26. Van Der Aalst, W.M., Ter Hofstede, A.H.: Yawl: yet another workflow language. Inf. Syst. **30**(4), 245–275 (2005)
27. Luo, W., Tung, Y.A.: A framework for selecting business process modeling methods. Ind. Manag. Data Syst. **99**, 312–319 (1999)
28. Awadid, A., Nurcan, S., Ghannouchi, S.A.: On leveraging the fruits of research efforts in the arena of business process modeling formalisms: a map-driven approach for decision making. Softw. Syst. Model. **18**(3), 1905–1930 (2019)
29. Abdel-Fattah, M.A., Khedr, A.E., Aldeen, Y.N.: An evaluation framework for business process modeling techniques. Int. J. Comput. Sci. Inf. Secur. (IJCSIS) **15**(5), 382–392 (2017)
30. Araújo, M.B., Gonçalves, R.F.: Selecting a notation to modeling business process: a systematic literature review of technics and tools. In: IFIP International Conference on Advances in Production Management systems, pp. 198–205 (2016). Springer
31. Pereira, J.L., Silva, D.: Business process modeling languages: A comparative framework. In: New Advances in Information Systems and Technologies, pp. 619–628. Springer (2016)
32. Scanavachi Moreira Campos, A.C., de Almeida, A.T.: Multicriteria framework for selecting a process modelling language. Enterp. Inf. Syst. **10**(1), 17–32 (2016)
33. Mili, H., Tremblay, G., Jaoude, G.B., Lefebvre, É., Elabed, L., Boussaidi, G.E.: Business process modeling languages: sorting through the alphabet soup. ACM Comput. Surv. (CSUR) **43**(1), 1–56 (2010)
34. Zimoch, M., Pryss, R., Schobel, J., Reichert, M.: Eye tracking experiments on process model comprehension: lessons learned. In: Enterprise, Business-Process and Information Systems Modeling, pp. 153–168. Springer (2017)
35. Kaschek, R., Pavlov, R., Shekhovtsov, V.A., Zlatkin, S.: Characterization and tool supported selection of business process modeling methodologies. In: Technologies for Business Information Systems, pp. 25–37. Springer (2007)
36. Ottensooser, A., Fekete, A., Reijers, H.A., Mendling, J., Menictas, C.: Making sense of business process descriptions: An experimental comparison of graphical and textual notations. J. Syst. Softw. **85**(3), 596–606 (2012)
37. Krogstie, J.: Perspectives to process modeling. In: Business Process Management, pp. 1–39. Springer (2013)
38. Greenspan, S., Mylopoulos, J., Borgida, A.: On formal requirements modeling languages: RML revisited. In: Proceedings of 16th International Conference on Software Engineering, pp. 135–147 (1994). IEEE
39. Menzli, L.J., Ghannouchi, S.A., Ghézala, H.H.B.: A guidance process for the selection of business process modelling techniques for the revised business process reengineering. J. Converg. Inf. Technol. **2**(2), 79–88 (2007)
40. Giaglis, G.M.: A taxonomy of business process modeling and information systems modeling techniques. Int. J. Flex. Manuf. Syst. **13**(2), 209–228 (2001)
41. Phalp, K.T.: The cap framework for business process modelling. Inf. Softw. Technol. **40**(13), 731–744 (1998)
42. Rosemann, M.: Potential pitfalls of process modeling: part A. Bus. Process Manag. J. **12**, 249–254 (2006)
43. Majumder, M.: Multi criteria decision making. In: Impact of Urbanization on Water Shortage in Face of Climatic Aberrations, pp. 35–47. Springer (2015)
44. Farshidi, S., Jansen, S., De Jong, R., Brinkkemper, S.: A decision support system for cloud service provider selection problems in software producing organizations. In: IEEE 20th Conference on Business Informatic (CBI), vol. 1(1), pp. 139–148 (2018)
45. Farshidi, S., Jansen, S., Deldar, M.: A decision model for programming language ecosystem selection: seven industry case studies. Inf. Softw. Technol. **139**, 106640 (2021)
46. Johnson, R.B., Onwuegbuzie, A.J.: Mixed methods research: A research paradigm whose time has come. Educ. Res. **33**(7), 14–26 (2004)
47. Gruber, T.R.: Automated knowledge acquisition for strategic knowledge. Mach. Learn. **4**, 293–336 (2005)
48. Garg, R.: MCDM-based parametric selection of cloud deployment models for an academic organization. IEEE Trans. Cloud Comput. **10**, 863–871 (2020)
49. Fitzgerald, D.R., Mohammed, S., Kremer, G.O.: Differences in the way we decide: the effect of decision style diversity on process conflict in design teams. Personal. Individ. Differ. **104**, 339–344 (2017)
50. Geyer, R.W., Fourie, C.J.: Determining the suitability of a business process modelling technique for a particular application: case study. S. Afr. J. Ind. Eng. **26**(1), 252–266 (2015)
51. Tangkawarow, I., Waworuntu, J.: A comparative of business process modelling techniques. In: IOP Conference Series: Materials Science and Engineering, vol. 128, p. 012010 (2016). IOP Publishing
52. Recker, J., Dreiling, A.: Does it matter which process modelling language we teach or use? an experimental study on understanding process modelling languages without formal education. In: ACIS 2007 Proceedings, vol. 45 (2007)
53. Saidani, O., Nurcan, S.: Towards context aware business process modelling. In: 8th Workshop on Business Process Modeling, Development, and Support (BPMDS'07), CAiSE, vol. 7, p. 1 (2007). Citeseer
54. Awadid, A., Nurcan, S., Ghannouchi, S.A.: Towards a decision-support system for selecting the appropriate business process modeling formalism: a context-aware roadmap. In: Enterprise, Business-Process and Information Systems Modeling, pp. 239–256. Springer (2017)
55. Nielen, A., Jeske, T., Schlick, C., Arning, K., Ziefle, M.: Interdisciplinary assessment of process modeling languages applicable for small to medium-sized enterprises. In: 8th International Conference on Computing, Communications and Control Technologies: CCCT, pp. 47–52 (2010)
56. Dodgson, J.S., Spackman, M., Pearman, A., Phillips, L.D.: Multi-criteria Analysis: A Manual. Department of the Environment, Transport and the Regions, London (2009)

57. Aruldoss, M., Lakshmi, T.M., Venkatesan, V.P.: A survey on multi criteria decision making methods and its applications. Am. J. Inf. Syst. **1**(1), 31–43 (2013)
58. Hafezalkotob, A., Hafezalkotob, A., Liao, H., Herrera, F.: An overview of multimoora for multi-criteria decision-making: theory, developments, applications, and challenges. Inf. Fusion **51**, 145–177 (2019)
59. Zhou, Y.: Multi-criteria decision making in software development: a systematic literature review. Master's thesis, University of Oulu, Faculty of Information Technology and Electrical Engineering, Department of Information Processing (2015)
60. Ishizaka, A., Siraj, S.: Are multi-criteria decision-making tools useful? an experimental comparative study of three methods. Eur. J. Oper. Res. **264**(2), 462–471 (2018)
61. Gavade, R.K.: Multi-criteria decision making: An overview of different selection problems and methods. Int. J. Comput. Sci. Inf. Technol. **5**(4), 5643–5646 (2014)
62. Farshidi, S.: Multi-criteria decision-making in software production. Ph.D. thesis, Utrecht University (2020)
63. Triantaphyllou, E.: Multi-criteria decision making methods. In: Multi-criteria Decision Making Methods: A Comparative Study, pp. 5–21. Springer (2000)
64. Farshidi, S., Jansen, S., Fortuin, S.: Model-driven development platform selection: four industry case studies. Softw. Syst. Model. **20**, 1–27 (2021)
65. Farshidi, S., Jansen, S., De Jong, R., Brinkkemper, S.: Multiple criteria decision support in requirements negotiation. In: REFSQ Workshops (2018)
66. Baninemeh, E., Farshidi, S., Jansen, S.: A decision model for decentralized autonomous organization platform selection: three industry case studies. Blockchain: Research and Applications, 100127 (2023)
67. Farshidi, S., Jansen, S., España, S., Verkleij, J.: Decision support for blockchain platform selection: Three industry case studies. IEEE Trans. Eng. Manag. **67**(4), 1109–1128 (2020)
68. Farshidi, S., Jansen, S.: A decision support system for pattern-driven software architecture. In: European Conference on Software Architecture, pp. 68–81 (2020). Springer
69. Fortus, D., Krajcik, J., Dershimer, R.C., Marx, R.W., Mamlok-Naaman, R.: Design-based science and real-world problem-solving. Int. J. Sci. Educ. **27**(7), 855–879 (2005)
70. Simon, H.A.: The Sciences of the Artificial, vol. 136. MIT Press, Cambridge (1996)
71. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. Manag. Inf. Syst. Q. **28**(1), 6 (2008)
72. Hevner, A., Chatterjee, S.: Design science research in information systems. In: Design Research in Information Systems, pp. 9–22. Springer (2010)
73. Chen, W.K.: The Electrical Engineering Handbook. Elsevier, Amsterdam (2004)
74. Myers, M.D., Newman, M.: The qualitative interview in is research: examining the craft. Inf. Organ. **17**(1), 2–26 (2007)
75. Harvey, W.S.: Strategies for conducting elite interviews. Qual. Res. **11**(4), 431–441 (2011)
76. Kwantes, I.B., Farshidi, S., Jansen, S.: A decision model for business process modeling language selection. Mendeley Data (2022). https://doi.org/10.17632/z2kkv2cfj8z.1
77. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. Citeseer (2007)
78. Mantyla, M.V., Adams, B., Khomh, F., Engstrom, E., Petersen, K.: On rapid releases and software testing: a case study and a semi-systematic literature review. Empir. Softw. Eng. **20**(5), 1384–1425 (2015)
79. Wohlin, C.: Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, pp. 1–10 (2014)
80. Wnuk, K., Garrepalli, T.: Knowledge management in software testing: a systematic snowball literature review. e-Inform. Softw. Eng. J. **12**(1), 51–78 (2018)
81. Corbin, J., Strauss, A.: Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. Sage Publications, Thousand Oaks (2014)
82. Yin, R.: Case Study Research and Applications: Design and Methods. Sage Publications, Thousand Oaks (2017)
83. Pinfield, L.T.: A field evaluation of perspectives on organizational decision making. Adm. Sci. Q. **31**(3), 365–388 (1986)
84. Gustafsson, J.: Single case studies vs. multiple case studies: a comparative study (2017)
85. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng. **14**(2), 131–164 (2009)
86. DSDM Consortium, R., et al.: The DSDM agile project framework handbook. Ashford, Kent, UK: DSDM Consortium (2014)
87. Farshidi, S., Jansen, S., de Jong, R., Brinkkemper, S.: A decision support system for software technology selection. J. Decis. Syst. **27**(sup1), 98–110 (2018)
88. Haisjackl, C., Barba, I., Zugal, S., Soffer, P., Hadar, I., Reichert, M., Pinggera, J., Weber, B.: Understanding declare models: strategies, pitfalls, empirical results. Softw. Syst. Model. **15**(2), 325–352 (2016)
89. Breitenmoser, R., Keller, T.: Case management model and notation-a showcase. Eur. Sci. J. **11**(25), 332–347 (2015)
90. Iso, I.: Iec 25010: 2011 systems and software engineering-systems and software quality requirements and evaluation (square)-system and software quality models. Int. Organ. Standard. **34**, 2910 (2011)
91. ISO, I.: ISO standard 9126: Software engineering-product quality, parts 1, 2 and 3. International Organization for Standardization (2003)
92. for Standardization, I.O.: Systems and software engineering - systems and software quality requirements and evaluation (square)—system and software quality models, ISO/IEC 25010:2011 edn. International Organization for Standardization (2011). https://www.iso.org/standard/35733.html
93. Clegg, D., Barker, R.: Case method fast-track: a rad approach. Addison-Wesley Longman Publishing Co., Inc., Reading (1994)
94. Xu, X., Bandara, H.D., Lu, Q., Weber, I., Bass, L., Zhu, L.: A decision model for choosing patterns in blockchain-based applications. In: 2021 IEEE 18th International Conference on Software Architecture (ICSA), pp. 47–57 (2021). IEEE
95. Klinger, P., Bodendorf, F.: Blockchain-based cross-organizational execution framework for dynamic integration of process collaborations. In: Wirtschaftsinformatik (Zentrale Tracks), pp. 1802–1817 (2020)
96. Group, O.M.: Business Process Model and Notation (BPMN 2.0). Object Management Group, OMG (2010). https://www.omg.org/spec/BPMN/2.0
97. Mendling, J., Reijers, H.A., Recker, J.: Activity labeling in process modeling: empirical insights and recommendations. Inf. Syst. **35**(4), 467–482 (2010)
98. De Man, H.: Case management: a review of modeling approaches. BPTrends, January, vol. 2009 (2009)
99. Wagner, G., Seck, M., McKenzie, F.: Process modeling for simulation: observations and open issues. In: 2016 Winter Simulation Conference (WSC), pp. 1072–1083 (2016). IEEE
100. Ladleif, J., Weber, I., Weske, M.: External data monitoring using oracles in blockchain-based process execution. In: International Conference on Business Process Management, pp. 67–81 (2020). Springer

101. España, S., González, T., Grabis, J., Jokste, L., Juanes, R., Valverde, F.: Capability-driven development of a soa platform: a case study. In: Advanced Information Systems Engineering Workshops: CAiSE 2014 International Workshops, Thessaloniki, Greece, June 16–20, 2014. Proceedings 26, pp. 100–111 (2014). Springer
102. Venable, J., Pries-Heje, J., Baskerville, R.: A comprehensive framework for evaluation in design science research. In: International Conference on Design Science Research in Information Systems, pp. 423–438 (2012). Springer
103. Straub, D., Boudreau, M.-C., Gefen, D.: Validation guidelines for is positivist research. Commun. Assoc. Inf. Syst. **13**(1), 24 (2004)
104. Aier, S., Fischer, C.: Criteria of progress for information systems design theories. IseB **9**(1), 133–172 (2011)
105. Munteanu, F., Purcarea, A.: A theoretical comparison upon the use of uml or bpmn in the modeling of logistic processes and the management of logistical operations. Scientific Bulletin" Mircea cel Batran" Naval Academy **22**(2), 53–60 (2019)
106. Ko, R.K., Lee, S.S., Lee, E.W.: Business process management (BPM) standards: a survey. Bus. Process Manag. J. **15**, 744–791 (2009)
107. Chinosi, M., Trombetta, A.: BPMN: an introduction to the standard. Comput. Stand. Interfaces **34**(1), 124–134 (2012)
108. Khudori, A.N., Kurniawan, T.A., Ramdani, F.: Quality evaluation of EPC to BPMN business process model transformation. JITeCS (J. Inf. Technol. Comput. Sci.) **5**(2), 207–220 (2020)
109. Silva, D.: Business process models: comparative analysis of notations (master's thesis). Master's thesis, Universidade do Minho, Portugal (2015)
110. Prat, N., Comyn-Wattiau, I., Akoka, J.: Artifact evaluation in information systems design-science research-a holistic view. PACIS **23**, 1–16 (2014)
111. Group, O.M., et al.: Object Management Group (OMG) (2013)
112. Recker, J., Rosemann, M., Indulska, M., Green, P.: Business process modeling-a comparative analysis. J. Assoc. Inf. Syst. **10**(4), 1 (2009)
113. Kocbek, M., Jošt, G., Heričko, M., Polančič, G.: Business process model and notation: the current state of affairs. Comput. Sci. Inf. Syst. **12**(2), 509–539 (2015)
114. Decker, G., Mendling, J.: Process instantiation. Data Knowl. Eng. **68**(9), 777–792 (2009)
115. España, S., González, A., Pastor, Ó.: Communication analysis: a requirements engineering method for information systems. In: Advanced Information Systems Engineering: 21st International Conference, CAiSE 2009, Amsterdam, The Netherlands, June 8–12, 2009. Proceedings 21, pp. 530–545 (2009). Springer
116. Perry, D.E., Sim, S.E., Easterbrook, S.M.: Case studies for software engineers. In: Proceedings. 26th International Conference on Software Engineering, pp. 736–738 (2004). IEEE
117. Tumele, S.: Case study research. Int. J. Sales Retail. Mark. **4**(9), 68–78 (2015)
118. Ihantola, E.-M., Kihn, L.-A.: Threats to validity and reliability in mixed methods accounting research. Qual. Res. Account. Manag. **8**, 39–58 (2011)
119. Ryan, B., Scapens, R., Theobald, M.: Research Method & Methodology in Finance & Accounting (Thomson, London), 2nd edn., p. 256. Cengage Learning EMEA, England (2002)
120. Eisenhardt, K.M.: Building theories from case study research. Acad. Manag. Rev. **14**(4), 532–550 (1989)
121. Rad, A.A., Benyoucef, M., Kuziemsky, C.E.: An evaluation framework for business process modeling languages in healthcare. J. Theor. Appl. Electron. Commer. Res. **4**(2), 1–19 (2009)
122. Lu, R., Sadiq, S.: A survey of comparative business process modeling approaches. In: Business Information Systems: 10th International Conference, BIS 2007, Poznan, Poland, April 25–27, 2007. Proceedings 10, pp. 82–94 (2007). Springer
123. Nizioł, M., Wisniewski, P., Kluza, K., Ligeza, A.: Characteristic and comparison of UML, BPMN and EPC based on process models of a training company. Ann. Comput. Sci. Inf. Syst. **26**, 193–200 (2021)
124. Kruchten, P.B.: The 4+ 1 view model of architecture. IEEE Softw. **12**(6), 42–50 (1995)
125. Bork, D., Schrüffer, C., Karagiannis, D.: Intuitive understanding of domain-specific modeling languages: proposition and application of an evaluation technique. In: International Conference on Conceptual Modeling, pp. 311–319 (2019). Springer
126. Johansson, L.-O., Wärja, M., Carlsson, S.: An evaluation of business process model techniques, using moody's quality criterion for a good diagram. In: BIR 2012: Emerging Topics in Business Informatics Research 2012, Nizhny Novgorod, Russia, September 24–26, 2012, vol. 963, pp. 54–64 (2012). Rheinisch-Westfaelische Technische Hochschule Aachen, Lehrstuhl Informatik V
127. List, B., Korherr, B.: An evaluation of conceptual business process modelling languages. In: Proceedings of the 2006 ACM Symposium on Applied Computing, pp. 1532–1539 (2006)
128. Aguilar-Saven, R.S.: Business process modelling: review and framework. Int. J. Prod. Econ. **90**(2), 129–149 (2004)
129. Nysetvold, A.G., Krogstie, J.: Assessing business process modeling languages using a generic quality framework. In: Advanced Topics in Database Research, Volume 5, pp. 79–93. IGI Global (2006)
130. Söderström, E., Andersson, B., Johannesson, P., Perjons, E., Wangler, B.: Towards a framework for comparing process modelling languages. In: International Conference on Advanced Information Systems Engineering, pp. 600–611 (2002). Springer
131. Ying, C., Hong, L., Zhengchuan, X.: An evaluation framework for inter-organizational business process modelling techniques. In: Proceedings of the 8th Pacific Asia Conference on Information Systems, pp. 04–12 (2004). Citeseer
132. Floudas, C.A., Pardalos, P.M.: Encyclopedia of Optimization. Springer, Berlin (2008)
133. Dvořák, O., Pergl, R., Kroha, P.: Affordance-driven software assembling. In: Enterprise Engineering Working Conference, pp. 39–54 (2018). Springer
134. Gil-Aluja, J.: The Interactive Management of Human Resources in Uncertainty, vol. 11. Springer, Berlin (2013)
135. Saaty, T.L.: How to make a decision: the analytic hierarchy process. Eur. J. Oper. Res. **48**(1), 9–26 (1990)
136. Krogstie, J., Sølvberg, A.: Information Systems Engineering—Conceptual Modeling in a Quality Perspective. The Norwegian University of Science and Technology, Trondheim (2003)
137. Greco, S., Figueira, J., Ehrgott, M.: Multiple Criteria Decision Analysis, vol. 37. Springer, Berlin (2016)
138. Krogstie, J., de Flon Arnesen, S.: Assessing enterprise modeling languages using a generic quality framework. In: Information Modeling Methods and Methodologies: Advanced Topics in Database Research, pp. 63–79. IGI Global (2005)
139. Lai, Y.-J., Liu, T.-Y., Hwang, C.-L.: Topsis for MODM. Eur. J. Oper. Res. **76**(3), 486–500 (1994)
140. Elias, M., Shahzad, K.: Using multi-criteria decision making to choose process representation format for a process repository. In: International Conference on Business Information Systems, pp. 19–24 (2010). Springer
141. Zarour, K., Benmerzoug, D., Guermouche, N., Drira, K.: A systematic literature review on BPMN extensions. Bus. Process Manag. J. **26**, 1473–1503 (2019)

Siamak Farshidi is an assistant professor at Utrecht University's Department of Information and Computer Science. From 2020 to 2022, he worked as a postdoctoral researcher at the University of Amsterdam, focusing on cross-domain search engines within the ENVRI-FAIR project. During his Ph.D. studies from 2016 to 2020, Siamak conducted research on multi-criteria decision-making problems in software production, with a particular focus on database technology and cloud service provider selection. Currently, his research interests revolve around recommender systems, search engines, and decision-making in software engineering.

Izaak Beer Kwantes is a software practitioner specializing in low-code platforms and web development. He obtained a Bachelor's degree in Information Science from Utrecht University, focusing on business-IT and entrepreneurship, between 2018 and 2022. Currently, Izaak is a Business Engineer at Blueriq, focusing on governance, finance, and insurance. Before his current position, he founded izaakbeer.nl and served as both Growth Marketeer and Web Developer from 2020 to 2022. He successfully led digital growth initiatives and managed digital business operations for various SME clients during this period.

Slinger Jansen is an associate professor at Utrecht University, leading the Software Ecosystems Security research group. He is a prominent researcher in the field of software ecosystems and a co-founder of the International Conference on Software Business and International Workshop on Software Ecosystems. He has authored and edited several books, including "Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry." Slinger Jansen is also an associate editor for the Empirical Software Engineering journal. In addition to his academic work, he actively supports startups and serves on advisory boards for multiple ventures.