International Journal on Advanced Science Engineering Information Technology

Vol.12 (2022) No. 1
ISSN: 2088-5334

Mapping Study on Traceability Between BPMN Models and Source Code

Olga Lucero Vega-Márquez [a,b,*], Helga Duarte [a], Victor Collazos [a]

[a] Departamento de Ingeniería de Sistemas e Industrial, Universidad Nacional de Colombia, Grupo de Investigación ColSwE, Facultad de Ingeniería, Carrera 45 N° 26-85 - Edificio Uriel Gutiérrez. ZIP CODE 11321, Bogotá, Colombia
[b] Facultad de Ciencias Básicas e Ingeniería, Universidad de los Llanos, Km 12 vía Pto López, ZIP CODE 500017, Villavicencio, Colombia
Corresponding author: *ovega@unal.edu.co

Abstract— **Considering the importance of traceability in software development and that new development approaches have emerged to adapt to the dynamic and innovative market environment, we identified the need to examine traceability in business process-driven software development. Specifically, we are interested in traceability when using the BPMN language for modelling the business processes, which are then executed by integrating different software artifacts deployed from a source code. The purpose of this research is to review traceability from BPMN models to source code literature to describe the progress of this area by identifying four research questions regarding approaches, technologies, techniques, traced artifacts, and tools. In the search stage, two hundred eighty-six primary studies were obtained through a systematic mapping study. After two screenings and applying inclusion/exclusion criteria, we obtained 24 relevant papers. Strictly within the framework of these works, we answered our four research questions: we found MDE as the main approach, SOA as the main technology, graphs use as an emerging technique for managing versioning and change impact analysis, and extension of existing tools for providing traceability management. As for traced artifacts, variety is commonplace. Therefore, we consider it prudent to classify traced artifacts as high-level, low-level, and BPMN-level models. Additionally, by classifying the papers with the SwEBoK, we found that issues, such as cost estimation and traceability in testing, are topics with little research and almost no development.**

Keywords— **Business process model; BPMN; traceability.**

Manuscript received 28 Apr. 2021; revised 16 Sep. 2021; accepted 25 Jan. 2022. Date of publication 28 Feb. 2022.
IJASEIT is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.

I. INTRODUCTION

Nowadays, businesses are characterized by a dynamic and innovative environment. This means that the associated software production must rapidly adapt to meet these new market conditions. Additionally, users have increasingly specific needs and are more demanding in terms of quality. In this sense, innovative approaches have emerged to facilitate software development with the idea of obtaining applications in a much shorter time. The advantage of such applications is that the maintenance and evolution tasks are easier to carry out. One of the most widely accepted approaches is the development of software applications (and even an entire information system), using the business process - BP strategy along with its respective methodology for modeling (e.g., Business Process Modeling and Notation/BPMN) and managing them (Business Process Management/ BPM). One of the advantages of implementing business processes in organizations is the ease with which they can be modified or improved to adapt to change, which implies a reduction in time and resources for the production output. Therefore, the implementation of business processes is also a software development project that, as any such project, evolves over time and requires resources for its proper management.

In this work, we are interested in knowing the advances in evolving software when the enterprise is driven using a BPM strategy, which involves using BPMN models and integrating different software solutions for executing them. Furthermore, we are interested in the use of traceability from BPMN models to evolve the integrated software for its execution. Consequently, we seek to know "what are the existing approaches to manage traceability between BPMN models and source code." We performed a systematic mapping study on traceability involving BPMN to find answers to this question. Particularly, we studied software development traceability where BPMN models are used to describe the business processes, and then they are used as raw material for developing the software that supports the business processes. This paper is divided into the following sections. In section 2, we describe and analyze the main subjects of our research. Section 3 presents our research methodology, followed by the results and data analysis in section 4. Section 5 is intended to answer our research questions, which are limited by the threats to validity that we present in Section 6. Finally, we outline our conclusions in section 7.

II. Materials and Method

Next, we summarize a background for giving a context and facilitate the research methodology explanation and its application in this study.

A. Background

Here we introduce the Business Process concept, then we briefly describe (a) the Business Process Management discipline as the current trend for managing an organization guided by their BP. As part of BPM, we present the (b) Business Process and Model Notation – BPMN language used to communicate how business processes are performed in an organization. This section also explains the (c) BP-based Software Development and an example of how the BP models described using BPMN, and the software that supports the modeled BPs are related. These relations center a (d) traceability between BPMN models and a source code.

A business process - BP is the way in which different activities, events, and actors are related and work to produce the desired outcome, i.e., a product, a service, or anything else related to the objective of the process [18]. An enterprise driven by processes improves the organization of activities, events, and participants to produce better results. Although each activity must be performed efficiently, the process management focuses on them, the events, and the participants.

In this context, Business Process Management - BPM is the discipline that studies how an organization works based on its processes. To outperform the competition, organizations must evaluate their processes, improve the quality of their results, reduce delivery time and costs, and react by evolving the process itself to respond to new requirements or innovation initiatives. A BP performance can be measured in time, cost, resource use, or any other important metric for the organization. BPM provides a well-developed set of principles and practices to implement for business process improvement [1].

For communicating how business processes are done and to understand how activities, events and participants are related. The OMG has developed the graphical BPMN language standard [2]. This language has become the de facto standard for communicating what and how the business works. Consequently, hence BPMN models became part of the software development process, making it easier for the software development team to understand why, what for, and how the software supporting BPs should work.

Currently, there is a growing industry on Business Process Management, according to a recent survey about the state of BPM, published in 2020 by BP Trends [3] and analysis of Gartner and Forrester reports, also in 2020, by Sahay *et al.* [4].

With the advent of BPM and the recognized relevance of business processes in an organization, the development of software-based on business processes became a trend in companies by associating the processing logic and the required software for executing the different activities in the process [5]. Next, the need to execute models arose, or at least the translation between activity in BPMN language to an understandable machine language [6].

It is worth noting that the software supports several activities in a BPMN model. When the process is in execution, each activity supported by software must know what associated executable software is running. This information is the execution link of the activity. A change in the activity may involve a change in the software that supports it. Therefore, it is fundamental to know what that software is. At any moment in the life cycle of a process, it can change, and the change could require updating the software that supports the activities involved in it. The relationships between these artifacts represent the BPMN traceability. Although recent literature reviews have been done on software traceability[7], [8], they are not centered in business process models as the artifacts driving the software development. Therefore, we present an overview of the research in software traceability between BPMN models and source code in this work.

B. Research Methodology

To conduct the Mapping Study (MS), we followed the previous guidelines [9]–[14]. Figure 1 depicts the process we followed and the expected output in each step. In the following subsections, we describe the specifics of each step.

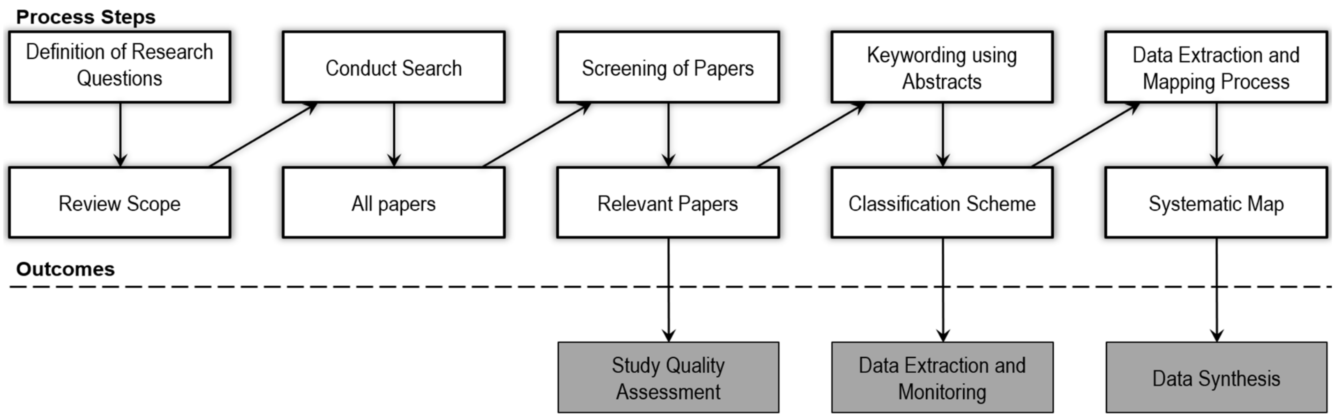

Fig. 1 MS process, adapted from Petersen [12]

1) Definition of Research Questions: This methodology step starts explaining the need to conduct the literature review, which derives from the definition of the research questions. The graphical language Business Process Model and Notation (BPMN) [1], [2] has been widely used by organizations for understanding and communicating their business processes. BPMN can help software developers understand the organization's processes, which are finally the raison d'être of the software. This MS aims to gather an overview of the research in software traceability between BPMN models and source code. Software traceability has been consolidated as an open issue in software engineering -SE, which is a discipline with a potential for empowering developers' teams "to develop higher-quality products at increasing levels of complexity and scale"[15]–[17]. The goal is to summarize the existing evidence of traceability approaches that manage trace links relating BPMN models with other software artifacts. We are interested in identifying research and results available within this kind of traceability. In order to address traceability issues, we must delimit the set of artifacts involved in the software process development. We intend to identify open issues and the way how researchers have studied traceability between BPMN and code.

To meet the need stated above, we formulated the next research questions as follows:

- RQ1 What are the existing approaches for managing traceability between BPMN models and source code?
- RQ2 For these approaches, what are the used technologies, traced artifacts, proposed techniques, and developed tools?
- RQ3 How have the approaches been evaluated, and what have been the results?
- RQ4 What are the current problems faced on traceability between BPMN and source code?

These research questions will be the guide to organize and classify the obtained studies. The main short-term goal is to find insights into the current challenges in the area of traceability between BPMN models and source code and to establish future directions.

2) Review Scope: Once we defined our research questions, we defined the scope of our study by establishing the sources to consult, the filters to use, and the search string. As sources for our literature search, we used the main scientific databases for software engineering publications: IEEEXplore, ACM Digital Library, and Web of Science. In each source, we used the same filters for (i) publication period, (ii) type of papers, (iii) type of publication, and (iv) topics. As for the publication period, we searched between 2004 and2020. The lower limit year was the publication of version 1.0 of the BPMN language by the Business Process Management Initiative [18]–[20]. We looked for research or early access papers published in journals or conferences on business process management and software engineering topics. The last restriction was used to avoid papers related to the acronym BPM denoting "Beats per minute" in cardiovascular topics, "Bit Patterned Media" in recording channels, or "Beam Pumping Motor Systems" (BPMSs) in mechanical engineering, among others. The way search engines organize topics is different, therefore, the selection of topics used is shown in the search string Table 1.

TABLE I
SOURCES, QUERIES AND # AND % OF PAPERS IN EACH ONE

Source	**Query** **String - Filters - Categories**	**#papers** **#**	 **%**
WoS	(from Web of Science Core Collection) You searched for: ((TS=(bpm$ AND (trace OR traceability OR link OR tracing OR tracking OR "answer set" OR association OR "reference set" OR ("source artifact" AND "target artifact"))) NOT TS = ("beam propagating method" OR "beats/min" OR heart OR "beat per minute" OR heart) AND PY=(2004-2019))) AND LANGUAGE: (English) AND DOCUMENT TYPES: (Article OR Abstract of Published Item OR Book Chapter OR Early Access OR Proceedings Paper) Refined by: WEB OF SCIENCE CATEGORIES (see appendix) Timespan: 1900-2019. Indexes: SCI-EXPANDED, SSCI, CPCI-S, CPCISSH, BKCI-S, ESCI.	152	53%
IEEE	(bpm OR bpmn) AND (trace OR traceability OR link OR tracing OR tracking OR "answer set" OR association OR "reference set" OR ("source artifact" AND "target artifact")) Filters -Year Range: 2004-2020 - Publication topics: business data processing - business process re-engineering - organizational aspects - data mining - workflow management - software service-oriented architecture - software architecture - Unified Modeling Language - Web services - XML	100	35%
ACM	Abstract: ((bpm OR bpmn) AND (trace OR traceability OR link OR tracing OR tracking OR "answer set" OR association OR "reference set" OR ("source artifact" AND "target artifact"))) ACM Full-Text Collection (602,065 records)	34	12%
TOTAL		**286**	**100%**

It is worth noting that we did not use Springer, because the search is done in all the text (including even footnotes), hence the results are big and very prone to be "false positives". Additionally, the ACM digital Library and Web of Science are indexed by Springer (see the Web of Science list[1] or the Information on Abstracting and Indexing of Springer[2]).

As stated in our RQ1, our research interest focuses on "BPMN" as the de facto standard language for modeling business processes. Therefore, we included the acronym BPM in the query, for obtaining those related papers that do not mention the BPMN language but the BPM discipline in their title or abstract. Regarding "traceability", we looked for

[1] http://mjl.clarivate.com/publist_sciex.pdf

[2] https://www.springer.com/gp/computer-science/lncs/information-on-abstracting-and-indexing/799288

existing publications that define it for obtaining the main terms used by the community when talking about software traceability. These terms will be our keywords in the search string. By examining the publications on software traceability, we found the proceedings document Grand Challenges of Traceability: The Next Ten Years, published in 2017, that summarizes the community efforts around software traceability since the 1990s [15]. This document presents the current work on the area in a series of short position papers. They followed the traceability process axes and fundamentals, previously stated by the community in the book Software and Systems Traceability, published in 2012 [21]; some advances are also presented in Pearson *et al.* [22], Filho and A. Zisman [23].

We examined the first chapter, "Traceability fundamentals," and its complementary "Glossary". The last one defines a total of 168 concepts. Some of them could be represented by a subset of words extracted from the root of the concept or as a common word in a noun phrase, e.g., the *trace* is the root for the concept *traceability,* and the common word for the noun phrases *trace asset* and *trace artifact*. With this analysis, we found the next words: *trace* (129), *tracing* (21), and *link* (5). The number in parenthesis is the frequency of the terms. The rest of the words were *answer set, association, reference set, tracking, source artifact,* and *target artifact*. There was a total of 7 no selected terms because they are used similarly in broader disciplines: artifact, artifact type, attribute, element, requirements management, requirements management tools, and Tracy project. The last one is the name of particular research on traceability. Finally, we found a total of nine keywords representing the software traceability concern.

With the keywords for each concern (*BPMN* and *traceability*), we constructed the following query to be used in the search engines: (*"bpm"* or *"bpmn"*) and (trace OR link OR tracing OR tracking OR "answer set" OR "association" OR "reference set" OR ("source artifact" AND "target artifact"))

3) Conduct Search: The queries were executed on July 6, 2020. Table 1 summarizes the databases consulted, the specific queries, and the number and percentage of papers retrieved. All references and additional information about the search process and its results are on the web page https://olvegam.github.io/bpmn2code/. It is worth noting that at this point in the study, we still have enough articles to apply a snowballing technique, therefore, we proposed to conduct first an abstract review and then, once the relevant papers were obtained, apply the snowballing technique.

4) All papers: The papers obtained after adding up the independent results of each source was a total of 286 papers. However, in a randomized review of the papers, we found several that were not related to the search, even though they fit the search string. For example, Djatna and Ginantaka [24] studies traceability for food safety and quality assurance, using BPMN as a tool for managing traceable units (approaches such as this can also be seen in Pradana *et al* [25]). We called this kind of papers "false positives". For this reason, we decided to make a review of the abstracts that would allow us to exclude those papers that were not related to traceability between BPMN models and source code.

5) Screening: We divided this stage into several reviews to tackle the issue of false positives and then identify target papers. The first author carried up this review, but to reduce bias, this researcher tagged those articles with doubts about Mendeley's REV label. The result was a population of 65 selected papers. It is worth noting that, when searching in the databases, we found that BPM has several meanings. For instance, we obtained the following: Basic Power Management, Beats Per Minute, 3D FD-BPM method for long-wavelength optical communication, Bayesian Probabilistic Matrix Factorization (BPMF), Business Performance Management, building product models (BPMs), Bundle Purchases with Motives, Bit-patterned media recording (BPMR), Balanced Performance Monitor, and Beam Pumping Motor Systems (BPMSs).

As a result of the first screening process, we obtained 65 papers for applying inclusion and exclusion criteria and evaluating their quality. Following the updated protocol for mapping studies on Software Engineering from Petersen *et al.* [12], we applied the next inclusion and exclusion criteria, as follows:

Inclusion Criteria:

- All papers are published in English Language,
- deal with relations among the BPMN model and the software that supports elements of the model,
- deal with processes evolution and traceability,
- use traceability among different components of a business processes management-based implementation, and
- propose some way for improving changes,

Exclusion Criteria:

- Papers are not published in English language,
- have no relation with traceability from BPMN to source code approaches,
- has no traceability either evolution concern,
- uses traceability, but for a different purpose than evolving the software,
- works with BPM process tailoring, but not with its relation to the software that supports the elements of the process model,
- does not work with traceability from BPMN models to source code neither with intermediate artifacts,
- are not capable of answering the research questions, or
- are identical studies (the latest published paper was included)

6) Relevant Papers: The results were: (i) 18 included, (ii) 25 excluded, and (iii) 10 with doubts. The second and third co-authors reviewed the ten articles with doubts, and as a result, was two papers were included and eight excluded. In this review, the agreement coefficient for the two co-authors was 0,782, by calculating this index with the Kappa coefficient of agreement [1]. This is a "substantial" agreement, according to Abraira [26].

Finally, we obtained 25 papers and the second and third author applied snowballing over them. The result was the addition of 2 papers and the exclusion of 3 papers for a total of 24 relevant papers. The TABLE VII Relevant papers, presented as a reference list, contains the references of the accepted papers. Figure 2 summarizes the process followed for selecting papers.

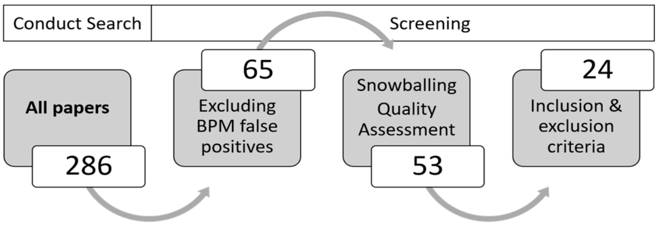

Fig. 2 Processed papers

7) Key-wording using abstract: Once the relevant papers were obtained, the next stage is the data extraction and the presentation of the insights through a classification scheme of the papers. For elaborating the classification scheme, first, we performed a general classification, then a topic-specific classification.

General Classification Scheme: As stated by Petersen *et al.*, we used the most common classification for this kind of studies: (i) venue, (ii) research type, and (iii) research method [12]. Similarly, we used the classification scheme venue proposed by them for (i), the disambiguation table for defining (ii), and their typification for (iii).

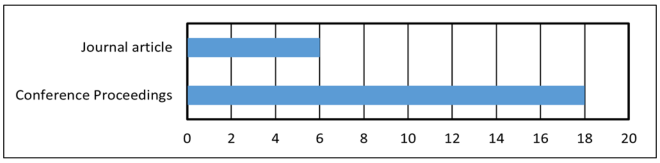

Fig. 3 Venue classification

As for *the* venue, most papers come from conferences with peer-review. Figure 3 presents the distribution. Regarding the research type, the distribution is presented in Figure 4; similarly, and in coherence with the research type, the classification by research method is presented in Figure 5.

The most common type of research is philosophical papers because basically, the papers propose a framework without implementing a solution, an empirical validation, an experience, or applying it in practice. However, the next two types are solution proposals and experience papers that at least present an implementation of an experience. As for the research method, the case study, prototyping, and action research are the most used.

Topic-specific Classification: In this step, Petersen *et al.* [12] used an existing classification scheme as a baseline for comparison purposes of the mapping studies. Therefore, we consulted the SwEBoK chapter for Software Maintenance[27] and used the body of knowledge organized there as a set of initial concepts for our classification. The concepts were: key issues in software maintenance, maintenance processes, techniques for maintenance, and software maintenance tools. Additional categories were produced as needed, sometimes using the nested concepts of the SwEBoK or following the open coding from grounded theory [28], centered in our research questions. Regarding the papers review, in those cases in which the abstracts were not very clear, we reviewed other sections, such as introduction or conclusions. Finally, we obtained an overall structure composed of categories resulting from the papers review's concepts mentioned above (sometimes merged or renamed).

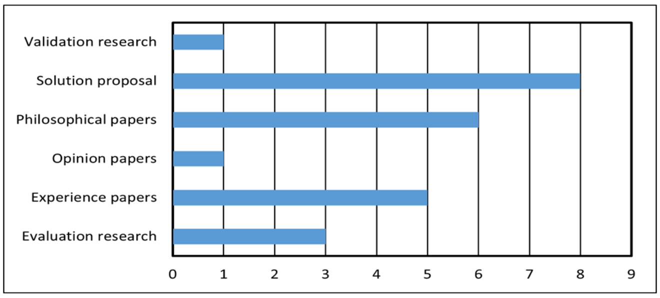

Fig. 4 Research type classification

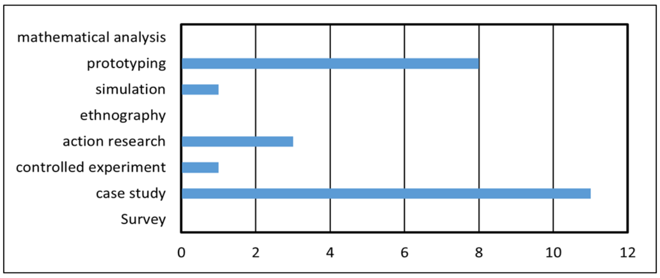

Fig. 5 Research method classification

III. Results and Discussion

Next, we present the analysis of the relevant papers, which was initially guided by the SwEBoK concepts [27] and then by our research questions. After presenting the research findings, we complement this discussion by analyzing threats to validity in this study.

A. SwEBoK Concepts Classification:

The classification with the SwEBoK concepts is presented in Table 2.

TABLE II
SwEBoK concept

SwEBoK concept	# Papers
Key Issues in Software Maintenance	23
Maintenance Process	12
Techniques for Maintenance	7
Software Maintenance Tools	17

It is worth noting that there is a missing paper from the key issues in software maintenance [29] because it was classified under the Software Maintenance Tools category. This classification presents an initial insight on the Software Engineering concepts on which the papers related to traceability from BPMN to source code are working on and those with none or scarce research. Those areas with more research are presented in Table 3 and Table 4.

TABLE III
Main areas of work from key issues in software maintenance

Key issues in software maintenance	18
Limited understanding	12
Testing	0
Impact analysis	5
Maintainability	18

The main key issue is the *maintainability* with half of these works centered on the *limited understanding*. The impact analysis appears with five out of the 24 relevant papers in third place. It is worth noting that none of the papers addresses the concept of testing.

TABLE IV
MAIN AREAS OF WORK IN SOFTWARE MAINTENANCE TOOLS

Software Maintenance Tools	17
Static analyzers	11
Dynamic analyzers	1
Data flow analyzers	3
Cross-referencers	9
Dependency analyzers	8

The next concept, with an important number of papers working on it, is software maintenance tools, in which the works are distributed among static analyzers, cross-references, and dependency analyzers. Finally, there are three papers for data flow analyzers and only one for dynamic analyzers.

A second insight that this initial classification unveils is about those concepts and areas not present in the relevant papers. Regarding concepts, the following are not present:

- Management issues, e.g., alignment with organizational objectives, staffing, process, organizational aspects of maintenance and outsourcing.
- Maintenance cost estimation: cost estimation, parametric models, and experience.
- Software maintenance measurement and their specific measures: analyzability, changeability, stability, testability, size of the software, the complexity of the software, understandability, and maintainability.

Regarding areas, research is scarce or inexistent in works related to traceability from BPMN to source code in the following:

- Maintenance Process: This category includes the processes and the activities. We first found six papers on modification implementation and problem and modification analysis, but less or none work in process implementation, maintenance review/acceptance, migration, and software retirement. In activities, there are only three papers on impact analysis and program understanding without not working on the rest of activities, e.g., modification request acceptance/rejection, maintenance help desk, maintenance Service-Level Agreements. Finally, except for one work on documentation, there are no supporting activities or maintenance planning activities, software configuration management, and software quality.
- Techniques for maintenance: Here only one paper mention program comprehension, 2 Re-engineering and one reverse engineering, without work on migration or retirement.

B. *Data extraction for research questions:*

For answering the research questions - RQ, we designed a data extraction table with these columns:

- Paper Title
- Bibliographic Reference
- Year
- Approach
- Traced Artifacts
- Technologies
- Technique
- Tools
- Evaluation
- Results
- Problems Faced

Then we organized the resulting table for analysis, obtaining the next insights: For the year of publication, as Figure 6 presents, the first reports on the subject appear in 2005, which coincides with community agreements on BPM and the need for a standard language, represented in the official publication of BPMN version 1.0 in 2004.

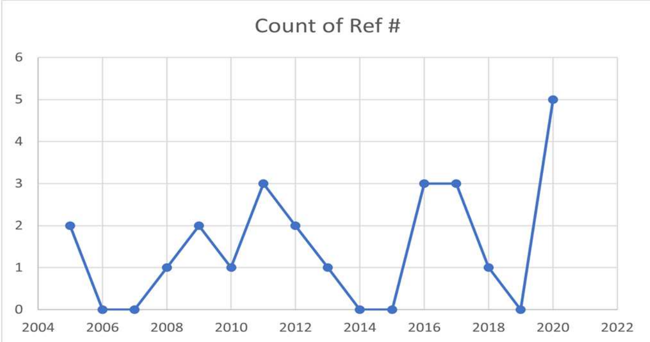

Fig. 6 Publications per year

The following reports are from 2008, with peaks in 2011 and 2017 and periods of no publication in the years 2006-2007 and 2014-2015. The other years exhibit at least one publication on the subject. In 70% of this period, we can say that the topic maintains relevance in the community.

TABLE V
PAPERS BY APPROACH

Approach	Reference Work
Content Management	[31]
SWS	[32] [33]
MDA	[34] [35]
MDE	[30] [36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47]
Reverse engineering	[48]
SOA	[34] [49] [36] [39] [29]
BPMN extension / BPM intervention	[50] [51] [52]
SPL	[42]
SRE	[42]

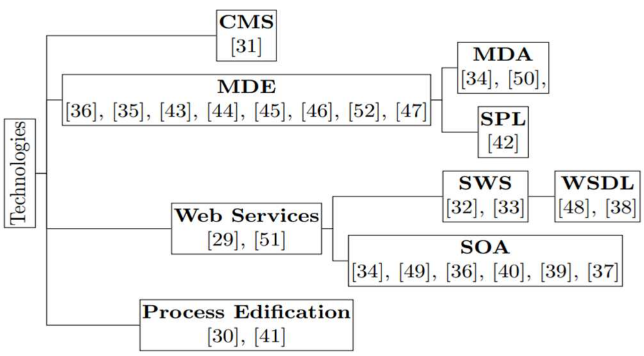

Fig. 7 Technologies

Table 5 presents MDE as the most used approach, followed by SOA. Although we discriminated MDA apart from MDE, they can be grouped in the same approach, obtaining that middle of the work uses the MDE approach. The next approach with an important presence in the SOA - Service Oriented Architecture, explained by how executable BPMN models are connected with external software through web services. Other approaches as Content Management, BPMN Extension, and SRE - Security Requirements Engineering, can be considered not as common because their applications are very specific. Finally, the approach proposed by Yadav *et al* [30] about using graph-based analysis is the most recently proposed, and we realize that it must be considered in future proposals for BP-based software evolution.

Regarding used *technique* in the reviewed papers, in Table 7 we present a classification (technique column). In this classification some papers fit into several groups. The Supra BPMs group refers to platforms for executing the BPMN model, which includes associations of the process with organization or technical information. Kim *et al* [31] integrates users and organization knowledge, [29] associate the concrete level of the BPM platform with the model and [37] that allows to integrate several processing models languages for creating an execution model, which can be understood for several BPMs. The following group of papers proposes a BPMN extension for representing information as business needs, IT hardware, information systems and business rules [32], or for enabling queries about the process that retrieve information on the characteristics of the model [33]. Next, we found those papers with a technique of Architecture extension that integrate the process to the architecture in several levels: at enterprise level [49], or at the architecture views level [50], [34] or integrating two levels, abstract and concrete [29]. In the next classification, Models relations, the majority of papers use the technique of linking already existing models with the process model. This technique includes the most diverse solutions, as can be seen in Table 7; these range from associations at the organization level to source code of legacy systems, passing through architecture levels. The last classification, besides having in common the use of graphs, seeks to facilitate process evolution: In [39] the researchers establish a taxonomy of changes and use a graph to relate the BPMN model with artifacts that allow its execution so that an analysis of the impact and propagation of the change can be made. In Yadav *et al* [30] the proposal is at the level of BPMN models and seeks to manage their versioning, using graphs that connect elements of the different versions of the models.

For *technologies* we represented our classification with the Figure 7. We can see the content management system as an initial proposal for BPMs implementation attempting to manage traceability, MDE and SOA as the main used technologies, and finally, as an emergent technology, we can appreciate the "process edification" proposed by Yadav *et al* [41] and Yadav *et al.* [30]. This last technology is limited to the model process versioning.

For *traced artifacts* we classified them in three categories: *High-level models, low-level models,* and *BPMN models level*, such as presented in Figure 8.

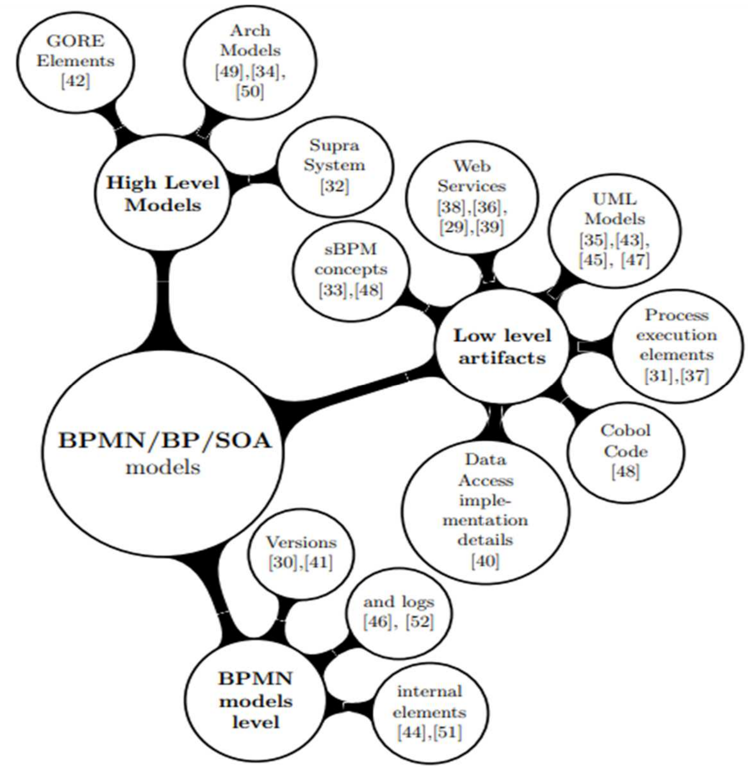

Fig. 8 Traced artifacts

The first category groups paper whose traceability is applied to link high-level models, mainly of architecture, but also of business strategy and organizational elements, with business process models. In this category, we classified five papers. In the next category, we classified papers whose traceability is applied for linking software artifacts more technical than those of the first category. In this group, we classified twelve papers, and the involved artifacts are concepts in SBPM ontology, Web Services, UML models, Process execution elements, Cobol code, and data access implementation details. Finally, in a third category but with only one paper, we classified the paper that proposes managing versioning at the BPMN models level [30].

Regarding tools, Table 6 classifies works according to whether they developed (or not) a tool and, for those that present a tool, presents their development style, i.e., whether it is an extension (or use) of an existing tool or a new development. For each group, we present the names of the tools. We can appreciate that one-third of papers are theoretical proposals without a tool, while the rest involve a development. Of the latter, 30% are new developments, while about 60% are extensions or use of existing tools. This group may represent a trend for evolving tools to incorporate traceability.

C. *Research Findings*

In this study, in addition to answering the questions that motivated its realization, we conceptually locate the works developed in the traceability area from BPMN to source code. Thus, these works, which we relate in the analysis stage with the area of Software Maintenance of the SwEBoK [27], are mainly in improving aspects such as Maintainability, Limited understanding, and Impact Analysis. These works have also contributed to developing software maintenance tools, mainly in static analyzers, dependency analyzers, cross-reference, and data flow analyzers. We located only one work in dynamic analyzers. We found scarce contributions to maintenance issues such as management, cost estimation, software maintenance measurement. When we take a closer look at these concepts, we find that there is scarce work related to the maintenance process and techniques for maintenance. The analysis of the relevant group of papers obtained allows us to answer our research questions as follows:

1) RQ1 What are the existing approaches for managing traceability between BPMN models and source code?: The first proposed approach dates to 2005, by Kim *et al.* [31]; it consists of using a web-based content management system - CMS that allows the execution of processes and, through an extension to the BPMN language allows linking conceptual aspects of the process (that explain the raison d'être of the process) with elements of the process. It is the only approach of its kind because from there on the approaches are developed around Model Driven Engineering (MDE) and Service Oriented Architecture (SOA). They represent two characteristic aspects of software solutions guided by business processes: (i) the need to relate different aspects and software artifacts represented by models (BPMN and UMLs for classes, data, relations, etc.) and (ii) the required integration for the execution of business processes, performed through links to web services. For answering this question, we observed Table 5 and conceptually grouped MDA, MDE, Reverse Engineering and SPL as Model Driven Engineering and considered SWS as part of SOA. Apart from these approaches, there is a recent one, from 2018, by *Yadav* et al [30], which uses graphs for managing versioning; however, as noted in the section 4, this work is a contribution to the versioning of the business process at the model level only.

2) RQ2 For these approaches, what are the used technologies, traced artifacts, proposed techniques, and developed tools?: In coherence with the approaches, the main used technologies are Model Driven Engineering and Web Services; with two different approaches that uses the technology of a CMS [31] and one called edification for managing versioning of BP at the BPMN model level.

In terms of traced artifacts, as detailed in 4, there is only one paper working with source code [48] that applies Reverse Engineering for modeling a legacy system as a business process. The objective is to discover what pieces of the COBOL software must be called from the BPMN model activity. This is the case when Legacy Systems must continue in production, but the organization is applying a BPM strategy that requires integrating the legacy system or some of its functionalities, calling them at the BPMN model execution time.

As for techniques, we would like to highlight the one used in the paper dealing with source code classified in section 4 in the Models relations category [48]. This technique includes parsing COBOL code, syntactic identification, renaming, WSDL construction WSDL to BPMN, and SBPM model generation. Regardless of the source code used to execute the elements of a business process model, this work presents a previous experience when it comes to linking the BPMN model and the source code; therefore, we highlight it as a reference work for traceability from BPMN to source code. Additionally, the classification made for techniques presented us in the category of Graphs another work that we consider of reference [39], which proposes a taxonomy of changes as a basis for the analysis of impact and propagation of changes; these are two fundamentals' activities when dealing with software evolution.

On the tooling side, as shown in section 4, in Table 6, the trend extends existing developments.

TABLE VI
TOOLS

<table>
<tr><th colspan="3">Tools</th><th>Ref</th><th>%</th></tr>
<tr><td colspan="3">No tool</td><td>[32], [49], [35], [33], [42] [51], [45]</td><td>29%</td></tr>
<tr><td rowspan="9">Existing Tool extension or use</td><td colspan="2">ARIS</td><td>[34], [50]</td><td rowspan="9">42%</td></tr>
<tr><td colspan="2">Rational Software Architect</td><td>[36]</td></tr>
<tr><td colspan="2">Petals link (former EBM websourcing)</td><td>[29]</td></tr>
<tr><td rowspan="2">Eclipse</td><td>BPMS intalio</td><td>[40]</td></tr>
<tr><td>Architect plugins</td><td>[39]</td></tr>
<tr><td colspan="2">jBPMN</td><td>[41]</td></tr>
<tr><td colspan="2">Generic Modelling Environment</td><td>[38]</td></tr>
<tr><td colspan="2">Visual Paradigm</td><td>[43]</td></tr>
<tr><td colspan="2">Machine Learning Tools</td><td>[46]</td></tr>
<tr><td rowspan="7">New Development</td><td colspan="2">A content management on web</td><td>[31]</td><td rowspan="7">29%</td></tr>
<tr><td colspan="2">SoftLink</td><td>[48]</td></tr>
<tr><td colspan="2">VerChor Framework</td><td>[44]</td></tr>
<tr><td colspan="2">Empower platform</td><td>[37]</td></tr>
<tr><td colspan="2">BPEM Tool</td><td>[30]</td></tr>
<tr><td colspan="2">Algorithm to calculate route</td><td>[52]</td></tr>
<tr><td colspan="2">BPtraceUCD</td><td>[47]</td></tr>
</table>

TABLE VII
RELEVANT PAPERS

Ref.	Authors	Publication Title	Year
[29]	W. Mu, F. B´enaben, H. Pingaud, N. Boissel-Dallier, J.-P. Lorr´e,	A Model-Driven BPM Approach for SOA Mediation Information System Design in a Collaborative Context	2011
[30]	V. Yadav, R. K. Joshi, S. Ling	A Tool for Traceable Evolution of Process Architectures	2018
[31]	Y. G. Kim, S. C. Park, C. Y. Kim, J. H. Kim	An effective content management methodology for business process management	2005
[32]	M. Hepp, F. Leymann, J. Domingue, A. Wahler, D. Fensel	Semantic business process management: A vision towards using semantic web services for business process management	2005
[33]	W. Abramowicz, A. Filipowska, M. Kaczmarek, T. Kaczmarek	Semantically Enhanced Business Process Modeling Notation	2007
[34]	S. Stein, J. Lauer, K. Ivanov	ARIS Method Extension for Business-Driven SOA	2008
[35]	M. C. Leonardi, M. V. Mauco, L. Felice, G. Montejano, D. Riesco, N. Debnath	Recovering business process diagrams from UML diagrams	2010
[36]	R. Sindhgatta, B. Sengupta	An extensible framework for tracing model evolution in SOA solution design	2009
[37]	J. A. Garcia-Garcia, J. Enriquez, L. Garcia-Borgonon, C. Arevalo, E. Morillo	A MDE-based framework to improve the process management: The EMPOWER project	2017
[38]	A. K. Mandal, A. Sarkar	Service Oriented System design: Domain Specific Model based approach	2016
[39]	M. Bouneffa, A. Ahmad, Change Management of BPM-based Software Applications, in: J. Hammoudi, S and Maciaszek, L and Cordeiro, J and Dietz (Ed.),	Proceedings of the 15th International Conference on Enterprise Information Systems, ESEO Grp; Inst Syst & Technologies Informat, Control & Commun; Assoc Advancement Artificial Intelligence; IEICE Special Interest Grp Software Interprise Modelling; ACM Special Interest Grp Management Informat Syst; ACM Special Interest Grp Comp Human Inte, SciTePress	2013
[40]	C. Mayr, U. Zdun, S. Dustdar	Enhancing traceability of persistent data access flows in process-driven SOAs	2013
[41]	V. Yadav, R. K. Joshi, S. Ling	Process Edification for traceability in evolving architectures	2016
[42]	D. Sprovieri, N. Argyropoulos, C. Souveyet, R. Mazo, H. Mouratidis, A.	Fish, Security Alignment Analysis of Software Product Lines	2017
[43]	Y. Wautelet, S. Poelmans	Aligning the Elements of the RUP/UML Business Use-Case Model and the BPMN Business Process Diagram, in: A. Grunbacher, P and Perini (Ed.)	2017
[44]	M. Gudemann, P. Poizat, G. Salaun, L. Ye, VerChor	A framework for the design and verification of choreographies	2016
[45]	M. Majthoub, Y. Odeh, M. Hijjawi	Non-Functional Requirements Classification for Aligning Business with Information Systems	2020
[46]	H. Al-Ali, A. Cuzzocrea, E. Damiani, R. Mizouni, G. Tello	A composite machine-learning-based framework for supporting low-level event logs to high-level business process model activities mappings enhanced by flexible BPMN model translation	2020
[47]	A. Bouzidi, N. Z. Haddar, M. Ben-Abdallah, K. Haddar	Toward the alignment and traceability between business process and software models	2020
[48]	H. M. Sneed, S. Schedl, S. H. Sneed	Linking legacy services to the business process model	2012
[49]	C.-k. Jung	Actionable Enterprise Architecture	2009
[50]	F. Gao, W. Derguech, M. Zaremba	Extending BPMN 2.0 to Enable Links between Process Models and ARTS Views Modeled with Linked Data	2011
[51]	W. Duangkeaw, T. Suwannasart	Monitoring Call Activity and Service Task Invocations for BPMN	2020
[52]	M. Ramos-Merino, J. M. Santos-Gago, L. M. AlvarezSabucedo	Fuzzy traceability: using domain knowledge information to estimate the followed route of process instances in non-exhaustive monitoring environments	2021

TABLE VIII
TECHNIQUE SUMMARY OF RELEVANT PAPERS

Technique	Ref	Technique detail
BPMs implementation	[31]	A web-based system that allows users to upload their process-related contents and to verify the metadata according to their authority. The systems execute the processes and allows to consult performance data. The bpmn notation is extended for tracing concepts with the elements of the process model.
	[29]	MIS Engineering - Two levels of modeling: Abstract and Concrete. Both used in transformation for executing a Mediation Information System that execute the BPM approach.
	[37]	A platform (named EMPOWER) that provides the application of BPM using MDE. It allows business process in several process modeling language and with transformations generates the process execution, allows the instances triggering and calculates the process performance metrics.
BPMN extension	[32]	SBPM - Semantic Business Process Management for adding knowledge to the processes: Business needs (as goals), IT hardware, information systems, business rules. Additionally, it provides query language
	[33]	Sbpmn ontology for executing queries about processes, retrieving information on what are the elements of a process, what are the sequence flow connection rules, etc.

Models relations	[49]	Enterprise Architecture to Enterprise Solution documentation for linking conceptual artifacts to physical level artifacts.
	[36]	Specifying Changes (Define changes for a meta-model, define relationships among them, relate changes with dependent models) and Managing Changes (in SOA solution designs)
	[35]	Heuristics definitions - Retrieval of business process diagrams from UML diagrams through the definition of heuristics.
	[29]	MIS Engineering - Two levels of modeling: Abstract and Concrete. Both used in transformation for executing a Mediation Information System that execute the BPM approach.
	[40]	Use a View-based Modeling Framework (VbMF) for integrating persistence data flows.
	[48]	Parsing COBOL code, syntactic identification, renaming, WSDL construction, WSDL to BPM and S-BPM model generation.
	[38]	The domain level concepts of SOS are defined from both business process and service representation perspectives. A set of traceability rules are devised to draw the correspondence between the business process and service domain concepts of SOS
	[44]	An intermediate format for describing choreographies. A set of properties to be respected for assuring the correctness of the system under development. Using model and equivalence checking techniques, via an encoding into process algebra, these properties are verified. This is for asynchronous communication semantics, that is, peers involved in the distributed version of the system exchange messages via FIFO buffers.
	[43]	Rules for translating BUCM elements into BPMN BPD elements.
	[42]	On the GORE model apply transformation rules to generate a hybrid reference process model, then instantiate different secure business process designs that contain security implementing activities
	[46]	Two main phases: clustering-based labeling approach and supervised ML-based classification. In the first phase it is used a flexible BPMN model translation methodology
	[51]	(i) Import and extract data, (ii) Analyze constraints of variables, (iii) Create and insert a Java Listener class name into XML format, (iv) Test BPMN with test cases (v) Create new test cases (vi) Generate test reports
	[45]	(I) Integrate the BPMN with its NFR (II) Convert the BPMN into Use Case Diagram and Map the Quality Requirements into its corresponding descriptions
	[52]	Using BPMN-E2 extension, it is proposed an algorithm offer a solution to the non-exhaustive traceability, the so-called fuzzy traceability. In this way, starting from the event log content, an estimation of the most probable route is made.
	[47]	An MDA compliant-approach called Business Process to-trace Use case model and Class Diagram (BPtraceUCD). According to the abstraction levels of MDA, the approach is a CIM to PIM one.
Graphs	[39]	A BPM meta-model as a BPM artifacts repository data schema. A BPM change operations taxonomy. Formalization of the change and the analysis of its impact propagation by graph rewriting rules.
	[30]	A process is decomposed in its elements (activities, gateways and flows), then changes are applied composing them by adding, deleting or changing their order. These compositions are recorded with graph techniques that will be used then for reasoning though the modifications, saving traceability among process versions.

With the pre-existence of legacy systems, the need for flexible integration, and the permanent creation of new technologies and trends, it is difficult to think that a new development that contemplates everything required for a business process management system in the current era can be made. Therefore, we consider that important research finding related to tools is the necessary use of existing ones contributing with extensions to solve emerging needs.

3) RQ3 How have the approaches been evaluated, and what have been the results? Although only one work corresponds strictly to evaluation research, more than half of papers present solutions (four papers) and report experiences (five papers) that contribute to overcoming needs in terms of traceability starting from BPMN models. Solutions papers and experiences reports evaluate their results by presenting the improvements to the gaps that motivated the paper in the specific context where the paper was developed. The main results can be summarized as contributions in (i) the execution of business processes [31], [37], (ii) relations between architectural elements and the business processes [49], [50], (iii) change management in process models [36], [39], [42], [30], and (iv) relations between business process and web services [38]. Our answer to the question is that the works are equally divided into theoretical and practical proposals, the latter being evaluated through results that are limited to an evaluation, mainly through case studies, specifically in the context in which the work is developed. Finally, related to our focus on traceability from BPMN models to source code, we have found the contribution of [38] as a reference for linking elements of the business process with the web services that are used for implementing the execution of the business process.

4) RQ4 What are the current problems faced on traceability between BPMN and source code?: There are significant contributions around traceability between different concerns in the organization or levels of technology management and business process models; they cover links from organizational subjects to business processes, architectural concerns, links between different software artifacts and business process models elements, e.g., between UML activities diagrams and activities in a BPMN model. In this study, we found an attempt for linking source code, in the COBOL language, with business process models in [48] and several proposals for linking the business processes with the required web services for executing them [39], [38], [40], [36], [32], [39], [38]. However, specifically for traceability between BPMN and source code, the research is scarce. This is a gap to be analyzed and with future research contributing to help organizations in reacting to continuous pressure for digital adaptation and business process evolution.

D. *Threats to Validity*

In this section, we identify the threats to validity following the description made by Rahimi and Cleland-Huang [17] to *Study selection validity, Data validity* and *Research validity*.

1) Study selection validity: the study selection validity groups threats in the search process or the study filtering phase. We were careful in the search process to mitigate these threats: (i) Digital Library - DL selection: we used widely recognized digital databases, whose publications include double-blind peer review. Additionally, we verified that these databases published computing, information technology, and software engineering full-text articles. (ii) We constructed a search string based on a detailed definition of terms related to our concern of "traceability from BPMN to source code". This construction was presented in section 3, subsection 3.2.2; once the search string was stated, we converted it to the specific syntax of each DL and presented them summarized in Table 1. This table also presents the numbers of papers obtained with the search. (iii) Finally, in section 3.3 we detailed additional parameters for the search process, e.g., year of search. We mitigated the possible threats regarding the study filtering phase by applying several filters phases that we detailed in section 3.5. In this phase of the study, we faced time limitations for co-authors applying for the same review as the first author. However, we applied a strategy of reviewing the papers that caused doubts when reviewed by the first author. This process was supported using the public groups organization capability available in Mendeley[3] . In the opinion of the first author, each paper that required revision was labeled in the Notes field for being identified by the co-authors.

2) The Data validity: this category described by Rahimi and Cleland-Huang [17] refers to threats in the sample size and the data extraction and analysis phases of the study. We recognize that although our initial sample of papers was sufficiently large (286), the filtering and quality assessment process left us with 24 papers. This number falls within the range of "*Good quality*" (20-40) proposed by Kitchenham and Charters [11] for quality evaluation. We try to overcome this situation of not being in the "*Excellent quality*" category by being highly conservative in our statements because of the analyses. We are aware, and this is expressed in the study, of the impossibility of generalizing. It is necessary to extend this study to include, gray literature. For mitigating threats in the data extraction, we used two classifications, the first one following the recommendations of Kitchenham and Brereton [9], Minhas *et al* [14] for using previous classifications. Therefore, we consulted the Software Maintenance chapter from the SwEBoK [27] and classified the relevant papers. The second data extraction table was constructed using the subjects of our research questions, depicted in section 4.2. The data extraction in this phase included examining all the sections of the relevant papers.

3) As for research validity: it considers all the research design, we mitigated possible threats following the proposed protocol by Kitchenham, updated for Petersen, and detailed in section 3 [9], [14]. We were careful to detail how the search was performed, the databases consulted, the search string, and the parameters required in each engine to ensure repeatability. Regarding coverage of research questions, we detailed each subject and developed each one in sections 4 and 5 of Results & Discussions and Research Findings. Regarding the research field, we developed the background in section 2, including related work.

[3] Reference manager - https://www.mendeley.com/search/

IV. CONCLUSION

Traceability in software is an emerging and evolving discipline; for the specific case in which software development is guided by business processes - BP, it evolves as BP approaches, techniques and technologies evolve. Pressure on enterprises for continuous evolution to digital solutions, increased since 2020 by the COVID-19 pandemic, requires organizations to develop agility in software evolution. Traceability is a lever that favors such agility by allowing more quickly and at lower cost to know the artifacts involved in a change and even to be able to propagate the changes through them.

In this study on the traceability from BPMN models to code source, we found initial proposals for executing the BPMN models integrating different knowledge aspects of the organizations, such as strategy, roles, architectural concerns, and technological issues. In the same way, the initial proposals sought to provide ways to measure performance in these business processes.

In the studies we analyzed, we found great diversity in approaches, techniques, technologies, tools, and, of course, in the set of artifacts on which traceability is applied. Despite this diversity, we found in common the development of approaches based on MDE, the incorporation of SOA as a technology that provides a solution to the integration of BPMN models with the software required for their execution, and emerging proposals to use traceability in the versioning of business process models.

With the caution required by the limitations of this study, we dare to conclude that traceability from BPMN models to source code is an emerging area, leveraged in MDE, in previous developments of tools available for modeling and execution of business processes, using web services as a link between BPMN models and source code and with necessary versioning of business process models. It is important to mention that traceability between business processes and security will also be subject to evolution in the emerging proposals in this area. Finally, as observed in the first classification, guided by the concepts established by the SwEBoK community, aspects such as cost estimation in changes and traceability in tests are issues with scarce or no development.

REFERENCES

[1] A. M. Ubaid and F. T. Dweiri, "Business process management (BPM): terminologies and methodologies unified," *Int. J. Syst. Assur. Eng. Manag.*, vol. 11, no. 6, pp. 1046–1064, 2020, doi: 10.1007/s13198-020-00959-y.

[2] C. F. H. Dos Santos, M. Fantinato, J. P. M. de Oliveira, and L. H. Thom, "Evaluating support for implementing BPMN 2.0 elements in business process management systems," in *ICEIS 2020 - Proceedings of the 22nd International Conference on Enterprise Information Systems*,

2020, vol. 2, pp. 193–200.
[3] P. Harmon and J. Garcia, “The State of Business Process Management 2020,” *BPTrends*, 2020.
[4] A. Sahay, A. Indamutsa, D. Di Ruscio, and A. Pierantonio, “Supporting the understanding and comparison of low-code development platforms,” in *Proceedings - 46th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2020*, 2020, pp. 171–178, doi: 10.1109/SEAA51224.2020.00036.
[5] A. M. Gutiérrez Fernández *et al.*, “Applying Business Architecture Principles with Domain-Specific Ontology for ACM Modelling: A Building Construction Project Example,” *Lect. Notes Bus. Inf. Process.*, vol. 362 LNBIP, pp. 388–399, 2019, doi: 10.1007/978-3-030-37453-2_32.
[6] A. Koschmider, “Microservices-based business process model execution,” in *CEUR Workshop Proceedings*, 2017, vol. 1859, pp. 158–161.
[7] H. Kaiya *et al.*, “Preliminary Systematic Literature Review of Software and Systems Traceability,” *Procedia Comput. Sci.*, vol. 112, pp. 1141–1150, 2017, doi: https://doi.org/10.1016/j.procs.2017.08.152.
[8] V. Yadav, R. K. Joshi, and S. Ling, “Evolution traceability roadmap for business processes,” in *ACM International Conference Proceeding Series*, 2019, pp. 1–5, doi: 10.1145/3299771.3299790.
[9] B. Kitchenham and P. Brereton, “A systematic review of systematic review process research in software engineering,” *Inf. Softw. Technol.*, vol. 55, no. 12, pp. 2049–2075, 2013, doi: 10.1016/j.infsof.2013.07.010.
[10] C. Marshall, B. Kitchenham, and P. Brereton, “Tool features to support systematic reviews in software engineering - A cross domain study,” *E-Informatica Softw. Eng. J.*, vol. 12, no. 1, pp. 79–115, 2018, doi: 10.5277/e-Inf180104.
[11] B. Kitchenham, L. Madeyski, and P. Brereton, “Meta-analysis for families of experiments in software engineering: a systematic review and reproducibility and validity assessment,” *Empir. Softw. Eng.*, vol. 25, no. 1, pp. 353–401, 2020, doi: 10.1007/s10664-019-09747-0.
[12] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, “Systematic Mapping Studies in Software Engineering,” 2008.
[13] J. S. Molléri, K. Petersen, and E. Mendes, “CERSE - Catalog for empirical research in software engineering: A Systematic mapping study,” *Inf. Softw. Technol.*, vol. 105, pp. 117–149, 2019, doi: 10.1016/j.infsof.2018.08.008.
[14] N. M. Minhas, S. Masood, K. Petersen, and A. Nadeem, “A systematic mapping of test case generation techniques using UML interaction diagrams,” *J. Softw. Evol. Process*, vol. 32, no. 6, 2020, doi: 10.1002/smr.2235.
[15] G. Antoniol, J. Cleland-Huang, J. H. Hayes, and M. Vierhauser, “Grand Challenges of Traceability: The Next Ten Years,” *CoRR*, vol. abs/1710.0, 2017.
[16] D. Falessi, J. Roll, J. L. C. Guo, and J. Cleland-Huang, “Leveraging historical associations between requirements and source code to identify impacted classes,” *IEEE Trans. Softw. Eng.*, vol. 46, no. 4, pp. 420–441, 2020, doi: 10.1109/TSE.2018.2861735.
[17] M. Rahimi and J. Cleland-Huang, “Evolving software trace links between requirements and source code,” in *Proceedings - 2019 IEEE/ACM 10th International Workshop on Software and Systems Traceability, SST 2019*, 2019, p. 12, doi: 10.1109/SST.2019.00012.
[18] Trisotech, “BPMN Introduction and History.” 2015.
[19] T. Tambo, *Business process management notation (BPMN) as a Lingua Franca in virtual communities*, vol. 11. 2017.
[20] M. M. Lankhorst, M.-E. Iacob, and H. Jonkers, “State of the art,” *Enterp. Eng. Ser.*, pp. 11–40, 2017, doi: 10.1007/978-3-662-53933-0_2.
[21] J. Cleland-Huang, A. Zisman, and O. Gotel, *Software and systems traceability*, vol. 9781447122. Springer-Verlag London Ltd, 2012.
[22] S. Pearson *et al.*, “Are Distributed Ledger Technologies the panacea for food traceability?,” *Glob. Food Sec.*, vol. 20, pp. 145–149, 2019, doi: 10.1016/j.gfs.2019.02.002.
[23] G. A. D. A. C. Filho and A. Zisman, “D3TraceView: A traceability visualization tool,” in *Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE*, 2017, pp. 590–595, doi: 10.18293/SEKE2017-038.
[24] T. Djatna and A. Ginantaka, “An Analysis and Design of Frozen Shrimp Traceability System Based on Digital Business Ecosystem,” in *2014 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2014, pp. 157–163.
[25] I. G. M. T. Pradana, T. Djatna, and I. Hermadi, “Blockchain Modeling for Traceability Information System in Supply Chain of Coffee Agroindustry,” in *ICACSIS 2020: 2020 12th International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2020, pp. 217–223.
[26] V. Abraira, “El índice kappa,” *Semer. - Med. Fam.*, vol. 27, no. 5, pp. 247–249, Jan. 2001, doi: 10.1016/s1138-3593(01)73955-x.
[27] P. A. Quezada-Sarmiento, J. Garbajosa, H. Washizaki, and L. Enciso, “Knowledge description model for bodies of knowledge in software engineering context,” 2017, doi: 10.23919/CISTI.2017.7976058.
[28] H. Coolican, *Research methods and statistics in psychology*. 2017.
[29] W. Mu, F. B´enaben, H. Pingaud, N. Boissel-Dallier, J.-P. Lorr´e, A Model-Driven BPM Approach for SOA Mediation Information System Design in a Collaborative Context. IEEE International Conference on Services Computing, IEEE, 2011, pp. 747–748. doi:10.1109/SCC. 2011.66. URL http://ieeexplore.ieee.org/document/6009339/
[30] V. Yadav, R. K. Joshi, S. Ling, A Tool for Traceable Evolution of Process Architectures. IEEE International Conference on Software Architecture Companion (ICSA-C), IEEE, 2018, pp. 101–106. doi:10.1109/ICSA-C.2018.00035. URL https://ieeexplore.ieee.org/document/ 8432190/
[31] Y. G. Kim, S. C. Park, C. Y. Kim, J. H. Kim, An effective content management methodology for business process management, in: F. VanderAalst, WMP and Benatallah, B and Casati, F and Curbera (Ed.), Business Process Management, Proceedings, Vol. 3649 of Lecture Notes in Computer Science, LORIA, Springerverlag Berlin, Heidelberger PLATZ 3, D14197 Berlin, Germany, 2005, pp. 416–421.
[32] M. Hepp, F. Leymann, J. Domingue, A. Wahler, D. Fensel, Semantic business process management: A vision towards using semantic web services for business process management, Proceedings - ICEBE 2005: IEEE International Conference on e-Business Engineering 2005 (1) (2005) 535–540. doi:10.1109/ICEBE.2005.110.
[33] W. Abramowicz, A. Filipowska, M. Kaczmarek, T. Kaczmarek, Semantically Enhanced Business Process Modeling Notation. Semantic Business Process and Product Lifecycle Management, CEUR Workshop Proceedings, 2007. URL http://ceur-ws.org/Vol-251/paper11.pdf
[34] S. Stein, J. Lauer, K. Ivanov, ARIS Method Extension for Business-Driven SOA, Wirtschaftsinformatik 50 (6) (2008) 436–444. doi:10.1365/s11576-008-0090-5. URL http://link.springer.com/10.1365/ s11576-008-0090-5.
[35] M. C. Leonardi, M. V. Mauco, L. Felice, G. Montejano, D. Riesco, N. Debnath, Recovering business process diagrams from UML diagrams, 2010 ACS/IEEE International Conference on Computer Systems and Applications, AICCSA 2010doi:10.1109/AICCSA.2010.5586982.
[36] R. Sindhgatta, B. Sengupta, An extensible framework for tracing model evolution in SOA solution design, Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications, OOPSLA (2009) 647–658doi:10.1145/1639950.1639960.
[37] J. A. Garcia-Garcia, J. Enriquez, L. Garcia-Borgonon, C. Arevalo, E. Morillo, A MDE-based framework to improve the process management: The EMPOWER project. IEEE 15th International Conference on Industrial Informatics (INDIN), Vol. 2, IEEE, 2017, pp. 553– 558. doi:10.1109/INDIN.2017.8104832. URL http://ieeexplore.ieee.org/document/8104832/
[38] A. K. Mandal, A. Sarkar, Service Oriented System design: Domain Specific Model based approach, 2016 3rd International Conference on Computer and Information Sciences, ICCOINS 2016 - Proceedings (2016) 489–494doi: 10.1109/ICCOINS.2016.7783264.
[39] M. Bouneffa, A. Ahmad, Change Management of BPM-based Software Applications, in: J. Hammoudi, S and Maciaszek, L and Cordeiro, J and Dietz (Ed.), Proceedings of the 15th International Conference on Enterprise Information Systems, ESEO Grp; Inst Syst & Technologies Informat, Control & Commun; Assoc Advancement Artificial Intelligence; IEICE Special Interest Grp Software Interprise Modelling; ACM Special Interest Grp Management Informat Syst; ACM Special Interest Grp Comp Human Inte, SciTePress - Science and and Technology Publications, AV D Manuell, 27A 2 ESQ, SETUBAL, 2910-595, Portugal, 2013, pp. 37–45. doi:10.5220/0004441400370045. URL http://www.scitepress.org/DigitalLibrary/ Link.aspx?doi=10.5220/0004441400370045.
[40] C. Mayr, U. Zdun, S. Dustdar, Enhancing traceability of persistent data access flows in process-driven SOAs, Distributed and Parallel Databases 31 (1) (2013) 1–45. http://dx.doi.org/10.1007/s10619-012-7102-6 URL http://link.springer.com/10.1007/ s10619-012-7102-6.
[41] V. Yadav, R. K. Joshi, S. Ling, Process Edification for traceability in evolving architectures, ACM International Conference Proceeding

Series 18-20-Febr (2016) 99–108. doi:10.1145/2856636.2856650.

[42] D. Sprovieri, N. Argyropoulos, C. Souveyet, R. Mazo, H. Mouratidis, A. Fish, Security Alignment Analysis of Software Product Lines, Proceedings - 4th International Conference on Enterprise Systems: Advances in Enterprise Systems, ES 2016 (2017) 97–103doi:10.1109/ES. 2016.19.

[43] Y. Wautelet, S. Poelmans, Aligning the Elements of the RUP/UML Business Use-Case Model and the BPMN Business Process Diagram, in: A. Grunbacher, P and Perini (Ed.), Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 10153 LNCS of Lecture Notes in Computer Science, SPRINGER International Publishing AG, Gewerbestrasse 11, Cham, CH-6330, Switzerland, 2017, pp. 22–30. doi:10.1007/978-3-319-54045-0_2. URL http://link.springer.com/10.1007/ 978-3-319-54045-0{_}2

[44] M. Gudemann, P. Poizat, G. Salaun, L. Ye, VerChor: A framework for the design and verification of choreographies, IEEE Transactions on Services Computing 9 (4) (2016) 647–660. doi:10.1109/TSC.2015.2413401.

[45] M. Majthoub, Y. Odeh, M. Hijjawi, Non-Functional Requirements Classification for Aligning Business with Information Systems, in: Proceedings of the 2020 International Conference on Big Data in Management, ICBDM 2020, Association for Computing Machinery, New York, NY, USA, 2020, pp. 84–89. doi:10.1145/3437075.3437091. URL https://doi-org.ezproxyegre.uniandes.edu.co: 8843/10.1145/3437075.3437091.

[46] H. Al-Ali, A. Cuzzocrea, E. Damiani, R. Mizouni, G. Tello, A composite machine-learning-based framework for supporting low-level event logs to high-level business process model activities mappings enhanced by flexible BPMN model translation, Soft Computing 24 (10) (2020) 7557–7578. doi:10.1007/s00500-019-04385-6.

[47] A. Bouzidi, N. Z. Haddar, M. Ben-Abdallah, K. Haddar, Toward the alignment and traceability between business process and software models. Proceedings of the 22nd International Conference on Enterprise Information Systems, Vol. 2, 2020, pp. 701–708. doi:10.5220/0009004607010708. URL https://www.scitepress.org/Papers/2020/ 90046/90046.pdf

[48] H. M. Sneed, S. Schedl, S. H. Sneed, Linking legacy services to the business process model, 2012 IEEE 6th International Workshop on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems, MESOCA 2012 (2012) 17–26doi:10.1109/Mesoca.2012.6392601.

[49] C.-k. Jung, Actionable Enterprise Architecture, in: 2009 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing, IEEE, 2009, pp. 294–299. doi:10.1109/SNPD.2009.106. URL http://ieeexplore.ieee.org/document/5286652/.

[50] F. Gao, W. Derguech, M. Zaremba, Extending BPMN 2.0 to Enable Links between Process Models and ARTS Views Modeled with Linked Data, in: K. Abramowicz, W and Maciaszek, L and Wecel (Ed.), Business Information Systems Workshops (BIS 2011), Vol. 97 of Lecture Notes in Business Information Processing, Springer-Verlag Berlin, Heidelberger PLATZ 3, D-14197 Berlin, Germany, 2011, pp. 41+. doi:10.1007/978-3-642-25370-6_5.

[51] W. Duangkeaw, T. Suwannasart, Monitoring Call Activity and Service Task Invocations for BPMN, in: Proceedings of the 2020 Asia Service Sciences and Software Engineering Conference, ASSE ’20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 151–155. doi:10.1145/3399871.3399884. URL https://doi-org.ezproxyegre.uniandes.edu.co: 8843/10.1145/3399871.3399884.

[52] M. Ramos-Merino, J. M. Santos-Gago, L. M. AlvarezSabucedo, Fuzzy traceability: using domain knowledge information to estimate the followed route of process instances in non-exhaustive monitoring environments, Journal of Intelligent Manufacturing. doi:10.1007/s10845-020-01636-4.