Information Systems 66 (2017) 43–58

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/is

A systematic literature review on the architecture of business process management systems

Shaya Pourmirza*, Sander Peters, Remco Dijkman, Paul Grefen

Eindhoven University of Technology, PO Box 513, 5600 MB Eindhoven, The Netherlands

ARTICLE INFO

Article history:
Received 11 January 2017
Revised 30 January 2017
Accepted 31 January 2017
Available online 1 February 2017

Keywords:
Business process management system
Workflow management system
Information systems architecture
Systematic literature review

ABSTRACT

Due to the high complexity of modern-day business, organizations are forced to quickly adapt to a wide range of cutting-edge developments. These developments influence the structure and behavior of the business processes that represent the work and of the Business Process Management Systems (BPMS) that support them. Consequently, the architecture of BPMS has changed a lot over the past two decades. However, there is no systematic overview of the research done in this area since the Workflow reference model first set the standard for BPMS architecture in 1995. To bridge this gap, this paper presents a Systematic Literature Review (SLR) of BPMS architectures, by analyzing 41 primary studies taken from a gross collection of 608 research papers. The BPMS architectures that served as primary studies were compared with respect to the reference architecture that they are based on, the level of elaboration at which they are described, the architectural styles that they use, the means with which they are evaluated, and the functionality that they support. The resulting comparison provides an overview of and insights into the current body of knowledge on BPMS architectures.

1. Introduction

Business Process Management Systems (BPMS) are information systems that interpret business processes to ensure that the activities specified therein are properly executed and monitored by an organization [1]. Such systems have seen significant industrial adoption and, therefore, their architectures are rapidly evolving in order to fulfill ever-expanding business requirements. Consequently, the architecture design of BPMSs has become an important development activity in the research community [2–12].

The study of existing architectures of BPMSs can provide a useful account of how such systems should be structured in order to support the intended functionalities. Therefore, this paper provides a comprehensive overview of the state-of-the-art by surveying existing Business Process Management System(BPMS) architectures and systematically identifying, classifying and analyzing them. For this purpose, a Systematic Literature Review(SLR methodology was used, because that provides a means of identifying, interpreting and evaluating the existing body of knowledge in a specific research discipline [13,14]. In particular, since we seek to provide insight into existing BPMS architectures, our study is considered a mapping study (a.k.a. scoping study) [15]. As such, this SLR contributes to research in the area by providing a structured and comprehensive overview of available BPMSs architectures and by identifying future research opportunities.

Against this background, the remainder of this paper is organized into four sections as follows. Firstly, Section 2 presents the review protocol that was employed as a basis for conducting our survey. Secondly, Section 3 discusses the evaluation methodology that was used for classifying and analyzing the selected studies. Subsequently, Section 4 reports on the obtained results. Then, Section 5 presents some possible research directions and, Section 6 concludes the paper.

2. Review protocol

The review protocol, used to conduct our SLR study, specifies the *research questions* (Section 2.1) as well as the *search protocol* (Section 2.2) and the *selection criteria* (Section 2.3), which were employed to select relevant primary studies.

In order to ensure the quality of the study, the guidelines proposed in [13,15–17] were followed. Accordingly, the involved researchers were organized into two groups, namely a *review team* and an *evaluation team*. The review team, which consisted of two

* Corresponding author.
E-mail addresses: s.pourmirza@tue.nl, shoopi@gmail.com (S. Pourmirza), s.p.f.peters@tue.nl (S. Peters), r.m.dijkman@tue.nl (R. Dijkman), p.w.p.j.grefen@tue.nl (P. Grefen).

http://dx.doi.org/10.1016/j.is.2017.01.007
0306-4379/© 2017 Elsevier Ltd. All rights reserved.

researchers in the domain of Business Process Management(BPM), was responsible for:

- formulating the research questions,
- developing the review protocol,
- searching and selecting the primary studies,
- developing a classification framework,
- extracting data from the selected primary studies, and
- synthesizing and reporting the outcomes of the review.

The evaluation team, which consisted of two researchers in the domain of BPM and information system architecture, was responsible for:

- evaluating the research questions,
- evaluating the review protocol,
- evaluating the final list of the selected primary studies,
- evaluating the final classification framework, and
- evaluating the final content of this research.

2.1. Research questions

This SLR study set out to acquire knowledge about existing BPMS architectures within the research communities. This goal can be achieved by answering the following central research question (RQ).

RQ Which relevant primary studies were published in the area of BPMS architecture?

In order to properly assess the relevance of primary studies, we decompose the central research question into five sub-questions. In particular, these sub-questions investigate the design, evaluation and provided functionalities of the architectures in the selected primary studies.

The first research sub-question seeks to find the foundation (i.e., *where* is the starting point) for the design and development of the identified architectures in the primary studies. Thus, *RQ1* has been formulated as follows:

RQ1 To what extent were the architectures in the primary studies built upon existing (reference) architectures?

The second and third research sub-questions are used to analyze the structure of the identified architectures in the primary studies (i.e., *how* these architectures have been presented). To this end, *RQ2* examines the level of detail that has been provided by the identified architectures in the primary studies. Thus, this research sub-question has been formulated as follows:

RQ2 To what extent were the architectures in the primary studies elaborated upon in terms of details and technologies?

RQ3 explores the high-level decision decisions that have been made to describe the overall structure (i.e., the architectural style) of the identified architectures in the primary studies. Thus, this research sub-question has been formulated as follows:

RQ3 Which architectural styles have been followed by the architectures in the primary studies ?

The fourth research sub-question focuses on how the identified architectures in the primary studies have been evaluated. Thus, *RQ4* has been formulated as follows:

RQ4 How were the architectures in the primary studies evaluated?

Finally, the fifth research sub-question considers the functionalities that are addressed by the identified architectures in the primary studies. Therefore, *RQ5* has been formulated as follows:

RQ5 Which main functionalities have been addressed in the primary publications?

2.2. Search strategy

The main strategy employed in our SLR study was to find *as many* scientific publications as possible and, subsequently, the results were narrowed down by applying predefined criteria. In this section, the search strategy, used to identify the preliminary set of primary studies, is discussed. We, firstly, provide a set of search strings in Section 2.2.1, and, then, we present the search sources (i.e., on-line databases) that were employed to conduct the search in Section 2.2.2.

2.2.1. Search strings

The first action in the search strategy was formulating a set of search strings. In order to develop the search strings we followed the guidelines suggested by Kitchenham et al. [17] and, consequently:

(i) the terms "BPMS" and "architecture" were derived from the research questions as the main search terms in this study;
(ii) "Business Process Management System", "workflow management system", "orchestration execution system" "choreography execution system" were utilized as alternative terms (i.e., alternative spelling or technical synonyms) for "BPMS";
(iii) the Boolean AND was used to connect the search terms identified in step (i) in order to narrow down the search results (e.g., we employed *"BPMS" AND "architecture"* as a group of search strings in our study);
(iv) the Boolean OR was used to incorporate alternative terms in step (ii) in order to provide a wider range of search results (e.g., *"BPMS" OR "Business Process Management Systems"* was employed as a part of the search strings in this study);

All the mentioned alternative terms in step (ii), in constructing the final search strings, were shortened in order to retrieve *as many results as possible*. For example, the term, "system" was removed from the end of the alternatives. The term, "workflow" was used instead of "workflow (management) system" since it has been used to refer to the same concept in the literature, whereas "business process" and "Business Process (management) System" refer to different concepts (i.e., business process, usually, has been used to refer to a business process model and not business process system). Where complex Boolean search strings were not supported by a database, a designated search string was used for that database. These guidelines, thus, led to the following search strings:

ST1 ("architecture" *AND* "bpm") *OR*
ST2 ("architecture" *AND* "business process management") *OR*
ST3 ("architecture" *AND* "workflow") *OR*
ST4 ("architecture" *AND* "orchestration") *OR*
ST5 ("architecture" *AND* "choreography")

It should be noted that a synonym for the term, "architecture" (i.e., *system structure*) was not considered since no results were found for the constructed search strings with the term, "system structure" instead of "architecture" (e.g., "system structure" *AND* "bpm").

2.2.2. Search sources

The second action in the search strategy was choosing the search sources. This action allows other researchers to obtain the same search outcome as that which we gathered from the mentioned search strings. Based on [18] and [19], scientific search engines and indexing systems in the field of computer science were used as preliminary sources. Table 1 shows the databases that were considered in the search strategy.

Table 1
Utilized electronic databases.

Database	Institution	Abbr.
ACM Digital Library	ACM	ACM
IEEE Xplore	IEEE	IEEE
SciVerse Scopus	Elsevier	ELSV
SpringerLink	Springer	SPRG
Web of science	Thomson Reuters	ISI

These databases were chosen since they reasonably cover most of the scientific publications (e.g., journal papers, conference proceedings and workshop papers) in the field of computer science. As also suggested by [20], these databases guarantee to provide the confidence level for inclusion of all the required primary studies.

2.3. Selection criteria

The aim of the selection criteria was to identify the relevant primary studies that adequately provided answers to the research questions. Therefore, according to these criteria, the obtained results were narrowed down by excluding the studies that were not relevant to the research questions. As suggested in [21], the selection criteria consisted of a set of inclusion criteria and a set of exclusion criteria. The inclusion criteria were further classified into three classes:

(i) *assess-ability criteria*, which measured whether a primary study was available for further assessments;
(ii) *paper specification criteria*, which measured whether the meta-information of a primary study (i.e., title, authors, journal or conference, publication year and citation) were satisfactory; and
(iii) *content criteria*, which measured whether the content of a primary study addressed the research questions.

The complete set of criteria determining the inclusion or exclusion of primary studies are presented in Table 2.

IC-A1 and *IC-A2* are the primary requirements. *IC-A1* only considers primary studies that include on-line accessible full-text versions and *IC-A2* only accepts publications that have been fully written in English. The next four inclusion criteria (*IC-P1* to *IC-P4*) consider the meta-information concerning a study. *IC-P1* simply accepts publications that fully contain one group of search strings in their titles. Contrary to expectations, two publications appeared in the search results from the Web of Science, while their titles did not fully meet the search strings' requirements and hence *IC-P1* was employed to remove them. *IC-P2* only allows primary studies that have been published in peer-reviewed scientific journals, conferences or workshops. *IC-P3* measures whether a study was duplicated. To this end, the duplication-measure, according to Kofod–Petersen's suggestion [22], was defined as follows:

- if a primary study with the same name and from the same authors has been indexed in more than one source, they will be considered as one study; and
- if more than one primary study from the same authors on an approximately the same topic have been published, only the most recent one will be included.

Finally, *IC-P4* measures whether a primary study passes the time-citation criterion. As pointed out by Meho [23], citation analysis has been used often over the past three decades to evaluate and quantify the importance of scientific research. It measures the quality of articles on the assumption that influential research studies are cited more frequently than others. However, citation analysis, solely, does not consider the publication date which may affect this analysis as recent articles are likely to have a lower citation number. In order to mitigate this limitation, the publication date was combined with the citation analysis. Consequently, the time-citation measure was defined, on the basis of Fig. 1, as follows.

- If a study has been published since 2010, it will directly pass the time-citation criterion (i.e., recent articles will not be assessed based on their number of citations).
- If a study has been published between 2010 and 2000, it will only pass the time-citation criterion when its number of citation (at the time of search) is equal to or greater than the minimum number of citation of the average number of citations for all publications in this period (i.e., 33.12) and the average number of citations for the publication year. For example, considering the publications that have been published in 2000, they are only included if their citation numbers are higher than 33 (i.e., $min(33.12, 139.1)$), while for publications that have been published in 2002, they are only considered if their citation numbers are higher than 11 (i.e., $min(33.12, 11.6)$). The reason for splitting this period into 10 individual groups is that, as shown in Fig. 1, the standard deviation of the average number of citations per year in this period is very high.
- If a study has been published before 2000, it will only pass the time-citation criterion when its number of citations is equal to or greater than the average number of citations of all the publications before 2000 (i.e., an article that has been published before 2000 is only included when its number of citations is higher than 34). As shown in Fig. 1, the standard deviation in this period is approximately one third of the average and, therefore, based on the so-called three-sigma rule of thumb [24], it suggests that data in this period are distributed normally. With further investigation, it is realized that the average number of citations per year in this period are consistent with a normal distribution ($P = 0.83$), where the normal distribution has an average value of 34.161 and a standard deviation of 14.8 and, therefore, this whole period is considered together.

The first content inclusion criterion, *IC-C1*, aims to consider studies that propose an architecture for a BPMS(thus if the title of a primary study comprises of "BPMS" and "architecture" but no architecture is actually presented in the paper, it will be excluded from the study). The second content inclusion criterion, *IC-C2*, is also defined because comparing various architectures with different application domain requirements is, indeed, not an easy process and, therefore, only the domain-independent components of the selected BPMS architectures are considered.

If a study does not meet any one of above-mentioned criteria, it will be excluded and, subsequently, other criteria will not be evaluated. Primary studies which pass these inclusion criteria are, nevertheless, subject to further assessments on the basis of two additional exclusion criteria. The first exclusion criterion, *EC1*, excludes all primary studies that only propose a specific component of a BPMS(e.g., if a publication meets all the inclusion criteria but it only presents an architecture for the Workflow definition tools it will be excluded from this study). There is an exception for this criterion; if a primary study only proposes an architecture for the Workflow enactment (i.e., the BPMS engine) component, it will not be excluded from the study. Finally, the very last selection criterion, *EC1*, excludes all primary studies that only propose non-functional requirements (e.g., performance or reliability) since, although non-functional requirements are critical, the focus of this investigation is mainly on the functional components of a BPMS architecture.

2.4. Conducting the review

In this section, we present the steps and intermediate results that lead to selecting the final set of primary studies.

Table 2
Inclusion/Exclusion criteria utilized for the SLR study.

Criteria		Description
Assess-ability	IC-A1	Is the full-text of study digitally accessible?
Inclusion criteria	IC-A2	Has the study been written in English?
Paper specification inclusion criteria	*IC-P1*	Does the study's title comply with the search strings?
	IC-P2	Has the study been published in a scientific peer-reviewed source?
	IC-P3	Does the study meet the duplication measure?
	IC-P4	Does the study meet the time-citation measure?
Content inclusion criteria	*IC-C1*	Has the study proposed an architecture for a BPMS?
	IC-C2	Has the study included domain-independent BPMS components?
Exclusion criteria	*EC1*	Has the architecture proposed a specific component for a BPMS?*
	EC2	Has the study only proposed non-functional requirements for a BPMS?

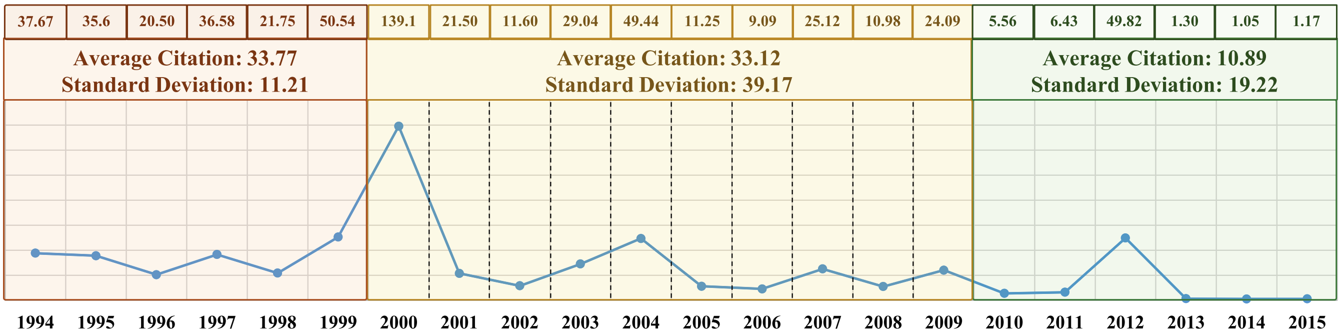

Fig. 1. Average number of citation per year over time.

To begin this selection procedure, we searched for papers whose titles contained a certain group of search strings in order to find a set of potential relevant primary studies. A total of 608 studies were taken from the five scientific databases. We designed a database to store the relevant bibliography information from the primary studies. This bibliographic information consisted of the following attributes:

- *electronic database*, which indicated the database that each primary study was retrieved from; the values for this field came from Table 1,
- *title*, which indicated the title of each primary study,
- *authors*, which indicated the authors of each primary study,
- *year*, which indicated the year that each primary study was published,
- *source*, which indicated the journal, conference or workshop where each primary study was published,
- *citation*, which indicated the number of citations for each primary study.

We then employed a developed program[1] to import the bibliographies as well as the numbers of the citations for the 608 found studies into the database. To this end, for each primary study, we extracted its bibliography from the search sources (e.g., IEEE Xplore) and its number of citations from *Google Scholar*. We used Google Scholar to obtain a fair comparison among the primary studies since the number of citations for the same article may differ in various sources, and – appropriately – all the primary studies are available in Google Scholar. In addition, because this number can change quickly over a short period, we retrieved these numbers for all the selected primary studies at the same time.

Having filled the database, we ran a script to determine the duplicated primary studies in the database. This action was not only based on the publications' titles but also we considered other available information, such as groups of authors (cf. Selection Criterion *IC-P3*). We, furthermore, ran another script to apply the time-citation measure (cf. Selection Criterion *IC-P4*). After having undertaken these two steps, the number of selected primary studies reduced to 301 and 135 respectively. Fig. 2 shows the distribution of the selected primary studies after applying the time-citation criterion over the search period. Then,

we extracted the full text of 128 relevant primary studies (7 articles were not available due to the University's License). Among the selected primary studies, two papers were written in a different language (i.e., one German and one Chinese) and two articles had been added to the database but their titles did not include any of the search string groups. Consequently, we manually excluded these four studies from the database due to *IC-P1* and *IC-P2*. Additionally, 6 papers had been published in non-peer-reviewed sources (e.g., magazines) and these were excluded as well. Finally, the review team members individually read all the 118 selected primary studies and critically assessed their relevance according to the content inclusion and exclusion criteria (i.e., *IC-C1, IC-C2, EC1* and *EC2*). Having discussed the results that were obtained, the review team agreed on 41 primary studies that were considered to be appropriate for the final inclusion. Table 3 shows the results of our selection procedure.

3. BPMS architecture classification framework

This section presents a framework to classify the BPMS architectures from the 41 primary studies. This framework covers five aspects, namely: *Root Architecture, Level of Elaboration (LoE), Architectural Style, Evaluation Method* and *Supported Functionality*. Each of these five aspects aims at providing the results relating to one of the research sub-questions (i.e., *RQ1* to *RQ5*). These aspects are further described in the rest of this section.

3.1. Root architecture

The first aspect in our classification framework is root architecture. This aspect is introduced to represent whether the proposed

[1] The source code of this program is available at https://github.com/shoopi/BPMS-Architecture-Survey

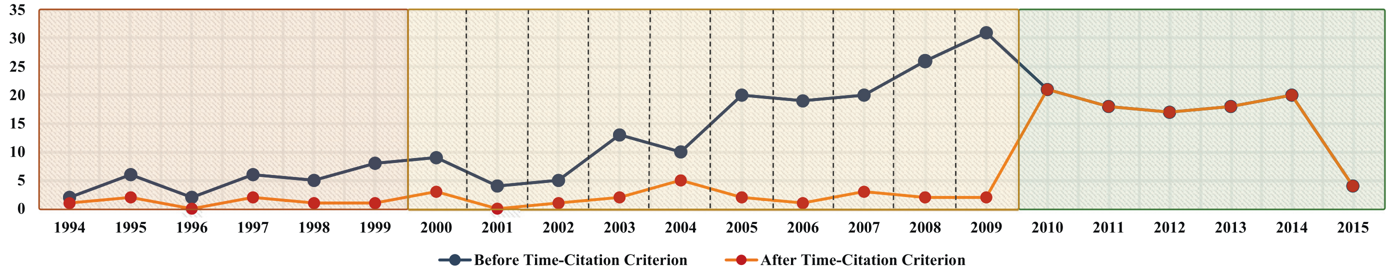

Fig. 2. Before and after applying the time-citation criterion.

Table 3
Final results of the primary study selection procedure.

Criteria	Reviewer 1	Reviewer 2	Selected
Excluded by *IC-C1*	68	72	**68**
Excluded by *IC-C2*	2	7	**3**
Excluded by *EC1*	6	5	**5**
Excluded by *EC2*	1	1	**1**
Total Excluded	77	85	**77**
Total Included	41	33	**41**

architecture in a primary study is built upon other proposed architectures. We have defined two classes for the root architecture aspect, namely: *Reference Architecture* and *Concrete Architecture*.

Since no commonly accepted definition of the term software reference architecture exists [25], we have adopted the definition of reference architecture given in [26], which introduces this concept as a predefined guideline (i.e., an abstract blueprint [27]) for the architecture of a particular domain (e.g., in our case BPMS), where the *structures, respective elements* and the *relationship among the elements* provide a template for concrete architectures.

A concrete software architecture presents the structure of a software system in terms of functional components and the interrelations among them for a specific system. In many cases, concrete architectures are designed in order to fulfill the business requirements of a single organization; however, these architectures can also be designed in an organization-independent manner [27].

Consequently, reference architectures typically aim at providing high-level design principles that can be reused in multiple situations in a specific domain [25] while concrete architecture aims at depicting the functional structure of a system that may not be reusable for other architectures. Ideally, a concrete architecture can be designed based on one or more reference architectures.

Firstly, we consider the following three reference architectures in our framework. Then, we added additional (reference) architectures, when the primary studies have been built according to other (reference) architectures.

3.1.1. WfMC reference model

The first discussions regarding reference architecture for a Workflow management system emerged during the 1990s when the Workflow Management Coalition (WfMC) put forward the Workflow reference architecture [28] in 1995. In this document the author presented a reference architecture for Workflow management systems by identifying their characteristics, by providing a common terminology and by introducing the necessary components for Workflow management systems. This reference architecture is shown in Fig. 3.

3.1.2. Mercurius reference architecture

Three years later, Grefen and Remmerts de Vries presented a new reference architecture for a full-fledged Workflow system, called Mercurius [3]. This reference architecture proposed supporting heterogeneous environments and mobile Workflow clients alongside the functionality that was offered in [28]. The Mercurius reference architecture is shown in Fig. 4.

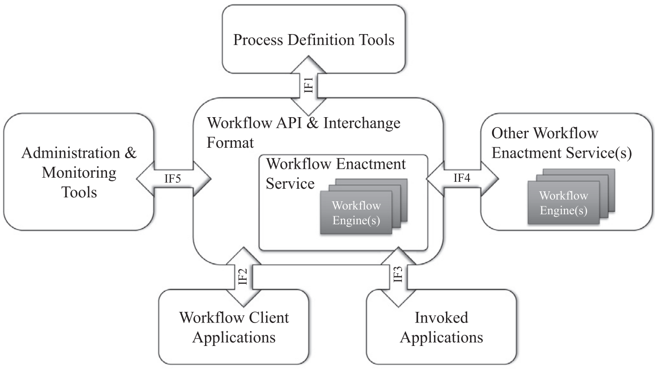
Process Definition Tools
IF1
Workflow API & Interchange Format
Administration & Monitoring Tools
IF5
Workflow Enactment Service
Workflow Engine(s)
IF4
Other Workflow Enactment Service(s)
Workflow Engine(s)
IF2
IF3
Workflow Client Applications
Invoked Applications

Fig. 3. The workflow reference model by WfMC (adapted from [28]).

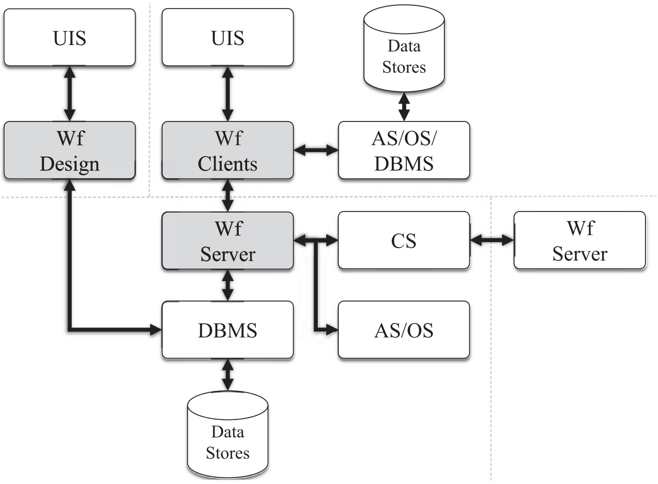

Fig. 4. Mercurius workflow reference architecture(adapted from [3]).

3.1.3. S3 reference architecture

The last decade has seen a growing trend towards developing more flexible and agile information systems in order to overcome the rapid changes in technologies as well as to deal with the dynamism issues in new business environments. Service-Oriented Architecture plays an important role in addressing these issues. A reference architecture for such systems has been proposed by Arsanjani et al. [29], called Service-Oriented Solution Stack (abbreviated as S3). The S3 provides a comprehensive architectural definition of a Service-Oriented Architecture(SOA) through nine layers. This reference architecture consists of architectural building blocks and the relationships among the blocks, the relationships among the layers, interaction patterns, and architectural decisions. One can employ

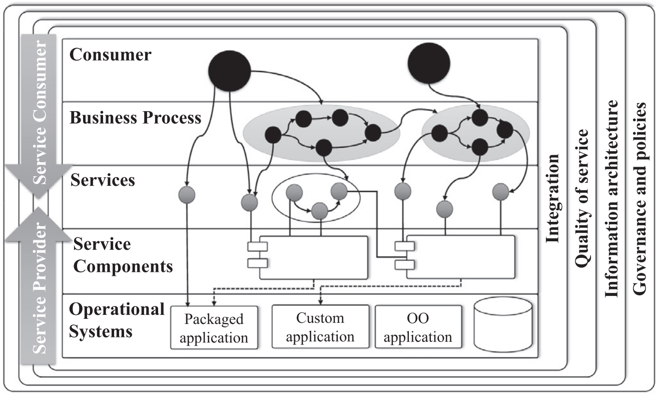

Fig. 5. Reference model and logical layers in S3(adapted from [29]).

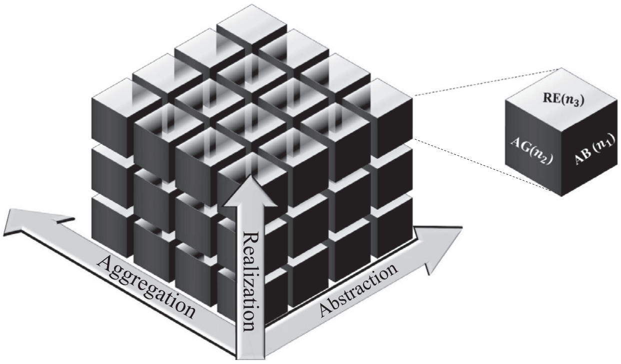

Fig. 6. The Level of Elaboration (LoE) Cube.

all these elements to create a SOA solution. This reference architecture has been also considered in this study since we have introduced two technical synonyms for BPMS with regard to Service-Oriented Architecture(SOA). In particular, we have employed orchestration and choreography (execution systems) as alternatives in our search strings (cf. Section 2.2). The S3 reference architecture is shown in Fig. 5.

3.2. Level of Elaboration (LoE)

The second aspect in our classification framework is the Level of Elaboration (LoE). This aspect facilitates a comparison among the selected architectures by positioning them into a three-dimensional cube [27], called the LoE cube (Fig. 6). This cube represents the dimensions with respect to which an architecture can be elaborated in more or less detail. The dimensions of the LoE cube are: the *Aggregation* dimension, the *Abstraction* dimension and the *Realization* dimension are presented in the remainder of this section.

3.2.1. Aggregation dimension

The term aggregation has been used to define the *level of detail* in an information system architecture with regard to the number of recognized components [27]. Based on this definition, an architecture, at the highest level of aggregation, presents a system as a single blackbox while an architecture at the lowest level of aggregation illustrates a system with very small identified subcomponents. We have further divided the aggregation dimension, based on the aspect framework for information systems [30], into two sub-dimensions, namely (i) the *system-aggregation* and (ii) the *data-aggregation*.

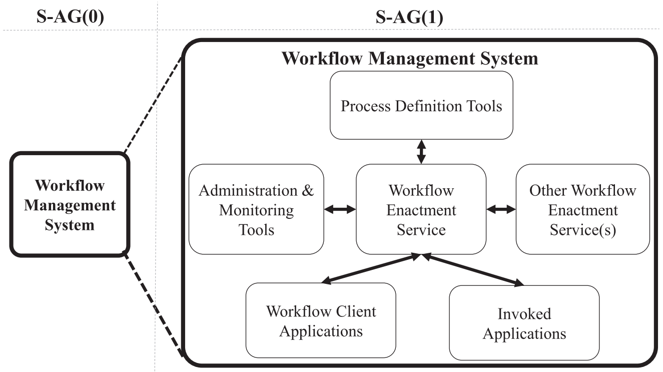
S-AG(0)
S-AG(1)
Workflow Management System
Workflow Management System
Process Definition Tools
Administration & Monitoring Tools
Workflow Enactment Service
Other Workflow Enactment Service(s)
Workflow Client Applications
Invoked Applications

Fig. 7. Example of architecture at the zero level and the first level of system aggregation.

System-Aggregation Dimension (S-AG). The system-aggregation dimension aims at positioning architectures in the LoE cube in terms of the granularity decomposition of their functional components. Based on the architecture that we have reviewed and also based on the granularity levels that have been used in presenting the Workflow reference model and Mercurius reference architecture, we have identified five distinct levels for the system-aggregation dimension.

S-AG(0). An architecture at this level of the system-aggregation dimension depicts a BPMS as a blackbox. If an architecture consists of a component that is in this granularity level, e.g., labeled with *Business Process Management System* or similar terms (e.g., Workflow management systems), we will position it at the S-AG(0) level of system-aggregation.

S-AG(1). An architecture at this level of the system-aggregation dimension shows a high-level architectural view of the components of a BPMS that are derived based on the high-level business requirements and the relationships among them. Architectures at this level are comparable with the Workflow reference model (as shown in Fig. 3) which includes: (i) *Workflow enactment services* or *Workflow engines*, (ii) *process definition tools*, (iii) *Workflow client applications*, (iv) *administration and monitoring tools*, (v) *invoked applications*, and (vi) *other Workflow enactment services*. If an architecture consists of components with a similar granularity level as mentioned in the Workflow reference model, we will position it at the S-AG(1) level of the system-aggregation dimension. Fig. 7 depicts the distinction between S-AG(0) and S-AG(1) levels.

S-AG(2). An architecture at this level of the system-aggregation dimension displays an architectural view on the employed components of a BPMS in more detail. This level can be seen as a view inside the components that are identified at the S-AG(1) level. Architectures at this level are comparable with Mercurius (as shown in Fig. 4), in which the *Workflow enactment server* module includes the *WF*[2] *Server Engine*, the *WF Client Interface* and platform interface modules such as: *AS/OS/DBMS Interfaces*. If an architecture consists of components with a similar granularity level, we will position it at the S-AG(2) level of the system-aggregation dimension. Fig. 8 illustrates the differences between S-AG(1) and S-AG(2) levels.

S-AG(3). An architecture at this level of the system-aggregation dimension illustrates an architectural view on the functional subcomponents of a BPMS that are derived based on the low-level

[2] Workflow

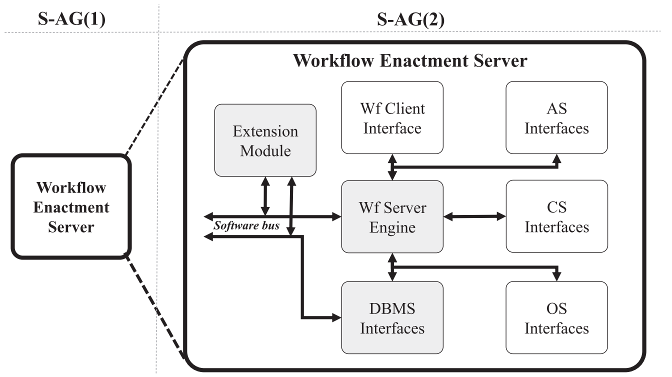

Fig. 8. Example of components at the first and second level of system aggregation.

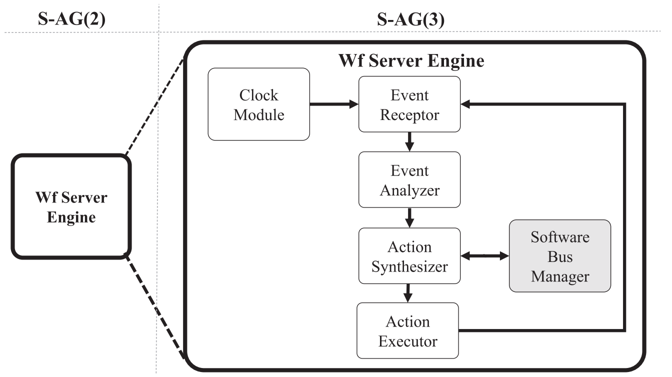

Fig. 9. Example of components at the second and third level of system aggregation.

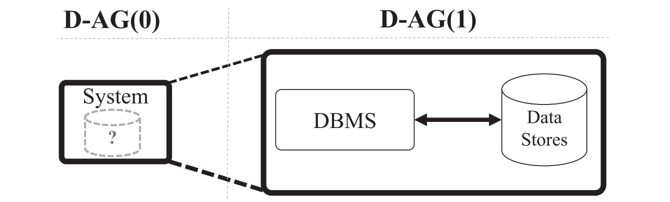

Fig. 10. Example architecture at the zero level and the first level of data aggregation.

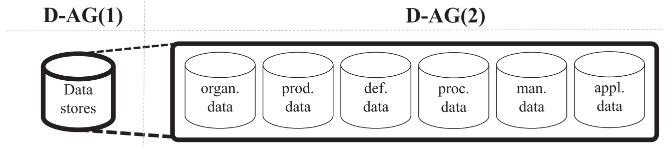

Fig. 11. Example of components at the first and second level of data aggregation.

requirements. This level can be seen as a view inside the components that are identified at the S-AG(2) level. Architectures at this level are comparable with Mercurius [3], in which the *Wf Server Engine* module includes the *Action Executor*, the *Action Synthesizer*, the *Event Analyzer* and the *Event Receptor*. If an architecture consists of components with a similar granularity level, we will position it at the S-AG(3) level of system-aggregation dimension. Fig. 9 shows the distinction between S-AG(2) and S-AG(3) levels.

S-AG(4). Finally, an architecture at this level of the system-aggregation dimension provides a very low-level architectural view on the functional detailed sub-components of a BPMS. We only observed one architecture at this level, where a component at the S-AG(3) level was broken down into inner components.

Data-Aggregation Dimension (D-AG). The data-aggregation dimension aims at positioning architectures in the LoD cube in terms of the granularity decomposition of their data-based components. Based on the architecture that we reviewed, we have identified four distinct levels for the data-aggregation dimension.

D-AG(0). An architecture at this level of the data-aggregation dimension depicts no specific data-related component in a BPMS architecture since, at this level, the whole system is represented as a black box and, thus, no specification data-related component is expected. However, some architectures at lower levels of the system-aggregation dimension may also have no *explicit* indication of any data-related components although data are *implicitly* considered. An example of this level is the Workflow reference model [28] which is positioned at the S-AG(1) for the system-aggregation dimension but at the D-AG(0) level of the data-aggregation dimension. In summary, if an architecture has no explicit data-related components – nevertheless, data are implicitly considered in the architecture – we will still position it at the D-AG(0) level of the data-aggregation dimension.

D-AG(1). An architecture at this level of the data-aggregation dimension shows a high level architectural view of the data-related components of a BPMS which can usually be depicted via a few Database Management System(DBMS) components. Architectures at this level are comparable with Mercurius at its Global Workflow Management System architecture, which includes: a *DBMS* component as well as two *Data Store* data-related components. If an architecture consists of data-related components with a similar granularity level, we will position it at the D-AG(1) level of the data-aggregation dimension. Fig. 10 depicts the distinctions between the D-AG(0) and D-AG(1) levels.

D-AG(2). An architecture at this level of the data-aggregation dimension displays an architectural view on the employed data-related components of a BPMS in more detail. This level can be seen as a view inside the components that are identified at the D-AG(1) level. Architectures at this level are comparable with Mercurius at the detailed architectures of inner components such as the Workflow enactment server architecture, which includes: a *Process Data*, a *Production Data* and an *Application Data*. If an architecture consists of data-based components with a similar granularity level, we will position it at the D-AG(2) level of the data-aggregation dimension. We observed three architectures at this level, where a component at the D-AG(2) level was broken down into inner components. Fig. 11 illustrates the differences between the S-AG(1) and S-AG(2) levels.

D-AG(3). Finally, an architecture at the fine-grained level illustrates an architectural view on the entities that are used in a BPMS. Architectures at this level are comparable with the agent-based cross-organizational Workflow architecture [31] which includes a class diagram for a process specification with a set of entities such as a *Role*, a *DataFlow* and an *EventGrouping*. If an architecture provides an Entity Relationship Diagram(ERD) or a Class Diagram of its entities, we will position it at the D-AG(3) level of the data-aggregation dimension. Fig. 12 shows the distinction between the D-AG(2) and D-AG(3) levels.

3.2.2. Abstraction dimension (AB)

The term abstraction has been used to define the level of concreteness in an information system architecture with regard to the software building blocks [27]. Based on this definition, an architecture at the highest level of abstraction presents rough information about the components therein while an architecture at the lowest

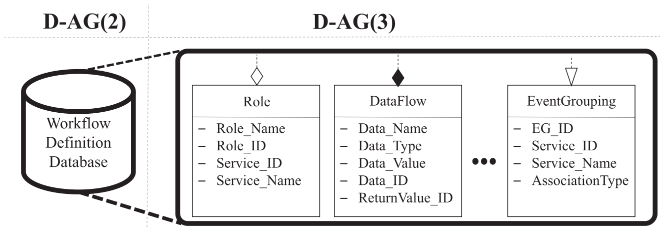

Fig. 12. Example of components at the second and third level of data aggregation.

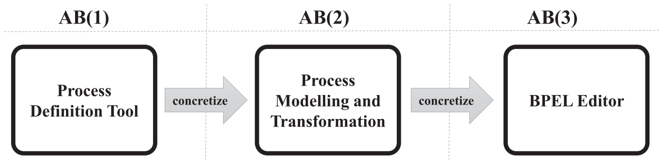

Fig. 13. Example of components at the first, second and third level of abstraction.

level of abstraction illustrates concrete information for the components therein (i.e., final decision about the products with their versions). Based on the architectures that we have reviewed, we have identified three distinct levels for the abstraction dimension.

AB(1). An architecture at this level of the abstraction dimension shows very-high level details concerning the components that constitute a BPMS architecture. An architecture is at this level of abstraction, if for the components therein no information about specific software products is provided. Architectures at this level are comparable with the Workflow reference model which includes the *process definition tools* component. If an architecture consists of components with a similar abstraction level as mentioned in the Workflow reference model, we will position it at the AB(1) level of the abstraction dimension.

AB(2). An architecture at this level of the abstraction dimension displays details concerning the components that constitute a BPMS architecture. An architecture is at this level of abstraction, if for each component therein some information about a *family* of software modules or a *family* of technologies are provided. A software family is defined as a set of systems with common high-level functionalities but with possibly different variations [32]. Architectures at this level are comparable with the architecture that has been proposed in [33] for a Secure BPMS in Service-Oriented Environments which includes the *Business Process Modeling and Transformation* components. Comparing these components with the *process definition tool*, it can be seen that the components at AB(2) provide more concrete information (i.e., the term, *tool*, is more abstract than the term, *modeling and transformation*). If an architecture consists of components with a similar abstraction level as mentioned in [33], we will position it at the AB(2) level of the abstraction dimension. The first two columns in Fig. 13 depict the distinction between the AB(1) and AB(2) levels.

AB(3). Finally, an architecture at this level of the abstraction dimension provides low-level details concerning the components that constitute a BPMS architecture. An architecture is at this level of abstraction, if for the components therein some information about *specific* software modules or technologies are provided. Architectures at this level are comparable with the architecture that has been proposed in [34] for a new web service composition technique which includes the *BPEL Editor* component. This component shows that the Web Services Business Process Execution Language(WS-BPEL) is the only modeling language for the modeling component which was identified at the AB(2) level. If an architecture consists of components with similar abstraction level as mentioned in [34], we will position it at the AB(3) level of the abstraction dimension. The second two columns in Fig. 13 illustrate the difference between the AB(2) and AB(3) levels.

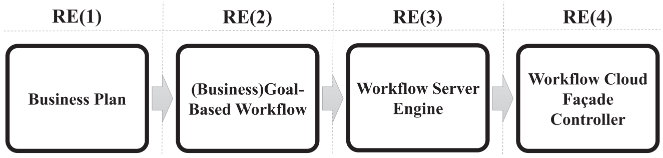
RE(1)
RE(2)
RE(3)
RE(4)
Business Plan
(Business)Goal-Based Workflow
Workflow Server Engine
Workflow Cloud Façade Controller

Fig. 14. Example of components at the different levels of the realization dimension.

3.2.3. Realization dimension (RE)

The term, realization, has been used to define the level of business-orientation in an information system architecture [27]. Based on this definition, an architecture at the first level of the realization dimension presents business goals that need to be fulfilled (i.e., what we want to achieve with the system described by an architecture) while an architecture at the fourth level of the realization dimension illustrates technologies that enables the fulfillment of the business goals (i.e., how we want to achieve things with an architecture). Based on [27] we have identified four distinct layers for the realization dimension.

RE(1). An architecture at this level of the realization dimension only illustrates the business requirements that can be addressed by the architecture. In contrast to the other three levels, how requirements can actually be met is out of the scope of this architecture. Architectures at this level of realization are designed to be used by domain experts. We observed no architecture at this level of realization among the architectures that we reviewed.

RE(2). An architecture at this level of the realization dimension displays more detail on the business requirements rather than on the technologies. Architectures at this level of realization are designed to be used by both domain experts and information systems developers. These architectures are comparable with the architecture that has been proposed in [10] for a new BPMS that integrates both goal- and activity-based perspective which includes the *(Business) Goal Workflow Engine* component. If an architecture consists of components with a similar realization level as mentioned in [10], we will position it at the RE(2) level of the realization dimension. We observed three architectures at this level of realization among the architectures that we reviewed. The first two columns in Fig. 14 depict the distinction between the RE(1) and RE(2) levels.

RE(3). An architecture at this level of the realization dimension shows more detail about the technologies rather than business requirements. In contrast to the previous levels, this level includes a very high-level view of the business requirements. Architectures at this level of realization are designed to be used by both information systems developers and domain experts. These architectures are comparable with the Workflow reference model [28]. If an architecture consists of components with a similar realization level as mentioned in [28], we will position it at the RE(3) level of the realization dimension. The majority of the architectures that we reviewed are positioned at this level of realization. The second and third columns in Fig. 14 illustrate the differences between the RE(2) and RE(3) levels.

RE(4). Finally, an architecture at this level of the realization dimension provides details concerning with how business requirements can be achieved in terms of using existing technologies. Architectures at this level of realization are designed to be used by information systems developers. These architectures are comparable with the architecture that has been proposed in [35] for a new dynamic Peer-to-Peer(P2P) BPMS which includes components that have been designed based on the concepts of the Web Workflow Peer Directory(WWPD) and, thus, it is only focused on how 'sharing process models among all Workflow participants' (as a main business requirement) can be achieved in terms of using the WWPD (as an existing technology). If an architecture consists of components with a similar realization level as mentioned in [35], we will position it at the RE(4) level of the realization dimension. The last two columns in Fig. 14 shows the distinction between the RE(3) and RE(4) levels.

3.3. *Architectural style*

The third aspect in our classification framework is architectural style. This aspect is introduced to represent higher-level architectural design decisions that have been made to describe the overall structure of an architecture. According to the definition proposed by Taylor et al. [36], architectural style is a set of design decisions and constraints that comprises the style of an architecture, as well as the beneficial qualities induced by these decisions and constraints. Consequently, the architectural style aspect in our classification framework provides the high-level design decisions (and constraints) regarding the style of BPMS architectures. Based on [36] and [27], we have adopted two main classes of architectural style: *Component-Based Style* and *Layered Style* since these two have been mainly used in the 41 primary studies.

3.3.1. *Component-based style*

The component-based style defines structure of an architecture by clustering coherent functionalities into components and encapsulating these functionalities by providing Application Programming Interfaces(APIs) that can be used by other components. This enforces high cohesion within and low coupling among the identified components in an architecture. Therefore, this style can increase the *modifiability* of an architecture which is defined by Fielding as the ease with which a change can be made to an architecture [37]. In addition, the component-based style provides autonomy to a great degree in the sense that the identified components therein can freely make use of each other. However, as suggested in [36], this lack of restriction in the structure of the component-based style may lead to a highly complex architecture and, thus, has a negative effect on the *scalability* of an architecture (which is defined by Fielding as the ability of an architecture to support large numbers of components, or interactions among the components, within an active configuration [37]).

If an architecture has been designed based on the above-mentioned characteristics, we will position it in the Component-Based Style class of the architectural style aspect in our framework. In the same way, the Workflow reference model, as shown in Fig. 3, is positioned in this class of architectural style.

3.3.2. *Layered style*

The layered style defines the structure of an architecture by distinguishing functionalities into an ordered sequence of layers, whereby each layer offers services (e.g., by providing APIs) that can be employed by the components residing in the above layers. There are two main designs of Layered Style: (i) *Strict Layering* (also known as *Linear Layering*) and (ii) *Loose Layering* (also known as *Acyclic Layering*). In a strict layering style, a layer can only make use of the layer directly below it while, in a loose layering style, a layer can make use of all the layers below it (i.e., it can bypass some layers below). This architectural style offers a very clear dependency structure since the lower layers are independent from the upper layers. Therefore, the upper layers can evolve independently from the lower layers as long as the interface semantics is unchanged and, thus, it can both improve the modifiability and the scalability of an architecture. However, as suggested in [36], the strict layering may cause a decrease in the performance of a system.

If an architecture has been designed based on the above-mentioned characteristics, we will position it in the Layered Style class of the architectural style aspect in our framework. It should be noted, however, that if a combination of these two styles has been used (i.e., using a component-based style combined with a layered style resulting in a stratified component-based architecture [27]), we will also position this architecture in the Layered Style class of the architectural style since layered architectures provide more structure. In the same way, [9] and [38] are positioned in this class of architectural style. In [9] the authors proposed an architecture design of a distributed BPMS in which the separation of the process logic (in one layer) and the application logic (in a layer above) can improve the flexibility and scalability of the realized BPMS. In [38] the authors proposed an extended BPMS architecture, in which a service contracting a layer, service binding layer and a service invocation layer have been used to enable flexibility by dynamically binding activities to their implementations at runtime.

Considering the layered architecture, since we are also interested in the rationale that underlies the structure and design of the constituent layers, we further classified this style into two subclasses: (i) the *Object-Oriented Web Workflow Peer Directory(SoC)* layering design principle, and (ii) the *Client-Server* Layering design principle. The former design principle, makes a separation between the presentation layer (i.e., (graphical) user interfaces), the logic/business layer(s), and the persistent layer (i.e., dealing with data) while the latter design principle makes a distinction between the provider of resources and services (i.e., called server), and the consumer of that resources and services (i.e., called client).

3.4. *Evaluation methods*

The fourth aspect in our classification framework is the evaluation method. This aspect is introduced to represent the way in which selected architectures have been evaluated. Considering the 41 selected primary studies, we have only identified 2 classes of the evaluation method: *Implementation* and *Case Study*.

3.4.1. *Implementation*

Implementation is defined as the process of transiting and mapping design decisions into specific implementation constructs (e.g., software components) [36]. These design decisions are usually described by the software architectures (e.g., in a formal architecture description language). Therefore, successful software artefacts appear as consequences of well-designed software architectures. Implementation is particularly useful in evaluating the applicability of a software architecture and, thus, it has been used in many articles as an evaluation method such as [39] where the authors designed, implemented, deployed and evaluated the SwinFlow-Cloud Workflow prototype based on the Amazon Web Services cloud.

3.4.2. *Case study*

In some studies, case studies have been designed to quantify the appropriateness of software architectures. Often, these case studies aim at measuring particular aspects of software systems such as evolution or performance. Therefore, if articles have employed a case study to evaluate their designed architecture, we will position them in this class of evaluation method in our classification framework. We also include *simulation*, which requires producing an executable model for (part of) a given system that is of

Table 4
Summary of selected BPMS surveys.

PS	Y	C	DB	S	ST	RA	S-AG	D-AG	AB	RE	STL	LYR	EVL
[42]	2015	7	ELSV	J	ST3	–	0	0	1	4	C	N/A	–
[43]	2015	0	SPRG	C	ST5	–	1	1	3	4	C	N/A	IMP
[44]	2014	3	SPRG	J	ST3	–	2	1	1	3	C	N/A	IMP
[12]	2014	0	ACM	C	ST3	–	2	0	3	3	C	N/A	IMP
[45]	2014	0	IEEE	C	ST3	–	2	1	1	3	L	SoC	IMP
[11]	2014	0	ACM	C	ST3	[28]	2	1	1	4	L	SoC	IMP
[39]	2013	3	SPRG	C	ST3	[28]	2	1	1	3	L	C-S	IMP
[46]	2013	1	IEEE	C	ST3	–	1	1	1	3	C	N/A	IMP
[47]	2013	0	IEEE	C	ST3	–	2	1	1	3	L	C-S	CS
[10]	2012	3	SPRG	W	ST3	[28]	2	3	1	2	C	N/A	IMP
[48]	2012	3	ACM	C	ST3	[28]	1	0	1	3	L	SoC	IMP
[9]	2012	3	IEEE	S	ST3	–	2	1	2	4	L	C-S	–
[40]	2011	15	IEEE	C	ST3	–	2	0	2	4	C	N/A	CS
[33]	2011	5	IEEE	C	ST2	[28]	2	0	2	3	L	other	–
[49]	2011	4	ELSV	J	ST2	–	2	1	2	3	L	SoC	–
[50]	2011	0	IEEE	C	ST3	[28]	4	1	2	3	C	N/A	IMP
[6]	2010	24	SPRG	J	ST1	–	1	1	1	3	C	N/A	CS
[7]	2010	16	SPRG	C	ST2	–	3	0	2	3	C	N/A	CS
[4]	2010	11	ELSV	J	ST3	–	2	1	1	4	C	N/A	CS
[51]	2010	11	ELSV	J	ST3	[28]	1	0	1	4	C	N/A	CS
[8]	2010	9	IEEE	C	ST1	–	1	0	2	4	L	SoC	IMP
[5]	2010	5	ACM	W	ST3	–	1	1	1	3	C	N/A	IMP
[52]	2010	1	ELSV	W	ST3	[28]	2	0	2	4	L	SoC	IMP
[34]	2010	0	IEEE	C	ST3	–	1	1	3	4	L	SoC	–
[53]	2009	97	IEEE	J	ST3	[3,28]	3	1	3	4	L	SoC	IMP
[54]	2008	73	ISI	J	ST3	–	4	1	3	4	C	N/A	IMP
[55]	2007	137	ELSV	J	ST2	–	2	2	2	3	C	N/A	IMP
[56]	2005	117	ACM	C	ST5	–	1	0	3	4	L	SoC	IMP
[57]	2005	95	ELSV	J	ST3	[28]	2	0	1	4	C	N/A	IMP
[58]	2004	170	ACM	W	ST3	–	3	3	3	4	L	SoC	IMP
[35]	2004	94	ELSV	J	ST3	[28]	2	0	2	4	C	N/A	IMP
[59]	2004	50	SPRG	C	ST3	[28]	2	2	1	3	L	SoC	IMP
[60]	2004	43	ACM	S	ST3	–	2	1	2	3	L	SoC	–
[38]	2003	38	ELSV	J	ST3	[28]	2	1	1	3	L	SoC	–
[31]	2002	39	IEEE	W	ST3	[61]	3	3	3	4	C	N/A	–
[62]	2000	1889	ISI	C	ST2	–	2	1	1	3	L	other	–
[3]	1998	63	ELSV	J	ST3	[28,63]	3	2	2	3	L	SoC	–
[64]	1997	131	IEEE	W	ST3	–	2	1	1	3	L	SoC	IMP
[65]	1995	249	SPRG	C	ST3	[28]	2	1	1	2	C	N/A	–
[63]	1995	72	SPRG	C	ST3	–	2	1	1	2	L	SoC	–
[66]	1994	53	IEEE	C	ST3	–	1	0	2	4	C	N/A	–

particular interest [36], in this class in our framework. In the same way, Barrett et al. who proposed a cloud Workflow scheduling approach based on a Markov Decision Process [40] and subsequently employed Cloudsim [41] to simulate four separate data centers in four geographic locations, is positioned in this class in our classification framework. In particular, Barrett et al. designed a case study in order to evaluate (i) the cost savings of their approach, and (ii) the variable work load on their system (i.e., load variance) when the amount of data increases across the system.

4. Results

This section presents the results that were obtained from our SLR study in relation to each of the research questions. Table 4 provides a summary of the selected primary studies (cf. Section 2) on the basis of the classification framework (cf. Section 3). This Table consists of 14 columns in which the first 6 columns are used to specify the selected primary studies, and the rest show our analysis based on the classification framework. For each entry in this table:

- the PS column provides a reference to the bibliography of a primary study,
- the Y column shows its publication year,
- the C column displays its number of citations,
- the DB column refers to its publication source, and its values are ACM for ACM Digital Library, IEEE for IEEE Xplore, ELSV for Scopus, SPRG for SpringerLink, and, finally, ISI for Web of Science,
- the S column expresses the publication source type and its values are J for Journal, C for conference proceeding, W for Workshop, and S for Symposium,
- the ST column denotes which group of search strings was used to find the primary study and its values are ST1 for “BPM”, ST2 for “Business Process Management”, ST3 for “Workflow”, ST4 for “Orchestration” and, finally, ST5 for “Choreography”,
- the RA columns investigates which Root Architecture (if any) has been used by the primary study,
- the System-Aggregation Dimension (S-AG), the Data-Aggregation Dimension (D-AG), the Abstraction Dimension (AB) and the Realization Dimension (RE) columns are employed to position the primary study into the LoE cube,
- the STL columns looks into the Architectural Style that has been employed by the primary study and its values are C for Component-Based Style and L for Layered Style,
- the LYR columns further analyzes the layering rationale that has been used by the primary study and its values are C-S for Client-Server design principle, SoC for Object-Oriented based Separation of Concern design principle, and not applicable (N/A) if the architecture is not layered, and finally
- the EVL explores the 3.4 (if any) that has been employed by the primary study, and its values are IMP for Implementation and CS for Case Study

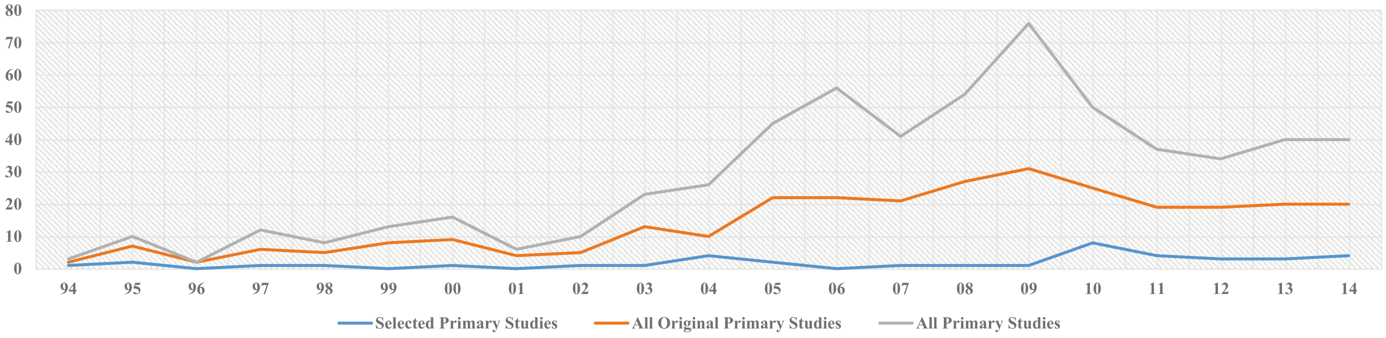

Fig. 15. Distribution of all the studies in comparison with the selected studies over the years.

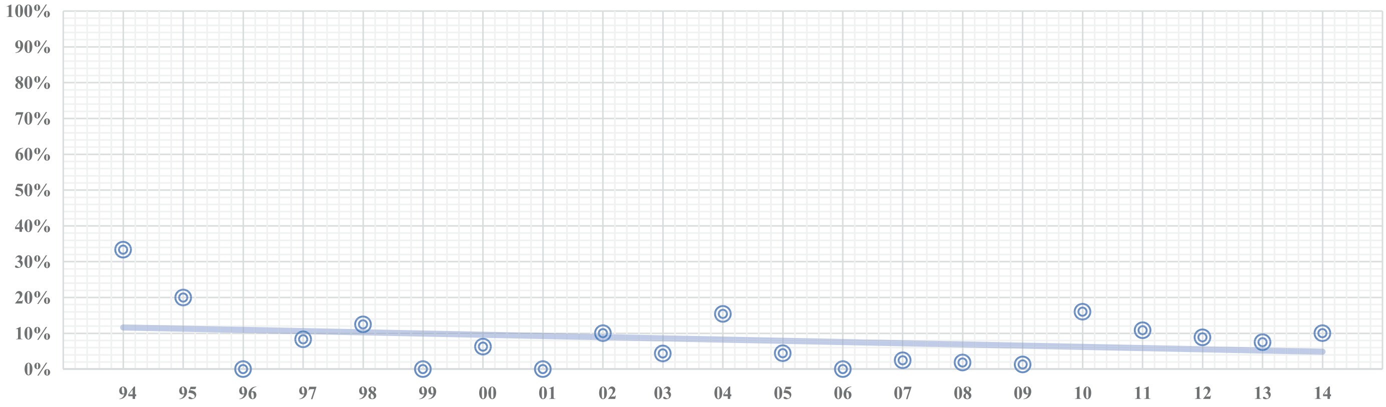

Fig. 16. Ratios of the selected studies to all the studies over the years.

4.1. Selected primary studies on BPMS architecture

The main research question that needs to be answered is *which relevant primary studies have been published in the area of BPMS architecture and where were they published?*

From the 608 primary studies, which were obtained by our search, we identified 301 original primary studies (i.e., after removing duplicated articles) and, subsequently, 41 studies to be considered for our SLR study after applying the inclusion and exclusion criteria. The line chart in Fig. 15 compares the distribution of all the original and selected primary studies over the years. Based on this chart, no substantial differences between the three graphs can be seen in terms of their trends. The dotted chart in Fig. 16 depicts the ratio of selected studies to all the published studies per year. In total, we selected approximately 7% of articles from which 1994 has the highest proportion at 33% while 1996, 1999, 2001 and 2006 have the lowest proportion at 0%. This fact also proved the reliability of our inclusion or exclusion criteria (cf. Section 2.3).

In addition, the Original Primary Studies trendline (the orange trendline) in Fig. 15 reveals that there was a gradual growth in the number of publications between 1994 and 2005 in the field of BPMS architecture research (as shown by the orange trendline). However, after 2005 this research continued at a fairly even level until there was a marked increase in 2009. One reason for this sharp rise may be the renewed interest of researchers in the field of process enactment infrastructure as pointed out by van der Aalst [67] in this year. Since 2011, again, there has been a even situation in the number of published articles. Fig. 15 only contains data up until 2014 because we conducted our search in July 2015 and, thus, for 2015 we only obtained the studies that have been published in the first half of this year. Although these studies are not shown in Fig. 15, we included them in our SLR analysis.

Among all the selected primary studies, three were published in proceedings of the Business Process Management (BPM) Conference, two were published in proceedings of the Web Services Conference and the same number was published in the Information and Software Technology (IST) Journal. The bar chart in Fig. 17 illustrates the distribution of the selected primary studies over the years in terms of their publications' source. Overall, it can be seen that 48.78% of the selected primary studies were reported in Conference proceedings which fall above Journal publications and Workshop proceedings which published 34.15% and 14.63% of the selected primary studies respectively. We also included one study (i.e., 2.44% of the population) that was published in a symposium on Distributed Computing.

Altogether, all the 41 selected primary studies proposed an architecture for a whole BPM system or at least for the Workflow Enactment Service (Engine) component. In order to properly answer the main research question, we decomposed that into five sub-questions. The next sections will provide answers to the five sub-questions by further investigating the content of the selected architectures according to the developed BPMS architecture classification framework (cf. Section 3).

4.2. Used root architectures

The first research sub-question that needs to be answered is *to what extent the architectures in the selected primary studies were built upon other existing architectures?* The developed classification framework seeks to provide an answer to this question by introducing the Root Architecture aspect. Accordingly, we presented three reference architectures: the *WfMC Reference Model* [28], *Mercurius Reference Architecture* [3] and *S3 Reference Architecture* [29]. However, contrary to our expectations, the most striking observation emerging from Table 4 is that the majority (i.e., around 60%) of the selected primary studies have been composed from scratch (i.e., not based on the existing architectures).

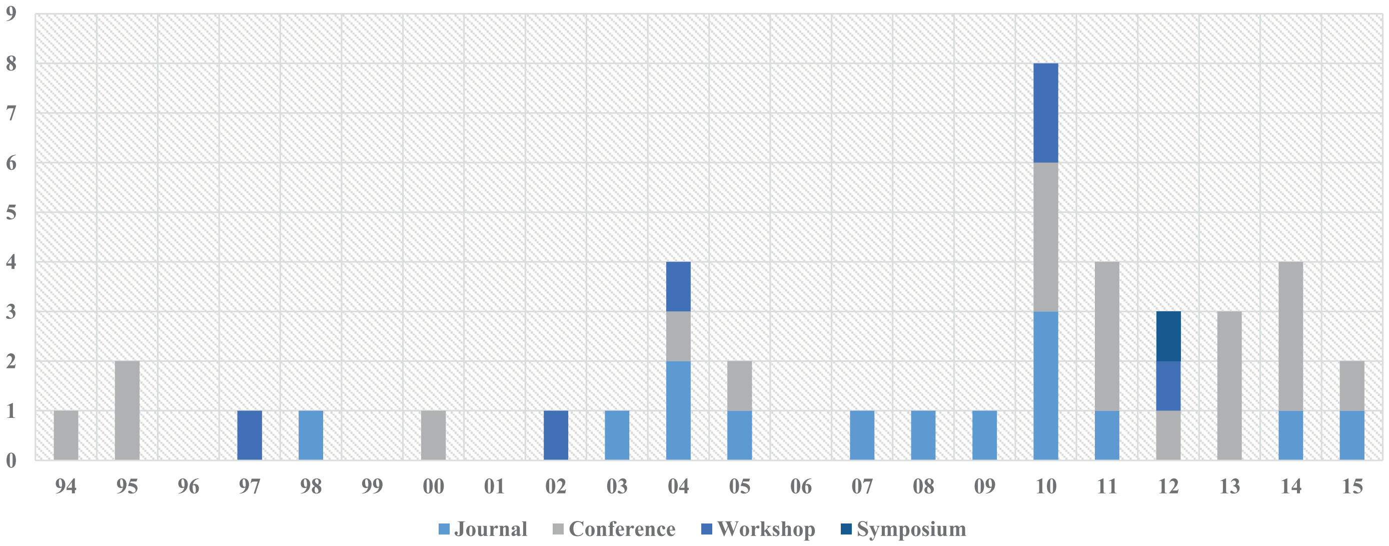

Fig. 17. Distribution of the selected articles over the years w.r.t. publications' sources.

Surprisingly, none of the selected primary studies have been *directly* composed based on the S3 reference architecture. We use the term directly, since although the S3 reference architecture has not been explicitly mentioned in the selected publications, it is still possible to claim that some of the architectures therein has *partially followed* this reference architecture. For example, in designing an SOA-BPM-based architecture for an intelligent power dispatching system [8], the authors have partially followed the layering logic that has been proposed in S3, but they have not explicitly mentioned S3 reference architecture in their work.

Another finding was the low number of the primary studies (i.e., only one) that have been built upon Mercurius reference architecture. Around 35% of the primary studies have used the Workflow reference model as a basis for other BPMS architectures.

In addition to these three reference architectures, two concrete architectures have also been used as starting points in designing the architectures in the selected primary studies: a Workflow-based architecture to support scientific database applications (i.e., *WASA*) [63], and Workflow automation through agent-based reflective processes (i.e., *WARP*) [61]. Each of these two architectures have been employed once in the primary studies.

4.3. Level of Elaboration (LoE) of BPMS architectures

The second research question that needs to be answered is*to what extent were the architectures in the selected primary studies elaborated in terms of details and technologies?* The developed classification framework seeks to provide an answer to this question by introducing the Level of Elaboration (LoE) aspect which consists of three dimensions.

We further decomposed the Aggregation Dimension dimension into the System-Aggregation Dimension (S-AG) and the Data-Aggregation Dimension (D-AG). According to the system-aggregation dimension, from Table 4 it can be seen that the second level, S-AG(2), has been predominantly used as a target for the architectures in the selected primary studies. In the same way, according to the data-aggregation dimensions, by far most of the architectures are designed at the first level, D-AG(1). The upper half of Table 5 summarizes our analysis of the selected primary studies.

From this tables, it is apparent that the majority of the primary studies present their architectures neither at the very coarse-grained level of system aggregation (S-AG(0)), nor at the very fine-grained level (S-AG(4)). Around 32% of the studies present architectures without any *explicit* data-related components, i.e., at the very coarse-grained level of the data-aggregation dimension (D-AG(0)).

Table 5
Summary of the Level of Elaboration's dimensions.

System-Aggregation Dimension (S-AG)				
S-AG(0)	S-AG(1)	S-AG(2)	S-AG(3)	S-AG(4)
2%	24%	**57%**	12%	5%
Data-Aggregation Dimension (D-AG)				
D-AG(0)	D-AG(1)	D-AG(2)	D-AG(3)	
32%	**54%**	7%	7%	
Abstraction Dimension (AB)				
AB(1)	AB(2)	AB(3)		
49%	31%	20%		
Realization Dimension (RE)				
RE(1)	RE(2)	RE(3)	RE(4)	
0%	7%	**49%**	44%	

Considering the second dimension of the LoE aspect, the Abstraction Dimension (AB), just below half of the selected primary studies (i.e., 49%) express their architectures at the abstract level (AB(1)). This result suggests that most of the selected architectures have been designed at the same level of abstraction as the Workflow reference model. There are also a number of primary studies (i.e., 31%) that presents their architectures at the middle level of abstraction (AB(2)). Finally, approximately one fifth of the selected studies (i.e., 20%) proposes architectures at the concrete level of abstraction (AB(3)). Most of these architectures have been developed for specific disciplines (e.g., health-care or scientific Workflow) rather than being domain-independent. The third part of Table 5 shows the distribution of the primary studies at different levels of abstraction.

Lastly, considering the third dimension of the LoE aspect, the Realization Dimension (RE), we observed no primary studies at the very-business-oriented level (RE(1)) and only three articles at the business-oriented level (RE(2)), which is only around 7% of the whole population. One reason why the selected primary studies have declined to provide architectures at the two business-oriented levels of realization dimension can be explained by their target audiences who tend to be technical computer scientists rather than business developers. Subsequently, the majority of the selected studies report their architectures as being either at the technology-oriented level (RE(3)) or at the very-technology-

Table 6
Analysis of the architectures that used the component- versus layered-based styles.

System aggregation			Data aggregation			Abstraction			Realization		
	C	L		C	L		C	L		C	L
S-AG(0)	5%	0%	D-AG(0)	40%	24%	AB(1)	50%	48%	RE(1)	0%	0%
S-AG(1)	30%	19%	D-AG(1)	45%	62%	AB(2)	30%	33%	RE(2)	10%	5%
S-AG(2)	45%	67%	D-AG(2)	5%	9%	AB(3)	20%	19%	RE(3)	40%	57%
S-AG(3)	10%	14%	D-AG(3)	10%	5%				RE(4)	50%	38%
S-AG(4)	10%	0%									
Total	20	21		20	21		20	21		20	21

oriented level (RE(4)). The former sub-dimension has a slightly greater share among the selected architectures (i.e., 49%), and the rest of studies (i.e., 44%) present their architectures at the latter sub-dimension level. The lower part of Table 5 illustrates the distribution of the primary studies at different levels of realization dimension.

In summary, these results show that most of the selected BPMS architectures:

- are composed of more detailed components than the Workflow reference model;
- include a few very coarse-grained data-related components;
- consist of components at the same level of concreteness as the Workflow reference model; and
- tend to be more technology-oriented rather than business-oriented.

4.4. Architectural style used by BPMS architectures

The third research question that needs to be answered is *to what extent were the architectures in the primary studies following a specific architectural style?* For this purpose, we introduced two main styles into our classification framework: the Layered Style and the Component-Based Style. We discussed the approaches that we used to classify architectures based on these two styles in Section 3.3. Based on our analysis, these two styles were employed almost equally. More specifically, the component based layered architectures were used in 20 primary studies, while the other 21 followed the layered based style.

We further investigated whether it is possible to reveal a relation between the level of elaboration in the selected architectures and the styles that they employed. However, according to the detailed analysis shown in Table 6 for the component-based and layered architectures, no relation was found. In particular, this table shows the distribution of the primary studies which employed a particular architectural style at different levels of the LoE cube. For example, around 30% of the selected primary studies that are positioned at the first level of system aggregation (S-AG(1)), have used the component-based architectural style (depicted by “C” in the Table) while approximately 19% of the studies at the same level of the LoE cube have followed the layered architectural style (specified by “L” in the Table). In addition, with regard to the total number of studies that have followed the component-based style (20 articles), Table 6 shows that half of the component-based architectures (i.e., 10 architectures) are positioned at the first level of the abstraction dimension (AB(1)), while the other half is divided into 6 and 4 component-based architectures that are positioned at the second (AB(2)) and third level (AB(3)) of the abstraction dimension respectively.

In addition, in order to gain a better understanding of the layered architecture, we also analyzed the rationales that underlie the structure of these architectures. To this end, we found more than 71% of the layered architectures have followed the Object-Oriented Separation of Concerns design principle (i.e., separation between the presentation layer, the logic/business layer(s), and the persistent layer), around 19% of the mentioned architectures’ designs have been based on the Client-Server design principle and the other two architectures have not been specifically designed according to these two principles. With respect to these results, it is clear that BPMS architects tend to use the Object-Oriented Separation of Concern design principle as the basis for their architectures.

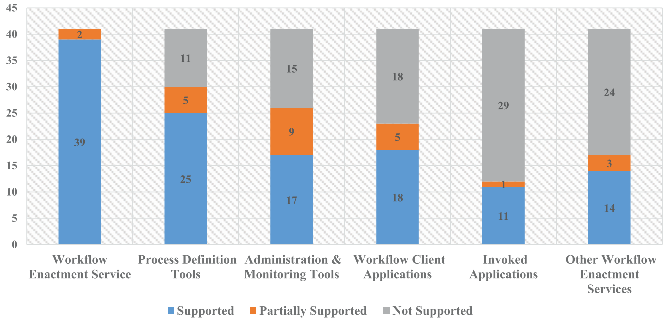

Fig. 18. Summary of the supported workflow reference model functionalities by the selected primary studies.

4.5. Evaluation methods used by BPMS architectures

The forth research question that needs to be answered is *how were the architectures in these studies evaluated?* For this purpose, we introduced two main evaluation approaches in our classification framework: Implementation and Case Study. We discussed the way that was used to classify architectures based on these two classes in Section 3.4. Based on our analysis, we found that none of the selected primary studies were evaluated on the basis of one of the well-known software architecture evaluation methods (e.g., SAAM [68] or ATAM [69]). However, the majority (i.e., approximately 54%) of these architectures were implemented. In addition to implementation, the applicability of 14% of the primary studies was evaluated based on designed case studies. Taken together, these results shows that 32% of the primary studies have not been evaluated and, therefore, there is still a need to assess the effectiveness of these studies.

4.6. Supported functionality by BPMS architectures

The final research question that needs to be answered is *which main functionalities have been addressed in the primary studies?* For this analysis, the functionalities offered by the main six components of the Workflow reference model have been adopted to explore the extent to which the selected 41 primary studies support each of these functionalities. Note that the focus of this paper is on the high-level (abstract) functionalities that need to be supported by a BPMS.

The bar chart in Fig. 18 summarizes the results from our analysis. The detailed results of our analysis have been presented in [70]. Because of the *EC1* exclusion criteria, over 95% of the selected primary studies explicitly support a functionality that provide the

Workflow Enactment Service functionality. However, the rest of studies also considered this functionality but they did not explicitly address that. Around 73% of the selected articles deal with the Process Definition Tools functionality. Just above 41% of the studies have explicitly considered the Administration & Monitoring Tools functionality; in addition to the 22% of the articles, which implicitly address this functionality in their architecture. One unexpected finding was the high extent (i.e., around 44% of studies) to which Workflow Client Application functionality has been neglected. One of the other interesting finding is the weak coverage of the external integration functionality, i.e., the functionality that is provided by the Invoked Application and Other Workflow Enactment Services. On average, approximately 70% of the selected primary studies have not treated the external integration in much detail.

According to these data, we can infer that the functionalities that are important for a single organization to be able to use a BPMS(e.g., WES and PDT) have been extensively covered by the selected primary studies. However, a possible explanation for the low support of the integration-based functionalities (e.g., IA and OWES) may be the lack of adequate research on inter-organizational Business Process Management Systems' collaborations.

5. Research opportunities for architecture of business process management systems

The findings from this SLR study provide substantial insight into the existing architectures of BPMSs and possible future research directions. The data from the SLR reveals three important research directions as we will explain in this section. Firstly, further work is required to establish an up-to-date reference architecture for BPM systems. Secondly, there is much room for further refining BPMS architectures with respect to technology for inter-organizational collaborations. Finally, there are still many questions that must be answered regarding the impact of different design choices for the Invoked Application and Workflow Client Application components.

5.1. Modernizing reference architectures

The data shows that the architecture of BPMSs has been a subject of research since the 1990s. However, as shown in Section 4.2, the majority of the existing architectures (around 60%) were built from scratch and were not based on existing (reference) architectures. This can partly be caused by the fact that apart from the long-established Workflow Reference Model [28] and the Mercurius reference architecture [3], which date back 20 years ago, there is a general lack of reference architectures for BPM systems. This points towards a need for the development of an updated BPMS reference architecture.

Additionally, the advent of big data analytics has changed the scope of BPM systems. This becomes apparent, for example, in the architecture presented in [11], where a component called Runtime Behavior Analytics has been proposed that employs past and current execution data to predict and control upcoming activities. Therefore, modern BPMS reference architectures must take functionalities into account that are provided by traditional BPM systems (e.g., business process modeling and execution) as well as functionalities that are provided by so called business process intelligence (e.g., process monitoring and control, data ingestion, data management, data analytics). When a BPMS reference architecture is designed to support the mentioned functionalities, we can argue that it supports the complete BPM life-cycle [71]: design, configuration, execution, control, and diagnosis.

Finally, modern BPMS reference architectures can be enriched by presenting them at a higher level of elaboration (cf. Section 3.2), in particular with respect to their interfaces. Some interfaces have been studied in more detail since the advent of BPMS reference architectures and, consequently, can be included in modern reference architectures. For example, the interface between Process Modeling Tools and Workflow Enactment Services has been thoroughly investigated as part of the BPMN standard specification [72]. At the same time, little work has been carried out on the standardization of other interfaces. Consequently, these interfaces can be studied in more detail to yield a BPMS reference architecture that can also serve as a standard.

5.2. Including inter-organizational aspects

Around 30% of the selected primary studies have addressed the concept of inter-organizational collaboration in the design of their BPM systems. However, given the high complexity of such collaborations [73], many challenges can be studied in more detail. One of these challenges concerns the *autonomy of organizations*, which requires that each BPMS involved in an inter-organizational collaboration works independently and is not obliged to divulge the details of its internal business processes to other BPMSs. To realize full autonomy, functionality is needed which provides a trust mechanism by offering features such as message encryption techniques, signature verification, authentication, and authorization in the context of inter-organizational Workflow.

Another challenge that requires further investigation is dynamism and flexibility in an inter-organizational collaboration [74], which is one of the key characteristics of a successful collaboration in modern-day business [75]. To support such functionality, a BPMS must enable inter-organizational processes to dynamically adapt to changes quickly and flexibly [76,77]. These changes can vary from business-driven changes (e.g., alterations in stakeholders' requirements or amending regulations and legal statements) to technology-driven changes (e.g., fostering innovation by adopting new technologies).

Finally, standardization and interoperability issues have to be considered in the design of BPMS-based collaborations [78]. The designed interfaces for such collaborations must be capable of handling both synchronous and asynchronous communication patterns. Furthermore, BPMSs must provide service invocation environments that aim at connecting, mediating, and managing interactions across heterogeneous platforms and other BPMSs. Ideally, such systems include service repositories or service registries, which provide catalogs of available and known services. Having provided these functionalities, BPMSs can enable inter-organizational collaborations.

While many of these issues have been studied – at least to some extent – in the context of SOA, a more detailed integration between BPMS, which traditionally focuses on intra-organizational processes, and SOA, which traditionally focuses on inter-organizational processes, remains to be studied.

5.3. Design choices for Invoked Applications and Workflow Clients

As the data in Fig. 18 shows, the internal structure of Invoked Applications and Workflow Client Applications has not been studied in much detail. This can be considered a gap in the literature, because different interaction scenarios with the BPM system can be envisioned that each require a different design and, eventually, different implementation costs and usability of the target components.

For example, on the one hand many traditional SOA systems were originally designed to not require any human intervention and consequently did not have a Workflow Client Application at all. On the other hand, sophisticated Workflow Client Applications have also been proposed that can be utilized to gain insights into the currently executed cases and activities within those cases and have advanced functionality for work distribution. In addition, Workflow Client Applications can be implemented both for thin clients and fat clients. They can be developed specifically for a particular Workflow Enactment Services, or they can be adapted from a general purpose client. They can be part of a dedicated information system (e.g., SAP Business Workflow), or they can be part of a programming framework (e.g., Windows Workflow Foundation [79]). Each of these - and other - possible scenarios lead to different architectural choices and can be investigated in more detail.

6. Conclusions

In this paper, we presented a Systematic Literature Review (SLR) that identifies and classifies existing architectures for Business Process Management Systems (BPMS).

We presented a review protocol which resulted in a collection of 608 research papers and by applying selection criteria returned 41 primary studies that describe BPMS architectures. We analyzed and classified these architectures based on a BPMS architecture classification framework, which consists of five main aspects: root architecture, level of elaboration (LoE), architectural style, evaluation method and supported functionality.

The root architecture aspect represents whether the proposed architecture in a primary study has been built upon other proposed architectures. Interestingly, our analysis has identified that the majority of selected architectures (i.e., around 60%) have been composed from scratch (i.e., not based on existing architectures).

The level of elaboration (LoE) aspect facilitates a comparison among the primary studies by classifying their architectures with respect to the level of aggregation and the level of abstraction at which they are described as well as the level of realization of their components. Considering this aspect, our analysis has shown that the selected primary studies have been described in more detail than the Workflow reference model and tend to be technology-oriented rather than business-oriented.

The architectural style aspect represents high-level architectural design decisions made by the authors of each primary study in order to describe the overall structure of their proposed architecture. Regarding this aspect, our analysis has demonstrated that almost half of the architectures have been designed based on the component-based architectural style while the other half have been composed based on the layered architectural style. We found no relation between the level of elaboration of the selected architectures and the architectural style that they employed.

The evaluation method aspect represents the way in which the architectures in the primary studies have been evaluated. With respect to this aspect we found that the majority of the selected primary studies (i.e., around 54%) have provided (proof-of-concept) implemented systems. However, approximately 32% of the primary studies did not include any kind of evaluation.

Finally, the supported functionality category represents the main functionalities that have been addressed by the proposed architecture. Our findings for this aspect suggest that although extensive research has been carried out on the architectures of BPMS for a single company, only a small part of them have adequately covered inter-organizational collaboration as well (i.e., only around 30% of the selected primary studies).

Taken together, these findings provide important insights into the current body of knowledge in the domain of BPMS architecture. As another result our research has thrown up opportunities in need of further investigation in the area of BPMS architecture. In particular, we found that, apart from the long-established Workflow Reference Model and a few more (reference) architectures from the late 1990s and early 2000s, there is a general lack of a modern reference architecture for BPM systems. Moreover, well-known issues in inter-organizational BPMS-based collaborations, such as interoperability, dynamicity and security, are important topics, but are mostly ignored in current BPMS architectures.

References

[1] A. Baumgrass, M. Botezatu, C. Di Ciccio, R. Dijkman, P. Grefen, M. Hewelt, J. Mendling, A. Meyer, S. Pourmirza, H. Völzer, Towards a methodology for the engineering of event-driven process applications, in: Business Process Management, 256, Springer, 2015, pp. 501–514.

[2] D. Hollingsworth, et al., The workflow reference model: 10 years on, Fujitsu Services, UK; Technical Committee Chair of WfMC, Citeseer, 2004.

[3] P. Grefen, R. Remmerts de Vries, A reference architecture for workflow management systems, Data Knowl. Eng. 27 (1) (1998) 31–57.

[4] C.-H. Tsai, K.-C. Huang, F.-J. Wang, C.-H. Chen, A distributed server architecture supporting dynamic resource provisioning for BPM-oriented workflow management systems, J. Syst. Software 83 (8) (2010) 1538–1552.

[5] M. Naseri, S.A. Ludwig, A multi-functional architecture addressing workflow and service challenges using provenance data, in: Workshop on PhD Students in Information and Knowledge Management, ACM, 2010, pp. 91–94.

[6] Y.-B. Han, J.-Y. Sun, G.-L. Wang, H.-F. Li, A cloud-based bpm architecture with user-end distribution of non-compute-intensive activities and sensitive data, J. Comput. Sci. Technol. 25 (6) (2010) 1157–1167.

[7] V. Muthusamy, H.-A. Jacobsen, BPM in cloud architectures: Business process management with SLAs and events, in: Business Process Management, Springer, 2010, pp. 5–10.

[8] L. Zhang, M. Ge, X. Bi, S. Chen, A SOA-BPM-based architecture for intelligent power dispatching system, in: Power and Energy Engineering Conference, IEEE, 2010, pp. 1–4.

[9] Z. Haibei, Y. Xu, The architecture design of a distributed workflow system, in: International Symposium on Distributed Computing and Applications to Business, Engineering & Science, IEEE, 2012, pp. 9–12.

[10] B.O. Pinto, A.R. Silva, An architecture for a blended workflow engine, in: Business Process Management Workshops, Springer, 2011, pp. 382–393.

[11] A. Kashlev, S. Lu, A system architecture for running big data workflows in the cloud, in: IEEE International Conference on Services Computing, IEEE, 2014, pp. 51–58.

[12] A. Pradhan, R.K. Joshi, Architecture of a light-weight non-threaded event oriented workflow engine, in: ACM International Conference on Distributed Event-Based Systems, ACM, 2014, pp. 342–345.

[13] B. Kitchenham, Procedures for Performing Systematic Reviews, Keele University, 2004.

[14] S. Keele, Guidelines for Performing Systematic Literature Reviews in Software Engineering, Technical Report, 2007.

[15] B. Kitchenham, R. Pretorius, D. Budgen, O.P. Brereton, M. Turner, M. Niazi, S. Linkman, Systematic literature reviews in software engineering–a tertiary study, Inf. Software Technol. 52 (8) (2010) 792–805.

[16] R. Vanwersch, K. Shahzad, K. Vanhaecht, P. Grefen, L.M. Pintelon, J. Mendling, G.G. van Merode, H. Reijers, Methodological support for business process redesign in health care: a literature review protocol, Int. J. Care Pathways 15 (4) (2011) 119–126.

[17] B. Kitchenham, E. Mendes, G.H. Travassos, Cross versus within-company cost estimation studies: a systematic review, IEEE Trans. Software Eng. 33 (5) (2007) 316–329.

[18] P. Brereton, B. Kitchenham, D. Budgen, M. Turner, M. Khalil, Lessons from applying the systematic literature review process within the software engineering domain, Syst. Software 80 (4) (2007) 571–583.

[19] Y. Levy, T.J. Ellis, A systems approach to conduct an effective literature review in support of information systems research, Inf. Sci.: Int. J.Emerging Trans. discipline 9 (1) (2006) 181–212.

[20] M.M. Ahmed, S. Letchmunan, A systematic literature review on challenges in service oriented software engineering, Int. J. Software Eng. Appl. 9 (6) (2015) 173–186.

[21] A. Fink, Conducting Research Literature Reviews: From the Internet to Paper, Sage Publications, 2013.

[22] A. Kofod-Petersen, How to do a structured literature review in computer science, 2012.

[23] L.I. Meho, The rise and rise of citation analysis, Phys. World 20 (1) (2007) 32.

[24] E.W. Grafarend, Linear and Nonlinear Models: Fixed Effects, Random Effects, and Mixed Models, Walter de Gruyter, 2006.

[25] S. Angelov, P. Grefen, D. Greefhorst, A framework for analysis and design of software reference architectures, Inf. Softw. Technol. 54 (4) (2012) 417–431.

[26] D. Garlan, P. Clements, R. Little, R. Nord, J. Stafford, Documenting Software Architectures: Views and Beyond, Pearson Education, 2010.

[27] P. Grefen, Business Information System Architecture (BISA), Eindhoven University of Technology (TU/e), 2015.

[28] D. Hollingsworth, The Workflow Reference Model, Technical Report, Workflow Management Coalition (TC00-1003), 1995.

[29] A. Arsanjani, L.-J. Zhang, M. Ellis, A. Allam, K. Channabasavaiah, S3: a service-oriented reference architecture, IT Prof. 9 (3) (2007) 10–17.

[30] J. Truijens, Informatie-infrastructuur: een instrument voor het management, Kluwer Bedrijfswetenschappen, 1990. N Dutch).

[31] M.B. Blake, An agent-based cross-organizational workflow architecture in support of web services, in: International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, IEEE, 2002, pp. 176–181.

[32] J. Coplien, D. Hoffman, D. Weiss, Commonality and variability in software engineering, IEEE Software 15 (6) (1998) 37–45.

[33] J. Müller, K. Böhm, The architecture of a secure business-process-management system in service-oriented environments, in: IEEE European Conference on Web Services, IEEE, 2011, pp. 49–56.

[34] Z. Qi, H. Hong, H. Ke-gang, S. Qin-bao, Web service composition modeling architecture based on workflow net, in: International Conference on Software Engineering and Data Mining, IEEE, 2010, pp. 203–206.

[35] G.J. Fakas, B. Karakostas, A peer to peer (P2P) architecture for dynamic workflow management, Inf. Software Technol. 46 (6) (2004) 423–431.

[36] R.N. Taylor, N. Medvidovic, E.M. Dashofy, Software Architecture: Foundations, Theory, and Practice, Wiley Publishing, 2009.

[37] R.T. Fielding, Architectural styles and the design of network-based software architectures, University of California, Irvine, 2000 Ph.D. thesis.

[38] J. Hu, P. Grefen, Conceptual framework and architecture for service mediating workflow management, Inf. Software Technol. 45 (13) (2003) 929–939.

[39] D. Cao, X. Liu, Y. Yang, Novel client-cloud architecture for scalable instance-intensive workflow systems, in: Web Information Systems Engineering (WISE), Springer, 2013, pp. 270–284.

[40] E. Barrett, E. Howley, J. Duggan, A learning architecture for scheduling workflow applications in the cloud, in: IEEE European Conference on Web Services, IEEE, 2011, pp. 83–90.

[41] R.N. Calheiros, R. Ranjan, C.A.F. De Rose, R. Buyya, Cloudsim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services, Technical Report, Grid Computing and Distributed Systems Laboratory, 2009.

[42] G. Borowik, M. Woźniak, A. Fornaia, R. Giunta, C. Napoli, G. Pappalardo, E. Tramontana, A software architecture assisting workflow executions on cloud resources, Int. J. Electron.Telecommun. 61 (1) (2015) 17–23.

[43] P. Bhuyan, A. Ray, D.P. Mohapatra, A service-oriented architecture (SOA) framework component for verification of choreography, in: Smart Innovation, Systems and Technologies, 33, Springer, 2015, pp. 25–35.

[44] S. Kapetanakis, M. Petridis, Evaluating a case-based reasoning architecture for the intelligent monitoring of business workflows, in: Successful Case-based Reasoning Applications-2, Springer, 2014, pp. 43–54.

[45] W. Tan, S. Li, Q. Zhang, S. Chen, A. Tang, X. Hu, Reliable service computing platform architecture for cross-organizational workflows, in: IEEE International Conference on Systems, Man and Cybernetics, IEEE, 2014, pp. 3066–3071.

[46] C. Gargiulo, D. Pirozzi, V. Scarano, An architecture for CFD Workflow management., in: IEEE International Conference on Industrial Informatics, 2013, pp. 352–357.

[47] X. Ye, Z. Li, S. Fu, RNM based decentralized architecture of cross-organizational workflow management system for e-government, in: IEEE International Conference on Software Engineering and Service Science, IEEE, 2013, pp. 60–63.

[48] X. Ma, S. Lu, K. Yang, Service-oriented architecture for SPDFLOW: A healthcare workflow system for sterile processing departments, in: IEEE International Conference on Services Computing, IEEE, 2012, pp. 507–514.

[49] S. Winn, M.T. Oo, Multi-agent model for automation of business process management system based on service oriented architecture, World Acad. Sci. Eng. Technol. 74 (2011) 670–675.

[50] X. Zhao, X. Li, J. Xu, Design of a multiple modeling language supported workflow architecture in educational information system, in: International Conference on Computer Science & Education, IEEE, 2011, pp. 120–125.

[51] G. Zacharewicz, M.E.-A. Hamri, C. Frydman, N. Giambiasi, A generalized discrete event system g-DEVS flattened simulation structure: Application to high-level architecture compliant simulation of workflow, Simulation.

[52] S. Richly, S. Goetz, U. Assmann, S. Schmidt, Role-based multi-purpose workflow engine architecture, International Workshop on Technologies for Context-Aware Business Process Management, 2010.

[53] C. Lin, S. Lu, X. Fei, A. Chebotko, D. Pai, Z. Lai, F. Fotouhi, J. Hua, A reference architecture for scientific workflow management systems and the VIEW SOA solution, IEEE Trans. Serv. Comput. 2 (1) (2009) 79–92.

[54] J. Dang, A. Hedayati, K. Hampel, C. Toklu, An ontological knowledge framework for adaptive medical workflow, J. Biomed. Inform. 41 (5) (2008) 829–836.

[55] J. Jung, I. Choi, M. Song, An integration architecture for knowledge management systems and business process management systems, Comput. Ind. 58 (1) (2007) 21–34.

[56] O. Zimmermann, V. Doubrovski, J. Grundler, K. Hogg, Service-oriented architecture and business process choreography in an order management scenario: rationale, concepts, lessons learned, in: ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, ACM, 2005, pp. 301–312.

[57] P. Senkul, I.H. Toroslu, An architecture for workflow scheduling under resource allocation constraints, Inf. Syst. 30 (5) (2005) 399–422.

[58] J. Yu, R. Buyya, A novel architecture for realizing grid workflow using tuple spaces, in: IEEE/ACM International Workshop on Grid Computing, IEEE, 2004, pp. 119–128.

[59] S. Bassil, R.K. Keller, P. Kropf, A workflow-oriented system architecture for the management of container transportation, in: Business Process Management, Springer, 2004, pp. 116–131.

[60] C. Combi, G. Pozzi, Architectures for a temporal workflow management system, in: Symposium on Applied Computing, ACM, 2004, pp. 659–666.

[61] M.B. Blake, WARP: workflow automation through agent-based reflective processes, in: Conference on Autonomous Agents, 2001, pp. 80–112.

[62] A.-W. Scheer, M. Nüttgens, ARIS Architecture and Reference Models for Business Process Management, Springer, 2000.

[63] C.B. Medeiros, G. Vossen, M. Weske, WASA: A workflow-based architecture to support scientific database applications, in: Workshop on Database and Expert Systems Applications, Springer, 1995, pp. 574–583.

[64] S. Ceri, P. Grefen, G. Sanchez, Wide-a distributed architecture for workflow management, in: Workshop on Research Issues in Data Engineering High Performance Database Management for Large-Scale Applications, in: RIDE '97, IEEE Computer Society, Washington, DC, USA, 1997, pp. 76–80.

[65] G. Alonso, C. Mohan, R. Günthör, D. Agrawal, A. El Abbadi, M. Kamath, Exotica/FMQM: A Persistent Message-Based Architecture for Distributed Workflow Management, Springer, 1995.

[66] H. Schuster, S. Jablonski, T. Kirsche, C. Bussler, A client/server architecture for distributed workflow management systems, in: International Conference on Parallel and Distributed Information Systems, IEEE, 1994, pp. 253–256.

[67] W. van der Aalst, Business process management: a comprehensive survey, ISRN Software Eng. 2013 (2013).

[68] R. Kazman, L. Bass, M. Webb, G. Abowd, SAAM: A method for analyzing the properties of software architectures, in: International Conference on Software Engineering, IEEE Computer Society Press, 1994, pp. 81–90.

[69] R. Kazman, M. Klein, M. Barbacci, T. Longstaff, H. Lipson, J. Carriere, The architecture tradeoff analysis method, in: International Conference on Engineering of Complex Computer Systems, IEEE, 1998, pp. 68–78.

[70] S. Pourmirza, Runtime party switch in an inter-organizational collaboration, Technische Universiteit Eindhoven, 2016 Ph.D. thesis.

[71] W. van der Aalst, M. Netjes, H. Reijers, Supporting the full BPM life-cycle using process mining and intelligent redesign, in: Contemporary Issues in Database Design and Information Systems Devevelopment, 2007, pp. 100–132.

[72] M. Chinosi, A. Trombetta, Bpmn: an introduction to the standard, Comput. Standards Interfaces 34 (1) (2012) 124–134.

[73] M. Weske, Business Process Management: Concepts, Languages, Architectures, Springer, 2012.

[74] P. Grefen, S. Rinderle-Ma, Dynamism in Inter-Organizational Service Orchestration-An Analysis of the State of the Art, Technical Report, BETA Working Paper, 2016.

[75] L.M. Camarinha-Matos, Collaborative networks: A mechanism for enterprise agility and resilience, in: Enterprise interoperability VI, Springer, 2014, pp. 3–11.

[76] S. Pourmirza, R. Dijkman, P. Grefen, Switching parties in a collaboration at run-time, in: IEEE International Conference on Enterprise Computing (EDOC), IEEE, 2014, pp. 136–141.

[77] M. Reichert, B. Weber, Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies, Springer, 2012.

[78] E. Folmer, D. Krukkert, P. Oude Luttighuis, J. van Hillegersberg, Requirements for a Quality Measurement Instrument for Semantic Standards, Verlagshaus Mainz GmbH Aachen., 2010.

[79] R.K. Ramasamy, F.-F. Chua, S.-C. Haw, Web Service Composition Using Windows Workflow for Cloud-based Mobile Application, in: Advanced Computer and Communication Engineering Technology, Springer, 2015, pp. 975–985.