Received: 25 July 2020 | Revised: 2 February 2021 | Accepted: 15 March 2021

DOI: 10.1049/sfw2.12024

REVIEW

Systematic literature review on software architecture of educational websites

Milton Campoverde-Molina[1] | Sergio Luján-Mora[2] | Llorenç Valverde[3]

[1]Unidad Académica de Tecnologías de la Información y la Comunicación (TIC), Universidad Católica de Cuenca, Cuenca, Ecuador

[2]Departamento de Lenguajes y Sistemas Informáticos, Universidad de Alicante, Alicante, Spain

[3]Departament de Matemàtiques i Informàtica, Universitat de les Illes Balears, Palma, Mallorca, Spain

Correspondence

Milton Campoverde-Molina, 1Unidad Académica de Tecnologías de la Información y la Comunicación (TIC), Universidad Católica de Cuenca, 010107, Cuenca, Ecuador.
Email: mcampoverde@ucacue.edu.ec

Funding information

This work was supported by the Catholic University of Cuenca; the EduTech project (609785-EPP-1-2019-1-ES-EPPKA2-CBHE-JP) co-funded by the Erasmus+ Programme of the European Union; and the project "Development of IoT systems for people with disabilities" (PID2019-111196RB-I00) of the Spanish Ministry of Science and Innovation; Ministerio de Ciencia e Innovación, Grant/Award Number: PID2019-111196RB-I00

Abstract

The modern world greatly depends on information systems and the software that governs them. The software architecture defines and designs the holistic structure that the software will have, its components, the interaction between them and all the development is done around it. The purpose of this systematic literature review (SLR) is to analyse the software architectures used in educational websites, methodologies, technological components and empirical results. The search of the SLR yielded 23 studies from the most significant academic sources in education and software engineering. The results of the SLR show that the analysed educational websites were proposing a software architecture, developing a system, proposing a model, assessing of a platform, proposing a Folksonomy-Based Ontology Maintenance, reviewing smart home design, and proposing a Web-based platform to aid parallel, distributed and high-performance computing education. Of the 23 selected studies, 13 carried out an evaluation of their research with either students, teachers, professionals or a combination of these. In conclusion, the selected studies present narrated experiments of projects or individuals that seek to improve collaborative learning in the educational area. Finally, an important finding is that the proposed software architectures do not contemplate laws or quality standards for universal access.

1 | INTRODUCTION

Nowadays, it is impossible to function within the modern world without software, as computer systems constitute the nervous system of an organization. Information systems are therefore indispensable to the functioning of organisations. The software architecture allows to design, project and build computer applications through a systematic and structured process [1]. With the growing demand for software in today's world, developers must build software applications that can evolve over time by considering the software architecture. As technologies evolve, businesses must adapt to these changes, so the design and development of software architectures and computer systems become increasingly complex [2]. To make changes to a software, it is necessary to determine its causes and effects before it is updated [3]. Software architecture has an interdependence between all its stages, and so software development is an activity that requires complete communication and collaboration. In addition, the participation of end users in the design and development of the system is necessary. Software architecture is itself a complex task, and so many dependencies add to its complexity and challenges [4]. Therefore, research in the field of software architecture is a challenge, because research projects are anecdotal test results of individuals or projects that are difficult to compare. This increases the possibility of establishing subjective value judgements [5].

Software architecture is a set of patterns that provide the necessary framework to guide the construction of software, so that programmers, analysts and software developers can share the same line of work and cover all their objectives. The software architecture model represents the logical structure of the software construction process. Conceptually, the architectural elements define and orchestrate the functionality of the software [6]. Software development architectures provide the general forms and guidelines that indicate the structure, operation and interaction between the parts of the software for problem solving.

One of the characteristics of software architecture that is becoming more important is universal access, so designers and systems analysts must take potential users, including people with disabilities, and accessibility standards into account in their developments [7]. The United Nations Convention on the Rights of Persons with Disabilities (CRPD) defined 'access to information and communication technologies, including the Web, as a basic human right' [8]. In the Article 21 – Freedom of Expression and Opinion, and Access to Information, it is stated that governments should urge 'private entities that provide services to the general public, including through the Internet, to provide information and services in accessible and usable formats for persons with disabilities' [8]. The Web [9] is an indispensable resource for people in multiple life activities: health care, education, government, commerce, recreation, employment and others. From the perspective of Akram and Bt Sulaiman [10], "educational institutions should have their own website to publish their content, academic and administrative resources, among others". These sites are available to those interested in education, that is, graduate and prospective students, students' relatives and so on. Also, universities publish their services online for library consultation, grades checking, course registration and more. This means that educational websites should work with accessibility guidelines to provide universal access for people with and without disabilities. In addition, article 24 – Education of the CRPD provides for the integration and participation of all individuals in the education system [8].

The research presented in herein reviewed papers on software architecture of educational websites. This research focuses on the studies that present empirical results, as we want to discover whether software architecture is considered a significant angle in the improvement of programming on educational websites. This systematic literature review (SLR) aims to determine the software architectures used in the selected papers, the methodologies, the technological components and the empirical results. Twenty-three papers were selected to be studied in this SLR, after the application of the search string, the exclusion and inclusion criteria. Applying the SLR methodology in a constructive way to the research process will contribute to the knowledge of software engineering [11]. An SLR is a method in which researchers summarise and analyse available information on a specific topic. In other words, an SLR makes it possible to select, synthesise and evaluate all the scientific evidence published on a subject [12]. An effective SLR establishes a solid base for the development of knowledge: it closes research areas, provides new theoretical knowledge and discovers new areas that need to be investigated [13]. An SLR is a methodological systematic examination of published research results to answer a set of research questions [14].

In Section 2, the SLR methodology and the research questions are presented. In Section 3, the results answer and discuss the research questions defined in the methodology. In Section 4, the discussion makes an analysis and interpretation of the SLR results to determine trends and gaps. In Section 5, the limitations of the SLR are presented, and in Section 6, the conclusions and future work are discussed.

2 | METHODOLOGY OF THE SYSTEMATIC LITERATURE REVIEW

The methodology of the SLR aims to establish a baseline on a topic of interest and search for new research niches. An SLR analyzes primary studies with various discrete activities [15]. This SLR is summarised in three main stages following the Kitchenham's guidelines which are: 'Plan the Review, Conduct the Review and Report the Review' [12].

2.1 | Planning a systematic literature review

The elaboration of a detailed plan or protocol to initiate a systematic review is essential. Therefore, the aim at this stage is to determine the need for an SLR and to develop a review protocol using the results of research from similar studies.

2.1.1 | Identification of the need for a systematic literature review

Considering the following investigations [12, 16–18] as a reference, a search string was set up to find similar SLRs on educational website software architecture and determine whether the proposed SLR would contribute to filling any gaps. The same query string was used in Scopus and the Web of Science (WOS). Below is the final search string that was created after several iterations:

TITLE (['systematic literature review' AND education* AND architectur* AND software] OR ['literature review' AND education* AND architectur* AND software) OR ('systematic review' AND education* AND architectur*])

We only found one SLR [19] that had a relationship with our proposal:

1. In 2013, Shafique [19] conducted a systematic review of the architecture and infrastructure of Pakistan's education information system. This research article determines the structure of an information system and its functionality; it also examines the different technical, social and institutional aspects of the infrastructure of an education information system in Pakistan. In searching the literature, the author consulted scientific databases of published and unpublished information (i.e., books, journal articles, conference/workshop proceedings, reports, dissertations, and theses). The author also examined and cited a few old classical studies in the document. Also, this research made the analysis of some existing information systems in developing and developed countries ($n = 23$). In the results, the selected research highlights the need to differentiate at least the basic dimensions: the social and technological dimensions. The author concludes that the technological dimension refers to the physical infrastructure,

networks and the security of communication and information flows. Finally, the author considers that the infrastructure of an information system for education should take into account the following technical and social aspects: (1) understanding of the background; (2) users and their needs; (3) types of users; (4) types of information needs; (5) coverage; (6) centralised or decentralised; (7) service; (8) organizational considerations; (9) personnel needs; and (10) qualitative considerations.

In summary, the SLR cited above presents a systematic review of the architecture and infrastructure of Pakistan's education information systems. However, this study is not as detailed the one presented herein. For example, there is a lack of information on the software architecture of educational websites, methodologies, technology components, empirical results and objectives proposed in the selected papers. Also, our SLR has been updated to June 2020, whereas [19] is updated until 2013. Therefore, this document does not have the scope of our research questions on educational website software architecture, nor does it have the same degree of detail and precision.

2.1.2 | Development review protocol

The goal herein is to make an analysis of the papers published between January 2009 and June 2020 on the software architecture of educational websites. To do so, it is necessary to develop a review protocol that defines the research questions and scope of the SLR, the search strategy, the inclusion and exclusion criteria and the quality assessment.

Research questions: The research questions establish the orientation about what is going to be investigated in the SLR. To do so, five research questions were established to achieve the purpose of this SLR [20], as presented in Table 1 along with their motivation.

Considering the research questions and their motivations raised in Table 1, the scope of the SLR is established using the PICOC approach of Petticrew and Roberts [21]:

- **Population (P):** Software architectures.
- **Intervention (I):** Education.
- **Comparison (C):** This SLR analyzes the software architecture of educational websites, methodologies, technology components and empirical results, so there is no comparison.
- **Outcomes (O):** Awareness raising in the creation of software architectures.
- **Context (C):** Educational websites.

The research questions were answered by summarising, analysing and discussing the results of the selected papers.

Search Strategy: A search query is intended to be as wide-ranging as possible when searching for information in a scientific database, and should be built around the results we want to obtain. For that reason, in this SLR the search string is created using terms derived from the research questions classified into four scopes presented below: (1) context, web or websites; (2) education, schools or universities; (3) architecture, software architecture or computer architecture; and (4) research type, empirical results of projects, infrastructures, platforms, prototypes, systems, software and developments. The Boolean operator OR was used to connect substitute terms and the Boolean operator AND was used to connect mandatory terms. The wildcard (*) allowed the authors to search for terms written in the singular, plural or words containing these characters and the double quotes found exact phrases. Table 2 shows the search string.

Inclusion and exclusion criteria: The results obtained in an SLR will depend on the selection process of the papers. All papers found with the search string were evaluated to determine whether their studies contribute to this SLR [22]. In addition, all selected papers must meet the inclusion and exclusion criteria as a requirement for consideration herein. This exclusion and inclusion criteria will help to minimise risks and make the results more reliable. The inclusion criteria defined for the selection of papers are:

- I1. A full or short research paper published in a journal.
- I2. The paper presents empirical results.

Papers that meet any of the criteria listed below have been excluded:

- E1. Papers that have been published before 2009. This criterion is based on references [18, 23–25] that have a range of established dates.
- E2. Papers that are not published in journals such as editorials, prefaces, discussions, comments, tutorial summaries, workshop summaries, panels, and so on.
- E3. Papers not written in English.
- E4. Papers without the keyword 'software architecture'.
- E5. Papers that do not analyse the software architecture of educational websites in the abstract.
- E6. Duplicated papers.

Quality assessment: This section considers that it is essential to evaluate the 'quality' of the selected papers, as a complement to the inclusion and exclusion criteria [14]. Therefore, the objective of this quality assessment (QA) consists in assessing and standardising the values of each selected paper after answering a set of quality questions. This will allow finding studies with results that allow answering of the defined research questions about the software architecture of educational websites [12]. For this, a score of 1 is assigned to each QA question that is answered in the affirmative: (1) software architecture is specified in the paper; (2) specifies the type of software architecture used; (3) specifies the method used in educational software architectures; (4) details the technological components used; (5) shows empirical results of software development architecture; (6) the journal is indexed in the SJR (to obtain the quartile score of the selected papers in the SJR, a search of the journal name in the SJR website[1] was carried out);

[1] https://scimagojr.com/

TABLE 1 Research questions

No.	RQ	Motivation
RQ1	What are the objectives proposed in the selected papers?	Identify which are the ends or goal to be achieved in each selected paper.
RQ2	What architecture styles are used to develop educational websites?	Examine the architecture styles that are used to develop educational websites.
RQ3	What software development methodologies are used in software architectures to develop educational websites?	Determine the software development methodologies that are used in educational software architectures.
RQ4	What are the technological components used in the educational software development architecture?	Analyse the technological components used in the educational software development architecture.
RQ5	What are the empirical results obtained with educational software development architectures?	Extract the empirical results obtained on the type of result, the software modules created, the evaluation and limitations of the proposals.

Abbreviation: RQ, research question.

TABLE 2 Query string

Scope	String
Context	(website* OR web* OR "web site*") AND
Education	(education* OR school* OR universit*) AND
Architecture	(architectur* OR "software architecture" OR "computer architecture") AND
Research type	(project OR infrastructure OR platform OR prototype OR system OR software* OR development)

(7) the journal is indexed in JCR (to obtain the quartile score of the selected papers in the JCR, we did a search of the journal name in the website Clarivate's JCR[2]). The questions posed for the quality evaluation can be seen in Table 3, where the minimum value of the score is equal to 0 and the maximum value of the score is equal to 7 (+1 for each question).

2.2 | Conducting a systematic literature review

2.2.1 | Identification of research

An SLR allows through a search process to extract from the literature research on different topics that have been discussed and published. The search process should determine the scientific databases, search resources, keywords and substitutes, and metadata to be searched. In 2019, after conducting research on 26 academic search systems, PubMed and Google Scholar, the authors determined in their conclusions that 'Google Scholar is inappropriate as a primary resource' [26]; for this reason it is not used as a query database in this SLR. Taking into account the above, for this research the following most significant academic sources in education and software engineering were selected: Scopus, ScienceDirect, WOS and IEEE Xplore Digital Library. The authors estimate that these databases are sufficient and that there are even duplicates in the results, that is the same article appears in several databases at the same time, which is proof that their coverage is very high. The selection of databases was made taking into account the following criteria:

- It indexes high-quality papers that are peer-reviewed.
- It contains the journals that publish scientific articles indexed in SJR and JCR.
- It allows advanced searches using query strings composed of keywords, Boolean operators, double quotes, wildcard, among others.
- It allows the search in several metadata fields.

2.2.2 | Selection of studies

The search and selection process for the papers was carried out in June 2020. With the query string defined in Table 2, a total of 9561 papers were found. After applying E1, 3516 papers were excluded from the initial 9561 papers because they were published before 2009. After applying E2, 4691 papers were excluded because they were not published in journals. After applying E3, 82 papers were excluded because they were not written in English. After applying E4, 1125 papers were excluded from the remaining 1272 papers because they did not have the keyword 'software architecture'. After applying E5, where the full texts were examined, 117 papers were excluded from the remaining 147 papers. Finally, after applying E6, 23 papers were selected from the remaining 30 papers because seven papers were duplicated. Many papers were excluded because their research did not consider software architecture on educational websites, for example papers that publish research on: digital libraries, games, software approaches, quality requirements for software development, application

[2] https://clarivate.com/webofsciencegroup/solutions/journal-citation-reports/

TABLE 3 QA checklist [20]

No.	QA Question	Expected Results
QA1	Is the software architecture specified in the paper?	Yes(+1)/No(+0)
QA2	Is the type of software architecture used specified in the paper?	Yes(+1)/No(+0)
QA3	Is the software methodology used specified in the paper?	Yes(+1)/No(+0)
QA4	Are the technological components used detailed in the paper?	Yes(+1)/No(+0)
QA5	Is the software architecture detailed in the empirical results of the paper?	Yes(+1)/No(+0)
QA6	Is the paper in a journal indexed in SJR?	(+1) if the paper is indexed in a Q1 journal, (+0.75) if the paper is indexed in a Q2 journal, (+0.5) if the paper is indexed in a Q3 journal, (+0.25) if the paper is indexed in a Q4 journal, (+0) if it is not in the ranking
QA7	Is the paper in a journal indexed in JCR?	(+1) if the paper is indexed in a Q1 journal, (+0.75) if the paper is indexed in a Q2 journal, (+0.5) if the paper is indexed in a Q3 journal, (+0.25) if the paper is indexed in a Q4 journal, (+0) if it is not in the ranking

Abbreviation: QA, quality assessment.

development, software testing and verification, and so on that are not the focus of this SLR. In Figure 1 the inclusion/exclusion diagram of the papers is illustrated.

2.2.3 | Quality assessment of studies

The QA allows to determine if the selected papers meet the parameters that will help to achieve the objective of the SLR [27]. The QA was carried out by applying the checklist presented in Table 3. The values obtained have been summed and normalised to select the papers that have a score higher than 0.70 to ensure the contribution of each paper to the SLR.

2.3 | Reporting a systematic literature review

This stage aims to respond to the questions presented in Table 1. The following section presents the literature review where RQ1, RQ2, RQ3, RQ4 and RQ5 are answered, making a synthesis and interpretation of the data and results of the selected studies.

3 | RESULTS

In this section, the results from the QA of the selected papers are first presented. Secondly, each research question is answered by summarising and interpreting and discussing the results of the selected papers.

The QA results for each of the selected papers can be seen in Table 4. A standardisation column is included to determine the final compliance value of each paper's quality questions between 0 and 1. To this end, the minimum–maximum normalisation formula [50] has been used, which calculates the final values on a scale of 0 to 1. Below is the formula used to calculate the value of this column (1):

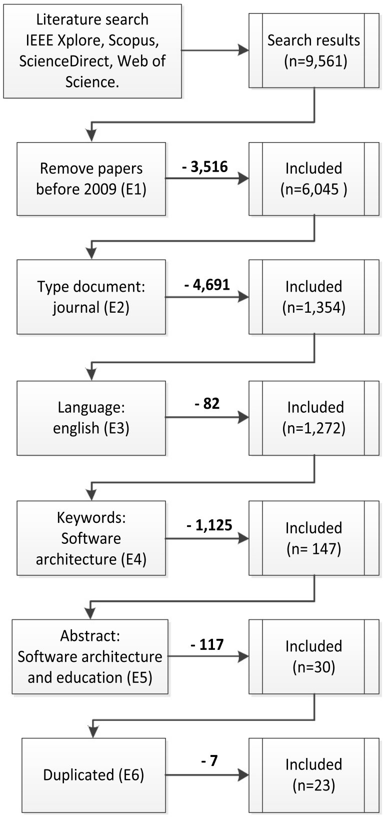
Literature search IEEE Xplore, Scopus, ScienceDirect, Web of Science.
Search results (n=9,561)
Remove papers before 2009 (E1)
- 3,516
Included (n=6,045)
Type document: journal (E2)
- 4,691
Included (n=1,354)
Language: english (E3)
- 82
Included (n=1,272)
Keywords: Software architecture (E4)
- 1,125
Included (n= 147)
Abstract: Software architecture and education (E5)
- 117
Included (n=30)
Duplicated (E6)
- 7
Included (n=23)

FIGURE 1 Flow diagram of the process of inclusion and exclusion of papers

$$Normalization = \frac{Score - min(Score)}{[max(Score) - min(Score)]} \quad (1)$$

TABLE 4 Results of the assessment quality of the selected papers

	Publication		QA								
Paper	Year	Name	QA1	QA2	QA3	QA4	QA5	QA6	QA7	Score	Normalisation
[28]	2009	IEEE Transactions on Knowledge and Data Engineering (TKDE)	1	1	1	1	1	1.00	1.00	7.00	1.00
[29]	2010	Journal of Systems and Software (JSS)	1	1	1	1	1	1.00	0.75	6.75	0.96
[30]	2011	Robotics and Autonomous Systems (RAS)	1	1	1	1	1	1.00	0.50	6.50	0.92
[31]	2011	Expert Systems with Applications (ESA)	1	1	1	1	1	1.00	1.00	7.00	1.00
[32]	2011	IEEE Transactions on Learning Technologies (TLT)	1	1	1	1	1	0.75	0.50	6.25	0.89
[21]	2011	Journal of Pain and Symptom Management (JPSM)	1	1	1	1	1	1.00	1.00	7.00	1.00
[33]	2012	International Journal of Sustainability in Higher Education (IJSHE)	1	1	1	1	1	0.75	0.75	6.50	0.92
[34]	2014	Journal of Visual Languages and Computing (JVLC)	1	1	1	1	1	0.75	0.50	6.25	0.89
[35]	2015	IEEE Transactions on Learning Technologies (TLT)	1	1	1	1	1	0.75	0.50	6.25	0.89
[36]	2015	International Journal of Emerging Technologies in Learning (iJET)	1	1	1	1	1	0.50	0.00	5.50	0.78
[37]	2016	Environmental Modelling and Software (EMS)	1	1	1	1	1	1.00	1.00	7.00	1.00
[38]	2016	Science of Computer Programming (SCP)	1	1	1	1	1	0.75	0.50	6.25	0.89
[39]	2017	IEEE Transactions on Learning Technologies (TLT)	1	1	1	1	1	0.75	0.50	6.25	0.89
[40]	2017	Journal of Systems and Software (JSS)	1	1	1	1	0	1.00	1.00	6.00	0.85
[41]	2017	TEM Journal (TEM)	1	1	1	1	1	0.00	0.00	5.00	0.71
[42]	2017	International Journal of Software Engineering and Knowledge Engineering (IJSEKE)	1	1	1	1	1	0.25	0.25	5.50	0.78
[43]	2017	Educational Technology and Society (ETS)	1	1	1	1	1	1.00	0.75	6.75	0.96
[44]	2017	Computers in Human Behavior (CHB)	1	1	1	1	1	1.00	1.00	7.00	1.00
[45]	2017	International Journal of Emerging Technologies in Learning (iJET)	1	1	1	1	1	0.50	0.00	5.50	0.78
[46]	2018	JournalofParallel and Distributed Computing (JPDC)	1	1	1	1	1	0.75	0.75	6.50	0.92
[47]	2019	International Journal of Emerging Technologies in Learning (iJET)	1	1	1	1	1	0.50	0.00	5.50	0.78
[48]	2019	International Journal of Information and Learning Technology (IJILT)	1	1	1	1	1	0.75	0.00	5.75	0.82
[49]	2019	IEEE Access	1	1	1	1	1	1.00	1.00	7.00	1.00

where the min(Score) has a value of 0, the max(Score) has a value of 7 and the Score takes the value of the sum of the QA values of each paper.

3.1 | RQ1. What are the objectives proposed in the selected papers?

This research question is divided into two main points: word cloud and purpose of objectives. Word clouds, also known as tag clouds, are the visual representation of the most important words that make up a text. Figure 2 shows the 30 most important words of the objectives of the selected papers using TagCrowd[3] and the number indicates the frequency of occurrence of each word.

It can be seen that the words 'architecture' and 'software' are those that stand out the most in the word cloud.

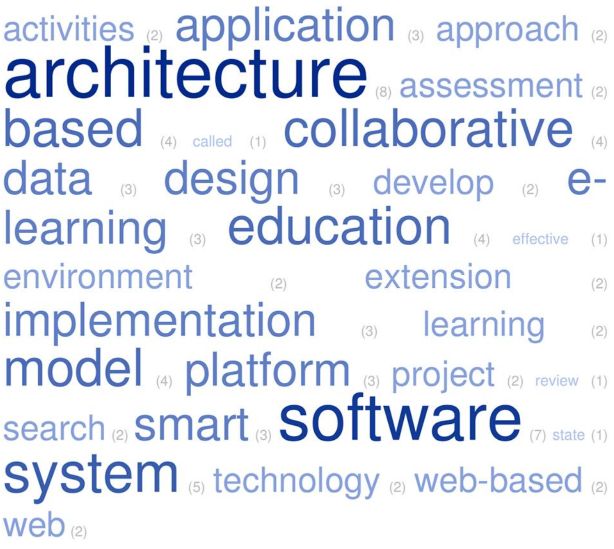

FIGURE 2 The word cloud of the objectives of the selected papers

[3] https://tagcrowd.com/

TABLE 5 Summary of the objectives of the selected papers

Paper	Objectives	Abbreviation Objectives
[28]	Build a prototype software referred to as Open Smart Classroom based totally at the multiagent system architecture.	OBJE1
[29]	Provide a software infrastructure for the systematised and effective construction of complex collaborative learning systems.	OBJE2
[30]	Introduce an interactive Java software platform that allows users to easily create advanced robotic applications.	OBJE3
[31]	Develop an educational Web-based e-testing system.	OBJE4
[32]	Present an ontology maintenance method based on the use of collaborative labels.	OBJE5
[21]	Develop a Web-based tool to collect symptoms, needed data and provide feedback to hospice and palliative care patients.	OBJE6
[33]	Provide a modular, extensible software architecture for an EMS.	OBJE7
[34]	Describe a software architecture for creating engaging linear narrations that can be shared on the Web.	OBJE8
[35]	Propose a hardware and software architecture for implementation of remote laboratories for automatic control.	OBJE9
[36]	Present a domain model and component model of an e-learning system.	OBJE10
[37]	Present the software architecture and implementation of SWATShare as a collaborative environment for hydrology research and education.	OBJE11
[38]	Present the SDE a search engine for educational resources that was built on top of the knowledge provided by Wikipedia.	OBJE12
[39]	Present evaluation data regarding the MAPIS3 architecture.	OBJE13
[40]	Present an approach to the introduction of software architecting activities in an agile project course.	OBJE14
[41]	Review the state of the art of service-oriented technologies in smart home design.	OBJE15
[42]	Present E-WAE	OBJE16
[43]	Present a model for content enrichment in Ubiquitous Learning Environments.	OBJE17
[44]	Propose and implement frameworks for a smart e-learning ecosystem.	OBJE18
[45]	Propose a PSM based on n-tier architecture.	OBJE19
[46]	Summarize Let's HPC project.	OBJE20
[47]	Design of a platform for the evaluation of physical education using data mining to help teachers provide individualised instruction.	OBJE21
[48]	Support the assessment of collaborative activities in e-learning: a design founded on the IMS-LD meta-model.	OBJE22
[49]	Propose a new mixed hardware–software architecture.	OBJE23

Abbreviations: EMS, environmental management system; E-WAE, Enterprise Web Application Extension; PSM, platform-specific model; SDE, search discover explore.

Furthermore, this result confirms that the selection of papers is correct, there are no strange words in the cloud and the most frequent ones are those expected.

In the second point, the objectives of the selected papers are summarised with their respective abbreviations, as can be seen in Table 5 (see Table 6 in the Appendix A for full details). Then, the selected papers are grouped according to their individual purposes. Table 7 shows this classification, where it can be seen that the largest number of papers have the purpose of proposing a software architecture and developing a system.

3.2 | RQ2. What architecture styles are used to develop educational websites?

Most of the reviewed studies describe software architectures for educational websites. Software architectures use architectural styles which are categorised into communication [service-oriented architecture (SOA), message bus], deployment (client/server, N-tier, 3-tier), domain (domain-driven design) and structure (component-based, object-oriented, layered architecture) [51–53]. The selected papers were grouped according to five architectural styles as shown in Table 8 (see Table 6 in

TABLE 6 Purpose of the objectives of the selected papers

Purpose	Objectives	Number of Papers
Propose a software architecture	OBJE2, OBJE7, OBJE8, OBJE9, OBJE11, OBJE13, OBJE14, OBJE23	8
Develop a system	OBJE1, OBJE3, OBJE4, OBJE6, OBJE12, OBJE16, OBJE18	7
Propose a model	OBJE10, OBJE17, OBJE19	3
Assess a platform	OBJE21, OBJE22	2
Propose a Folksonomy-Based Pntology Maintenance	OBJE5	1
Review smart home design	OBJE15	1
Summarize Let's HPC project	OBJE20	1

TABLE 7 Architecture styles of the selected papers

Architecture Styles	Frequency	Percentage	Papers
Layered architecture	10	45.45 %	[21, 30, 33, 36, 38, 39, 44, 47–49]
Client-server architecture	6	27.27 %	[31, 32, 34, 35, 37, 46]
Service-oriented architecture (SOA)	4	18.18 %	[28, 29, 41, 42]
N-tier/3-tier	1	4.55 %	[45]
Component-based architecture	1	4.55 %	[43]

Abbreviation: SOA, service-oriented architecture.

the Appendix for full details). Of the 23 papers selected, 45.45% use a layered architecture, 27.27% use a client-server architecture, 18.18% use an SOA, 4.55% N-tier/3-tier and 4.55% component-based architecture. However, paper [4] does not present a defined architecture as it is aimed at teaching software architectures in agile projects in education. This approach is aimed at second-year, applied studies, software engineering students. Also, the authors state that it can be used in higher years with modifications (e.g., in terms of cases and more formal architecture evaluations).

In summary, architectural styles are a set of principles that guide the design of software development in an organised and goal-focussed manner. The choice of architectural styles depends on the requirements, type of application, constraints, scenarios, and patterns and practices that software architects, developers, programmers, and so on are most familiar and comfortable with [51].

3.3 | RQ3. What software development methodologies are used in software architectures to develop educational websites?

Table 9 shows the software development methodologies of the selected papers. The selected studies have been classified according to the software development methodologies and characteristics exposed in the article by Despa [54]. In addition, the keywords of the selected papers for this classification have been taken into account, which are detailed below by methodology and paper (see Table 6 in the Appendix for full details):

1. Dynamic systems development method
    - Paper [30] Modelling, robot simulation, robotics education, visualization tools.
    - Paper [35] Architectures, industrial hardware, PLC laboratories, process control, remote laboratories.
    - Paper [41] Architecture, security, smart home, standards, Web services.
    - Paper [46] HPC education, HPC database, multicore architecture, parallel & distributed programming, performance analyser.
    - Paper [49] Architecture, embedded system, online experimentation, remote laboratory, scalability.
2. Model-driven engineering
    - Paper [29] Component-based software engineering, e-learning, computer-supported collaborative learning, model-driven engineering, service orientation, software and systems education, software architecture and design, SOA, software reuse, software engineering methods.
    - Paper [32] Collaborative learning, ontology design, computer uses in education, applications and expert knowledge-intensive systems.
    - Paper [34] Annotation, education ontology, storytelling, Tag, Web3D.
    - Paper [36] Component, framework, software architecture, Web service.
    - Paper [38] Bag-of-concepts (BoC) representation, exploratory search, information retrieval, software architecture.
    - Paper [42] Enterprise applications, mockup, multitier architecture, MDA, MDD, UML.
    - Paper [44] e-Learning ecosystem, learner model, ontology, personalisation, software architecture, Semantic Web Rule Language.
    - Paper [45] e-Learning, MDA, n-tiers, QVT.
    - Paper [47] Data analysis, decision tree, physical education performance, Web.

TABLE 8 Software development methodologies of the selected papers

Methodology	Frequency	Percentage	Papers
Model-driven engineering	9	39.12 %	[29, 32, 34, 36, 38, 42, 44, 45, 47]
Dynamic systems development method	5	21.74 %	[30, 35, 41, 46, 49]
Prototyping	5	21.74 %	[28, 31, 33, 37, 43]
SCRUM	2	8.70 %	[21, 40]
Rapid application development	1	4.35 %	[48]
Test-driven development	1	4.35 %	[39]

Abbreviation: MDE, model-driven engineering.

3. Prototyping
   - Paper [28] Computer uses in education, pervasive computing, Web-based services.
   - Paper [31] Educational information system, interactive learning environments, self-testing knowledge, software architecture, Web-based system.
   - Paper [33] Environmental management, computer software, environmental management systems, energy dashboard, facilities, universities, university sustainability, university operations, university facilities, United States of America.
   - Paper [37] Cyberinfrastructure, hydrology, SWAT, SWATShare, WaterHUB, XSEDE.
   - Paper [43] Social and collaborative learning, ubiquitous learning environments, educational content enrichment.
4. Rapid application development
   - Paper [48] Assessment, assessment strategies, collaborative activities, e-learning IMS-LD, WfMS/LMS platform.
5. SCRUM
   - Paper [21] Flexible Web technology, health science, hospice, patient-reported outcomes, palliative care, software architecture.
   - Paper [40] Agile, course, project, software architecture, SCRUM, software engineering education, students, teaching.
6. Test-driven development
   - Paper [39] computer-supported collaborative learning (CSCL), IMS-LD, software tool orchestration, software architecture evaluation, Web services.

Of the 23 papers selected, 39.12% use a model-driven engineering (MDE), 21.74% use a dynamic systems development method, 21.74% use prototyping, 8.70% use SCRUM, 4.35% use rapid application development and 4.35% use test-driven development, as shown in Table 9.

3.4 | RQ4. What are the technological components used in the educational software development architecture?

Web services are the most commonly used component to exchange data between applications in conjunction with protocols and standards. The technological components have been classified into tools, programs, databases, systems and laboratories. The classification of the technological components used in the selected papers is shown below (see Table 6 in the Appendix for full details):

Programs. PHP, Java, C/C++, Matlab, LOCO-Analyst, ExpressJS, Angular, Ruby, ToBoA-3D, GeoServer, Globus, Python, Protégé, data mining technology, Node, ASP.NET, ADO.NET, ProcessMaker.

Database. WfMS database, MongoDB, MYSQL, Microsoft SQL Server 2005.

Systems. Health Science Process Framework (HSPF), Clinical Research Management System (CRMS).

Laboratories. Laboratory WebServer, Remote Laboratory Management System (RLMS), Interactive Live-streaming Platform.

Tools. Tomcat, Axis, Web Services (WSDL, SOAP), HTTP, XML, UML, Easy Java Simulations (EJS), object-oriented Java library (EjsRL), HTTPS, HTML, XHTML, DHTML, JavaScript, triggers, e-mail, CSS, Apache, AJAX, Google Web Toolkit (GWT) SDK, UDDI protocol, XML data bus, Black-box framework, CORBA, EJB, COM+, Flex software development kit, Dublin Core, Nokogiri library, Geocoder library, Google Maps API, DOM, Capybara library, Apache Solr, Mentorchat, Google visualization API, Google forms tool, Devices Profile for Web Services (DPWS), Java Server Faces (JSF), X3D, Tag cloud, Model-View-Controller (ASP.NET MVC) frameworks, WAE4x, JSON, Spring MVC, Spring IOC, Hibernate.

3.5 | RQ5. What are the empirical results obtained with educational software development architectures?

This section extracts the empirical results obtained from each of the selected papers, such as the type of result, the software modules created, and their evaluation and limitations. The results of each of the selected papers are detailed below (see Table 10 in the Appendix for full details):

1. The result of paper [28] is a prototype of the Open Smart Classroom and contains three modules: Open Smart Platform Gateway (OSPG), Smart Platform Agent Webservice (SPAW) and Web Service Wrapper Agent (WSWA). This prototype has been evaluated by seven students from

Tsinghua and five students from Kyoto who participated in the experiment. The experiment lasted 140 min. However, the time was adequate for the application of real-time transmission due to the limitation of the HTTP protocol, the reverse translation is a high load assignment, and Langrid does not yet have a proper load setting device.

2. The result of paper [29] is a software infrastructure called Collaborative Learning Purpose Library (CLPL), based on this platform, a prototype Web application called discussion forum (DF) was created. In addition, it has been evaluated by master's students from 10 courses at the Open University of Catalonia who developed complex software applications in the domain of Computer-Supported Collaborative Learning (CSCL) demonstrating that high-quality CSCL applications can be developed.
3. The result of paper [30] is an interactive tool for computer vision, industrial robots simulation and remote operation. The highest level API of the EjsRL defines four important modules: Matrix Computation, Robotics, Remote Operation and Computer Vision. In addition, the bookstore also integrates import and export functions in different file formats, so users can save and restore their designs. To test the functionality of the system, the system has been used in various robotics courses since 2009 as a teaching tool for university students, with positive acceptance.
4. The result of paper [31] is a conceptual approach for the development of educational Web-based e-testing system. A prototype e-testing system has been developed, using the HTML and JavaScript languages to illustrate the proposed approach. The prototype of the e-testing system has five modules: password, choice, shuffle and random, checking and scoring, and evaluation. The prototype was self-examined and evaluated by the students of the course 'Programming Language C' through an official examination. The application of the proposed Web-based e-testing tool approach is very useful to promote the understanding and application of teachers' attitudes considering the experimental results obtained.
5. The result of paper [32] is an approach to Folksonomy-Based Ontology Maintenance for learning environments. This approach was evaluated through a questionnaire to participants from a private company based in Canada that develops and provides technology and content for vocational training, Athabasca University, University of Belgrade and Simon Fraser University. In total, 22 people (17 men and 5 women) participated. With respect to the 'visualization and interactive interface for ontology maintenance' the experiments revealed a very high value of perception by the educators involved. The best performance metric is the PMI-Gwikipedia of the nWMSR in relation to the experiences of the different metrics \enleadertwodots
6. The result of paper [21] is a Web tool that standardises, collects and systematically uses data to facilitate communication between patients, families and hospice and palliative care providers (Tell Us^{TM}). Tell Us^{TM} includes modules to create and display clinical consultations and completion schedules, register clinical and patient sites. In addition, it allows patients and/or families to complete assigned assessments and providers to view patient-reported data. Also, Tell Us^{TM} gives the provider automatic, personalised e-mail alerts based on patient responses (e.g., uncontrolled symptoms or need for medication) and also provides educational materials tailored to patient needs.
7. The result of paper [33] is the development of an environmental management system (EMS) on the modern university campus. The authors designed a flexible and modular software architecture for a variety of electronic information and a powerful and secure research structure. Environmental data were identified, collected and entered into the evolving system such as transport services, utility usages and waste generation. The EMS system has three modules: real-time data, manually collected data and spatial data types. The system is adaptable to new types of environmental data, and allows for manual and automated data entry, customised 'data entry at source' mechanisms, and uses flexible tools to analyse captured environmental data and present them visually.
8. The result of paper [34] is creating Web3D educational stories from crowd-sourced annotations. To this end, a novel software architecture is proposed to create attractive stories that can be shared on the Web. Three-dimensional (3D) representation and storytelling are two amazing methods for teaching students while connecting with them.
9. The result of paper [35] is a configurable and flexible architecture for remote automatic control laboratories. Multipurpose hardware and software architecture (MHSA) is proposed. Four systems were implemented with this architecture: thermo-optical plant, DC motor, hydraulic system with coupled tanks and air heat exchanger. The authors conducted the evaluation with 32 students who participated in various courses (Integrated Control in Process Engineering, Process Control, Process Modelling and System Identification). The main limitation of MHSA is that it can only be used to develop laboratories that are designed with PLCs.
10. The result of paper [36] has been to design an e-learning-oriented software architecture. The e-learning software architecture was developed according to the client's requirements and implemented into a Web-based learning website. This software includes all the functions, tests, ask-answer, exercises, including the management of learning content, cooperation, user management, memos and so on. The software architecture according to the research shows that it has achieved positive results.
11. The results of paper [37] are the architecture of the SWATShare software, the functional capabilities and its application as a collaborative environment for research and education in hydrology. Three case studies demonstrate the usefulness of SWATShare for collaborative research and education: case study 1 – educational application through model sharing (19 students in CE-54900:

Computational Watershed Hydrology course); case study 2 – collaborative research through model sharing (a model created by user A is extended and/or reused by users B and C); case study 3 – simultaneous simulations of multiple SWAT models (multiple SWAT models). The authors demonstrated the potential of SWATShare as a research tool, a collaborative platform and an educational tool through three case studies. In addition, it is stipulated that SWATShare can be used as a decision support system for policy makers and river basin managers.

12. The result of paper [38] is an architecture of a concept-based information retrieval system for educational resources. The information retrieval software architecture has several software components that cooperate with each other: a semantic annotation module, an information extraction module, an application module and an indexing module. Both the construction of the SDE (Search DiscoverExplore), an exploratory Web crawler of educational resources and its evaluation was developed with the EU FP7 iTEC project, in several locations in Europe with a total of 65 members [21 Oulu (Finland), nine Lisbon (Portugal), 9 Budapest (Hungary), six Vigo (Spain), 10 Bad Hofgastein (Austria) and 10 Vilnus (Lithuania)]. The authors in the results after the final evaluation of the concept-based search engine demonstrated through workshops that they are very positive. The main limitation of the study is the evaluation process. The evaluation protocol has not been fully formalised and the method used is neither formal nor systematic. Another limitation of the proposed method is that as users navigate and tag resources, user interests accrue over time.
13. The result of paper [39] is an evaluation of the MAPIS3 architecture which is proposed as a solution to transfer data between various tools and promote a flexible collaborative learning design. The evaluation of the MAPIS3 architecture is carried out with three case studies: case study 1: visualization of the MAPIS3-based LD forum, 13 graduate students participated, duration 1 week; case study 2: visualization of the forum, 83 adult high school students, duration 6 weeks; case study 3: orchestration of chat-forum tools, 39 students, duration 1 week. The limitations of the study are as follows: (a) the volume of stakeholders is small and they play the role of teacher; (b) the background of the students is relatively limited; (c) the method of data analysis, the percentage trend provided by this method is not absolute, it is worth further research; and (d) there are few interconnected instruments in the case studies.
14. The result of paper [40] is a software architecture approach to agile projects in education. The approach is designed for second-year software engineering students in applied studies. This approach allows the development of software architecture activities in agile project courses. The approach has been applied in two consecutive course runs. There is a better understanding of the value of architectural activities and a greater appreciation by students of the combination of architectural activities and agile development. With respect to internal validity, the authors state that there are many variables that limit the conclusions, which require much attention (historical bias, teacher bias, selected case, agile theory bias, group composition, group context) and with respect to external validity of the experiment (cultural bias, educational context, age).
15. The result of paper [41] is a review of the state-of-the-art of service-oriented technologies in smart home design. It provides the educational aspect of service orientation in the implementation and design stage of smart homes, which is a fundamental unit for the smart city. For data exchange and integration of units with reuse the service-oriented software architecture is studied. The standards used in unit design for a smart home like 'The Open Gateway Services Initiative' (OSGi) and 'Universal Plug n Play' (UPnP) are reviewed. Discoveries, events on resource-constrained devices and secure Web service messaging are reviewed. The main limitation of the UPnP is that to access the service one must be connected to the same local area network, another limitation is that to establish the connection a person is required.
16. The result of paper [42] is an Enterprise Web Application Extension (E-WAE). WAE4x profiles have been defined that provide code-oriented semantics for E-WAE models. Thus, WAE4x Platform-Specific Models (PSMs) can be used to automatically generate JSF/ASP.NET MVC code from them. E-WAE has been used with different applications. The purchase process on the Amazon website, the use of disposal resources in OdAJ2EE, an educational application developed by the authors, and the search service of the United States Library of Congress Online Catalog (17 million records) are examples of its applicability.
17. The result of paper [43] is a loosely coupled software architecture. The model proposed by the authors has been applied and integrated into the study room experience, using a multimedia capture platform for instructional scenarios. The platform contains two modules: one to search and store system information and the other to make service requests and display graphic components. The authors analyse the receptivity, impact and proposed features by monitoring the use of the platform during three school semesters by 121 university students. As a result, it was observed that both the system access rate and student performance improved, indicating that the interactive function took advantage of the collaborative learning interaction and promoted the teaching process. The results cannot be generalised because they may be biased by the mastery of the courses (Computer Science and Information Systems).
18. The result of paper [44] is a paradigm for a smart learning environment based on the semantic Web. It proposes and implements a framework for the intelligent learning ecosystem using the ontology and the Semantic Web Rule Language (SWRL). The four ontologies used are: a student model ontology, a teaching methods ontology, a learning object ontology and a learning activities ontology.

19. The result of paper [45] is a model-driven architecture (MDA) approach for developing an e-learning platform respecting the n-tiers architecture by a series of transformations starting from the coded in QVT (Query View Transformation Language), the PSM (Platform Specific Model) to the PIM (Platform Independent Model). The course management system for students has been considered an example of the development of an e-learning platform with the MDA approach.
20. The result of paper [46] is a summary of the Let's HPC project. Let's HPC is an open-access Web-based platform that complements traditional education in high-performance computing (HPC) and parallel and distributed computing (PDC). The purpose of the development of the platform has been to allow users to learn, evaluate, teach and observe the performance of parallel algorithms from a system perspective. The platform also allows students to prepare standard lab/project reports to help teachers perform unified assessments. At the end of the 2017 fall semester, the authors conducted a comprehensive survey of 53 students who had used the platform during the CS301 course to collect information on assignments and projects. Survey results showed that students learn best using the HPC platform.
21. The result of paper [47] is a physical education assessment platform that has been designed using data mining technology and is intended to address issues such as: the large workload on the actual sports performance test of school students and to help teachers provide individualised instruction. This platform adopts the ASP.NET technology and employs three-layer architecture. The platform included five modules: modules for examination type management function, examination elements and grades, a module for statistical analysis and calculation of grades, and a module for platform administration.
22. The result of paper [48] is the proposed extension of an IMS-LD meta-model in order to support the assessment of collaborative activities in e-learning. An infrastructure has been proposed to support the extension of the IMS-LD meta-model, this infrastructure is based on an architecture composed mainly of a WfMS/LMS plus the proposed application. The evaluation is done through a case study: summative assessment (a month with learners' groups in the third year of licence, computer science); in addition, a survey is used to collect learners' opinions. To analyse performance under large-scale use, more experiments are needed to check the effectiveness of the proposed software.
23. The result of paper [49] was to create a remote laboratory architecture focussed on embedded structure experiments. A structure based mainly on Redis is proposed with a modular layout and a technology that allows sharing the hardware to meet the combined requirements of high scalability and efficiency (costs vs. benefits). This combined hardware and software architecture has laid the foundations for the development of remote laboratories focussed on the experimentation of structures based on microcontrollers and embedded devices. The structure and remote laboratory have been compared with other contemporary remote laboratories and their architectures. In addition, a technical evaluation, a price effectiveness evaluation and a scalability evaluation are carried out.

4 | DISCUSSION

In the current study, we conducted an SLR on an educational website software development architecture, architecture styles, methodologies, technology components and their empirical results. The research papers analysed are narrated experiments of projects, infrastructures, platforms, prototypes, systems that seek to improve collaborative learning in the educational area. The selected papers use five styles of development architectures and six development methodologies according to the classification made by the authors and a wide variety of technological components.

With the bibliometric information extracted from the selected articles, an analysis was made to determine the evolution and interest of the research topic. In addition, it was possible to evaluate scientific activity and the impact of both research and sources. The data gathered reveal that the selected papers were published in 18 journals. The largest number of publications occurred in 2017 with seven papers; four papers were published in 2011; three papers in 2019; two papers in 2015 and 2016; and one paper in 2009, 2010, 2012, 2014 and 2018. Most of the selected research has been published in journals that are indexed in the WOS database collection. The countries that contribute most to the topic of software architecture are Spain, China and United States. The publication sources of the selected papers are four papers indexed in SJR journals, 18 papers indexed in SJR and JCR journals and one paper whose journal is not indexed in either SJR or JCR. In SJR, 10 papers were indexed in Q1 journals, seven papers were indexed in Q2 journals, four papers were indexed in Q3 journals and one paper was indexed in Q4 journals. In JCR, seven papers were published in journals indexed in Q1, four papers in journals indexed in Q2, six papers in journals indexed in Q3 and one paper in a journal indexed in Q4. At the time of writing, the SJR and JCR 2020 quartiles have not been released. Therefore, this SLR uses the 2019 quartiles for papers published in 2020 (see Table 11 in the Appendix for full details).

The SLR made it possible to identify, evaluate, interpret and synthesise 23 research works in educational website software development architecture. All the objectives of the papers are focussed on the educational context. Of the 23 analysed papers in this SLR, eight papers propose a software architecture; seven papers develop a system; two papers evaluate a platform; one paper proposes an ontology based on the Folksonomy maintenance; one paper reviews the design of a smart house; and one paper summarises the HPC project. Of the 23 research works, 13 were validated with students, professors and professionals.

An important finding found in the selected papers is that the 'layered architecture' is the most widely used for the

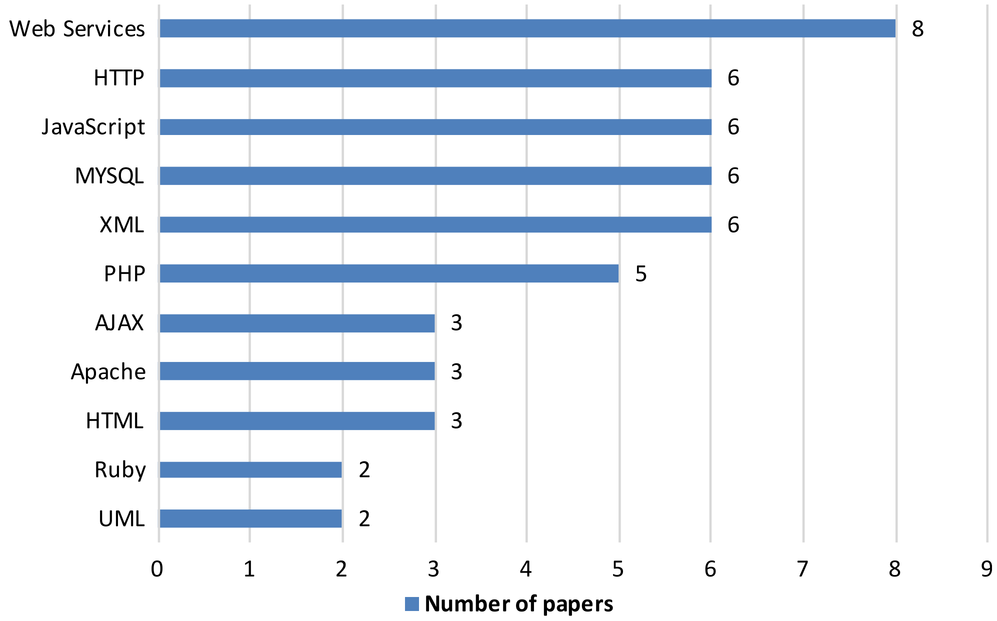

FIGURE 3 Technological components most commonly used in the development of educational website software

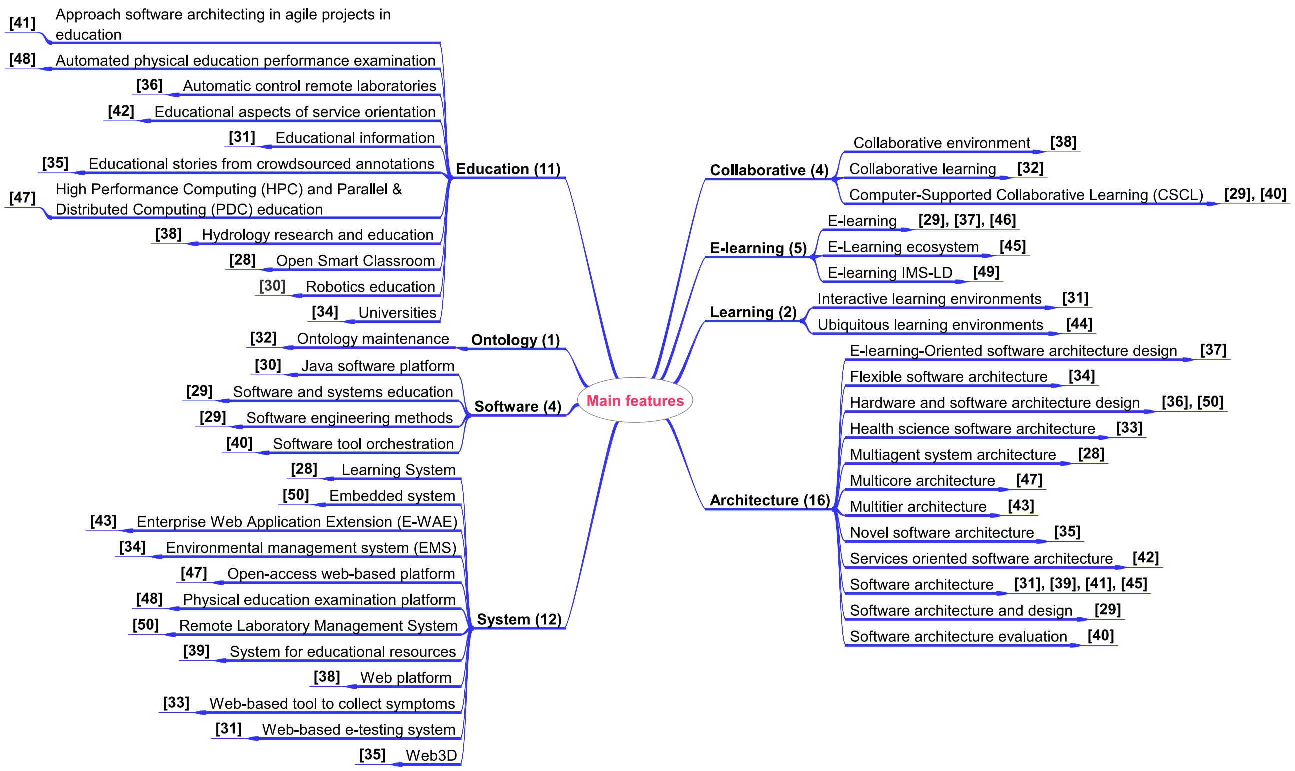
Main features
Education (11)
[41] Approach software architecting in agile projects in education
[48] Automated physical education performance examination
[36] Automatic control remote laboratories
[42] Educational aspects of service orientation
[31] Educational information
[35] Educational stories from crowdsourced annotations
[47] High Performance Computing (HPC) and Parallel & Distributed Computing (PDC) education
[38] Hydrology research and education
[28] Open Smart Classroom
[30] Robotics education
[34] Universities
Ontology (1)
[32] Ontology maintenance
Software (4)
[30] Java software platform
[29] Software and systems education
[29] Software engineering methods
[40] Software tool orchestration
System (12)
[28] Learning System
[50] Embedded system
[43] Enterprise Web Application Extension (E-WAE)
[34] Environmental management system (EMS)
[47] Open-access web-based platform
[48] Physical education examination platform
[50] Remote Laboratory Management System
[39] System for educational resources
[38] Web platform
[33] Web-based tool to collect symptoms
[31] Web-based e-testing system
[35] Web3D
Collaborative (4)
Collaborative environment [38]
Collaborative learning [32]
Computer-Supported Collaborative Learning (CSCL) [29], [40]
E-learning (5)
E-learning [29], [37], [46]
E-Learning ecosystem [45]
E-learning IMS-LD [49]
Learning (2)
Interactive learning environments [31]
Ubiquitous learning environments [44]
Architecture (16)
E-learning-Oriented software architecture design [37]
Flexible software architecture [34]
Hardware and software architecture design [36], [50]
Health science software architecture [33]
Multiagent system architecture [28]
Multicore architecture [47]
Multitier architecture [43]
Novel software architecture [35]
Services oriented software architecture [42]
Software architecture [31], [39], [41], [45]
Software architecture and design [29]
Software architecture evaluation [40]

FIGURE 4 Main characteristics of the papers analysed

development of educational websites. This architecture groups related functionalities in different layers vertically, one on top of the other. The layered architecture is also appropriate if different devices and client types are to be supported, or if complex and/or configurable business rules and processes are to be implemented. The advantages of layered architecture are abstraction, isolation, manageability, performance, re-usability and test-ability [52]. Therefore, taking this software architecture as a reference, online collaborative educational environments can be developed to foster the teaching–learning process.

Another important finding in the selected articles is the 'model-driven engineering' software development methodology used for the creation of educational websites. An MDE helps to reduce risks by combining tested code fragments according to the model specification, which is easier to understand than the source code [55]. In addition, this approach greatly guarantees the quality of the software product because of its high level of abstraction, productivity, compatibility, portability, shorter time-to-market and maintenance costs [54]. In 2020, Ordoñez, Hilera and Cueva [56] conducted an SLR on model-driven development of accessible software. The authors state that the model-driven approach has attracted the attention of the accessible software development community because of its ability to generate code from models. In addition, it contributes to software quality requirements, such as those related to performance, functionality, human–computer interaction (usability and accessibility), compatibility, reliability, maintainability, security and portability, which are represented in ISO/IEC 25010 [57]. Therefore, incorporating quality standards (accessibility, usability, and so on) in a holistic manner will help to avoid infringing the legal rights and interests of people with and without disabilities and ensure inclusive online education. Considering that education is a right for all and that the Web is considered a basic resource in the information society, it is imperative that educational websites are accessible.

Another important point is the technological components used for the development of educational websites in the selected papers. In Figure 3 the technological components most commonly used in the development of educational websites can be seen.

Finally, another finding in the empirical results of the 23 selected papers was that eight papers aim to propose a software architecture, seven to develop a system, three to propose a model, two to evaluate a platform, one to propose an ontology maintenance based on Folksonomy, one to review the design of smart homes and one to summarise the Let's HPC project (see Table 7). In addition, the main characteristics that stand out the most in the selected studies were analysed, such as: architecture, education, collaboration, learning, e-learning, ontology, software and system, as can be seen in the conceptual map shown in Figure 4.

In summary, the architecture that stands out most in the analysed works is the layered architecture, the methodology that stands out most is MDE and the technological tools that stand out most are UML, Ruby, HTML, Apache, AJAX, PHP, XML, MYSQL, JavaScript, HTTP and Web services. In addition, these studies present wide knowledge on the development of Web-based learning platforms in interactive and online collaborative environments, remote laboratories, design and evaluation of software architectures, maintenance ontology and analysis of the developed systems.

5 | LIMITATIONS

An SLR can be affected by many limitations. The main limitation of this research is that it only reflects the software architectures and methods declared in the scientific articles that the SLR yielded. Therefore, the results and subsequent analysis are not based on a sample obtained from all educational websites, but on those websites that have been described in a scientific article.

Another limitation in this SLR is the bias of the authors in the extraction of the data. To minimise this bias, inclusion, exclusion and QA criteria have been applied in the paper selection process. Furthermore, terms such as websites, education, software architecture and empirical results were included in the search string, and their replacement terms were also included to have a greater scope in the search. Additionally, the results of the preliminary query string were assessed to determine whether the information retrieved met the expectations of this literature review. This refines the search query through an iterative process to ensure accurate and useful data extraction. It should be noted that all authors participated and contributed to the research at all stages described in the SLR methodology.

Another limitation is that the search string could have excluded documents that may contribute significantly to this SLR. Despite having followed a systematic and very well-defined plan, it cannot be guaranteed that all documents related to the subject of the investigation have been recovered. Since only databases that index high-quality content are considered, Google Scholar is excluded from this review [58].

Another important limitation of this study is that not all papers specify the software development methodologies. However, they have been classified according to its keywords taking into account the most relevant software development methodologies.

Another limitation is that the SLR used does not consider gray literature or unconventional literature such as publications in blogs, videos, white papers, etc. The multivocal literature review (MLR) is a form of SLR that consists of gray literature. Both researchers and practitioners use MLRs to carry our research in a particular area on the techniques and practices used. There is much research using MLR in other areas, but recently it has started to be seen in software engineering [59].

6 | CONCLUSIONS AND FUTURE WORKS

An SLR was conducted to summarise the software architectures proposed or used in the selected papers. Furthermore, this SLR analyzes the objectives, software development methodologies, technological components and empirical results detailed in the selected papers.

Through this SLR, it was possible to determine the software architecture of educational websites from 2009 to June 2020. After searching in four different electronic databases, 9561 papers were found. Of the 9561 papers, after making use of the inclusion, exclusion and evaluation criteria, the number of papers selected for the study was 23. The results of the SLR provide a comprehensive analysis of the trajectory of this research in the selected papers, answering five research questions.

Education is in a process of continuous development, adapting to the new requirements of laws, regulations and teaching [60]. In addition, we are living in an era of technological change and new models of education. From the perspective of education, there are many empirical research results on how technology transforms the teaching–learning process. This means that educational websites, remote laboratories, digital libraries, virtual classrooms, and so on must include quality standards, laws and regulations for universal access and inclusive collaborative learning in their software architectures.

The results of the SLR show that the software architectures in educational websites development analysed in the selected papers propose a software architecture, develop a system, propose a model, assess of a platform, propose a Folksonomy-Based Ontology Maintenance, review smart home design and summarise the Let's HPC project. Of the 23 papers, 13 carry out an evaluation of their research with either students, teachers or professionals or a combination of these. These research papers are narrated experiments of projects or individuals that seek to improve collaborative learning in the educational area.

The educational process has changed the way students are taught around the world in recent years. Virtual collaborative teaching–learning environments in educational institutions are a necessity today. This means that software architectures must be created that encourage teaching–learning processes using online collaborative educational environments. In addition, software architectures must comply with standards, regulations and laws regarding educational inclusion to ensure education for all.

Future work will analyse the evolution of the architecture of software for educational websites, and also their good practices or existing strategies for the creation and management of learning resources, open educational resources, accessible massive open online courses (MOOC), adaptability and accessibility features used in education virtual platforms or campuses.

ORCID

Milton Campoverde-Molina https://orcid.org/0000-0001-5647-5150
Sergio Luján-Mora https://orcid.org/0000-0001-5000-864X
Llorenç Valverde https://orcid.org/0000-0002-9163-568X

REFERENCES

1. Qureshi, N. et al.: Empirical evidence in software architecture: a systematic literature review protocol. In: The Sixth International Conference on Software Engineering Advances, Barcelona, Spain, pp. 534–538 (2011)
2. Breivold, H.P., Crnkovic, I., Larsson, M.: A systematic review of software architecture evolution research Jan. Inf. Software Technol. 54(1), 16–40 (2012). https://doi.org/10.1016/j.infsof.2011.06.002
3. Williams, B.J., Carver, J.C.: Characterising software architecture changes: a systematic review Jan. Inf. Software Technol. 52(1), 31–51 (2010). https://doi.org/10.1016/j.infsof.2009.07.002
4. Qureshi, N., Usman, M., Ikram, N.: Evidence in software architecture, a systematic literature review, Apr. In: 17th International Conference on Evaluation and Assessment in Software Engineering, Porto de Galinhas, Brazil, pp. 97–106 (2013). https://doi.org/10.1145/2460999.2461014
5. Van Der Ven, J.S., Bosch, J.: Busting software architecture Beliefs: a survey on Success factors in architecture decision making, Oct. In: 42th Euromicro conference on Software Engineering and Advanced Applications (SEAA), Limassol, Cyprus, pp. 42–49 (2016). https://doi.org/10.1109/SEAA.2016.35
6. López-Sanz, M., de Castro, V., Marcos, E.: An architecturecentric process for service oriented systems development: developing for the intelligent pavement, Nov. In: XL Latin American Computing Conference (CLEI), Montevideo, Uruguay, pp. 1–9 (2014). https://doi.org/10.1109/CLEI.2014.6965176
7. Newell, A.F., Gregor, P.: User sensitive inclusive design--- in search of a new paradigm Jan. In: Proceedings of the Conference on Universal Usability, Arlington, Virginia, Unated States, pp. 39–44 (2000). https://doi.org/10.1145/355460.355470
8. United Nations: Convention on the Rights of Persons with Disabilities – Articles. United Nations New York (2006). [Online] http://cort.as/-GlqK
9. Lawton, H.S.: Introduction to web accessibility Jun. World Wide Web Consortium (2019). [Online]. Available: https://www.w3 . org/WAI/fundamentals/accessibility-intro/. https://www.w3.org/WAI/fundamentals/accessibility-intro/
10. Akram, M., Bt, R.: A systematic literature review to determine the web accessibility issues in Saudi Arabian university and government websites for disable people Jun. ijacsa. 8(6), 321–329 (2017). https://doi.org/10.14569/IJACSA.2017.080642
11. Barbara, K., Budgen, D., Brereton, P.: Using mapping studies as the basis for further research – a participant-observer case study Jun. Inf. Software Technol. 53(6), 638–651 (2011). https://doi.org/10.1016/j.infsof.2010.12.011Special Section: Best papers from the APSEC
12. Kitchenham, B.: Procedures for Performing Systematic Reviews. Jul, vol. 33, pp. 1–33.Keele Univ., Keele, (2004)
13. Webster, J., Watson, R.T.: Analysing the past to prepare for the future: writing a literature review Jun. MIS Q. 26(2), xiii–xxiii (2002)
14. Kitchenham, B., et al.: Systematic literature reviews in software engineering - a systematic literature review Jan. Inf. Software Technol. 51(1), 7–15 (2009). https://doi.org/10.1016/j.infsof.2008.09.009
15. Kitchenham, B., et al.: Systematic literature reviews in software engineering - a tertiary study Aug. Inf Software Technol. 52(8), 792–805 (2010). https://doi.org/10.1016/j.infsof.2010.03.006
16. Dias de Carvalho, D., et al.: Software process lines: a systematic literature review, Nov. In: 14th Software Process Improvement and Capability Determination, Vilnius, Lithuania, pp. 118–130 (2014). https://doi.org/10.1007/978-3-319-13036-1_11
17. Amin, A., et al.: A Snapshot of 26 Years of research on Creativity in software engineering – a systematic literature review, Jun. In: Mobile and Wireless technologies, Kuala Lumpur, Malaysia, pp. 430–438 (2017). https://doi.org/10.1007/978-981-10-5281-1_47
18. Martinez-Mosquera, D., Navarrete, R., Luján-Mora, S.: Modelling and management big data in databases-A systematic literature review Jan. Sustainability. 12, 634 (2020). https://doi.org/10.3390/su12020634
19. Shafique, F.: Architecture of a nation-wide educational information system infrastructure: a systematic review. May, pp. 15–21 (2013). https://doi.org/10.1109/ECONF.2013.49
20. Zapata, B.C., et al.: Empirical studies on usability of mHealth apps: a systematic literature review Jan. J. Med. Syst. 39(2), 1 (2015). https://doi.org/10.1007/s10916-014-0182-2
21. Dy, S.M., et al.: Tell Us: a web-based tool for improving communication among patients, families, and providers in hospice and palliative care through systematic data specification, collection, and use Oct. J. Pain Symptom Manag. 42(4), 526–534 (2011). https://doi.org/10.1016/j.jpainsymman.2010.12.006
22. Stapić, Z., et al.: Scrutinising systematic literature review process in software engineering Feb. TEM J. 5, 104–116 (2016). https://doi.org/10.18421/TEM51-16
23. Héctor, C., Vasilios, A., Paris, A.: Architecting systems of systems: a tertiary study Oct. Inf. Software Technol. 118, 106–202 (2020). https://doi.org/10.1016/j.infsof.2019.106202
24. Bermejo, B., Juiz, C.: Virtual machine consolidation: a systematic review of its overhead influencing factors Jan. J. Supercomput. 76(1), 324–361 (2020). https://doi.org/10.1007/s11227-019-03025-y

25. Campoverde-Molina, M., Luján-Mora, S., Garcia, L.V.: Empirical studies on web accessibility of educational websites: a systematic literature review May. IEEE Access. 8, 91676–91700 (2020). https://doi.org/10.1109/ACCESS.2020.2994288
26. Gusenbauer, M., Haddaway, N.R.: Which academic search systems are suitable for systematic reviews or meta-analyses? Evaluating retrieval qualities of Google Scholar, PubMed, and 26 other resources Oct. Res. Syn. Meth. 11, 181 (2020). https://doi.org/10.1002/jrsm.1378
27. Akers, J.: Systematic Reviews: CRD's Guidance for Undertaking Reviews in Healthcare. Centre for Reviews and Dissemination, University of York York (2009)
28. Suo, Y., et al.: Open smart classroom: Extensible and scalable learning system in smart Space using web service technology Jun. IEEE Trans. Knowl. Data Eng. 21(6), 814–828 (2009). https://doi.org/10.1109/TKDE.2008.117
29. Caballé, S., Xhafa, F.: CLPL: Providing software infrastructure for the systematic and effective construction of complex collaborative learning systems Nov. J. Syst. Software. 83(11), 2083–2097 (2010). https://doi.org/10.1016/j.jss.2010.06.013
30. Jara, C.A., et al.: EJS+EjsRL: an interactive tool for industrial robots simulation, Computer Vision and remote operation Jun. Robot Autonom. Syst. 59(6), 389–401 (2011). https://doi.org/10.1016/j.robot.2011.02.002
31. Mustakerov, I., Borissova, D.: A conceptual approach for development of educational Web-based e-testing system Oct. Expert Syst. Appl. 38(11), 14060–14064 (2011). https://doi.org/10.1016/j.eswa.2011.04.214
32. Gasevic, D., et al.: An approach to folksonomy-based ontology maintenance for learning environments Oct. IEEE Trans. Learning Technol. 4(4), 301–314 (2011). https://doi.org/10.1109/TLT.2011.21
33. Bero, B.N., et al.: Challenges in the development of environmental management systems on the modern university campus Apr. Int. J. Sus. Higher Ed. 13(2), 133–149 (2012). https://doi.org/10.1108/14676371211211827
34. Gatto, I., Pittarello, F.: Creating Web3D educational stories from crowdsourced annotations Dec. J. Vis. Lang. Comput. 25(6), 808–817 (2014). https://doi.org/10.1016/j.jvlc.2014.10.010
35. Kalúz, M., et al.: A flexible and configurable architecture for automatic control remote laboratories Jul. IEEE Trans. Learning Technol. 8(3), 299–310 (2015). https://doi.org/10.1109/TLT.2015.2389251
36. Lu, P., Cong, X., Zhou, D.: E-learning-Oriented software architecture design and case study Sep. Int. J. Emerg. Technol. Learn. 10(4), 59–65 (2015). https://doi.org/10.3991/ijet.v10i4.4698
37. Rajib, M.A., et al.: SWATShare - a web platform for collaborative research and education through online sharing, simulation and visualization of SWAT models Jan. Environ. Model Software. 75, 498–512 (2016). https://doi.org/10.1016/j.envsoft.2015.10.032
38. Pérez-Rodríguez, R., et al.: Architecture of a concept-based information retrieval system for educational resources Nov. Sci. Comput. Program. 129, 72–91 (2016). https://doi.org/10.1016/j.scico.2016.05.005
39. Magnisalis, I., Demetriadis, S.: An architecture combining IMS-LD and web services for flexible data-transfer in CSCL. Apr, IEEE Trans. Learning Techno. 10, 205–218 (2017). https://doi.org/10.1109/TLT.2015.2512604
40. Angelov, S., de Beer, P.: Designing and applying an approach to software architecting in agile projects in education May. J. Syst. Software. 127, 78–90 (2017). https://doi.org/10.1016/j.jss.2017.01.029
41. Muhammad, A., et al.: Educational aspects of service orientation: smart home design issues and technologies May. TEM J. 6(2), 250–257 (2017). https://doi.org/10.18421/TEM62-09
42. Cortés, H., Navarro, A.: Enterprise WAE: a lightweight UML extension for the characterisation of the presentation tier of enterprise applications with MDD-based mockup generation May. Int. J. Soft. Eng. Knowl. Eng. 27(8), 1291–1331 (2017). https://doi.org/10.1142/S0218194017500486
43. Araujo, R.D., et al.: Social and collaborative interactions for educational content enrichment in ULEs Jul. Educ. Technol. Soc. 20(3), 133–144 (2017)
44. Ouf, S., et al.: A proposed paradigm for smart learning environment based on semantic web Jul. Comput. Hum. Behav. 72, 796–818 (2017). https://doi.org/10.1016/j.chb.2016.08.030
45. Srai, A., et al.: Generated PSM web model for e-learning platform respecting n-tiers architecture Nov. Int. J. Emerg. Technol. Learn. 12(10), 212–220 (2017). https://doi.org/10.3991/ijet.v12i10.7179
46. Chaudhury, B., et al.: Let's HPC: a web-based platform to aid parallel, distributed and high performance computing education Aug. J. Parallel Distr. Comput. 118, 213–232 (2018). https://doi.org/10.1016/j.jpdc.2018.03.001
47. Cheng, F., Yin, Y.: Application of computer data analysis technology in the development of a physical education examination platform Jan. Int. J. Emerg. Technol. Learn. 14(6), 75–86 (2019). https://doi.org/10.3991/ijet.v14i06.10158
48. Aouine, A., Mahdaoui, L., Moccozet, L.: A workflow-based solution to support the assessment of collaborative activities in e-learning Mar. Ijilt. 36(2), 124–156 (2019). https://doi.org/10.1108/IJILT-01-2018-0004
49. Villar-Martínez, A., et al.: Improving the scalability and Replicability of embedded systems remote laboratories through a Cost-effective architecture Nov. IEEE Access. 7, 164164–164185 (2019). https://doi.org/10.1109/ACCESS.2019.2952321
50. Yogendra Kumar, J., Santosh Kumar, B.: Min max normalisation based data perturbation method for privacy protection. Int. J. Comput. Commun. Techn. 3, 45–50 (2014)
51. Meier, J., et al.: Application Architecture Guide 2.0, 2nd ed. patterns & practices Washington (2008)
52. Microsoft: Chapter 3: Architectural Patterns and Styles. Jan. Microsoft Corporation (2010). [Online]. https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ee658117(v=pandp.10)
53. Plaza, A.M., Diaz, J., Perez, J.: Software architectures for health care cyber-physical systems: a systematic literature review J. Software-Evol. Process. 30(7), 1–23, Jul. (2018). https://doi.org/10.1002/smr.1930
54. Despa, M.L.: Comparative study on software development methodologies. Database Systems Journal. 5(3), 37–56 (2014)
55. Lu, Q., et al.: Integrated Model-Driven Engineering of Blockchain Applications for Business Processes and Asset Management. Practice and Experience. Nov. 51(2), 1–21 (2020). https://doi.org/10.1002/spe.2931
56. Ordoñez, K., Hilera, J., Cueva, S.: Model-driven development of accessible software: a systematic literature review. Sep. Univ. Access Inf. Soc. 1–30 (2020). https://doi.org/10.1007/s10209-020-00751-6
57. Hussain, A., Mkpojiogu, E.O.C.: An application of the ISO/IEC 25010 standard in the quality-in-use assessment of an online health awareness system Nov. Jurnal. Teknologi. 77(5) (2015). https://doi.org/10.11113/jt.v77.6107
58. Vázquez-Ingelmo, A., Garcia-Peñalvo, F.J., Therón, R.: Information dashboards and tailoring capabilities - a systematic literature review Aug. IEEE Access. 7, 109673–109688 (2019). https://doi.org/10.1109/ACCESS.2019.2933472
59. Garousi, V., Felderer, M., Mäntylä, M.V.: Guidelines for including grey literature and conducting multivocal literature reviews in software engineering Feb. Inf. Software Technol. 106, 101–121 (2019). https://doi.org/10.1016/j.infsof.2018.09.006
60. Campoverde-Molina, M., Luján-Mora, S., Valverde García, L.: Web accessibility in the web portals of the educational institutions of Ecuador. Preliminary analysis, Jul. In: 11th International Conference on Education and New Learning Technologies (EDULEARN19), Palma de Mallorca, Spain, pp. 4697–4707 (2019). https://doi.org/10.21125/edulearn.2019.1171

How to cite this article: Campoverde-Molina, M., Luján-Mora, S., Valverde, L.: Systematic literature review on software architecture of educational websites. IET Soft. 15(4), 239–259 (2021). https://doi.org/10.1049/sfw2.12024

APPENDIX

Data extracted

TABLE 9 Data extracted for RQ1, RQ2, RQ3 and RQ4

Paper	RQ1 Objective	RQ2 Architecture	RQ3 Methodology	RQ4 Technological Components
[28]	Build a prototype software referred to as Open Smart Classroom based totally on the multiagent system architecture.	Service-oriented architecture (SOA)	Prototyping	Tomcat, Axis, Web services (WSDL, SOAPClient), HTTP
[29]	Provide a software infrastructure for the systematised and effective construction of complex collaborative learning systems.	Service-oriented architecture (SOA)	Model-driven engineering	PHP, Java, XML, UML, Web services (SOAP, WSDL)
[30]	Introduce an interactive Java software platform that allows users to easily create advanced robotic applications.	Layered architecture	Dynamic systems development method	Easy Java simulations (EJS), object-oriented Java library (EjsRL), C/C++, Java, Matlab, HTTPS
[31]	Develop an educational Web-based e-testing system.	Client-server architecture	Prototyping	HTML, DHTML, XML, JavaScript, triggers, e-mail
[32]	Present an approach to ontology maintenance based on the use of collaborative tags contributed by learners while using learning environments.	Client-server architecture	Model-driven engineering	Tag cloud, LOCO-Analyst
[21]	Develop a Web-based tool to collect symptoms and needed data and provide feedback to hospice and palliative care patients, caregivers and providers.	Layered architecture	SCRUM	Health Science Process Framework (HSPF), Clinical Research Management System (CRMS), Ruby and Microsoft SQL Server 2005
[33]	Provide a modular, extensible software architecture for EMS.	Layered architecture	Prototyping	PHP, MYSQL, JavaScript
[34]	Describe a software architecture for creating engaging linear narrations that can be shared on the Web.	Client-server architecture	Model-driven engineering	ToBoA-3D, X3D, XHTML, CSS, JavaScript, Apache, MYSQL, PHP, JavaScript
[35]	Propose a novel approach in hardware and software architecture design for implementation of remote laboratories for automatic control.	Client-server architecture	Dynamic systems development method	JavaScript, XML, AJAX, GWT SDK, HTTP
[36]	Present a domain model and component model of an e-learning system and component integration method on the basis of web service.	Layered architecture	Model-driven engineering	Web services, XML, UDDI/WSDL/SOAP protocol, XML data bus, Black-box framework, CORBA, EJB, COM+
[37]	Present the software architecture, functional capabilities and implementation of SWATShare as a collaborative environment for hydrology research and education using the models published and shared in the system.	Client-server architecture	Prototyping	Flex software development kit, GeoServer, Tomcat, Web services, MYSQL, Globus, PHP, Python, Apache server, Dublin Core
[38]	Present the SDE, a search engine for educational resources that was built on top of the knowledge provided by Wikipedia.	MVC	Model-driven engineering	Ruby, Nokogiri library, Geocoder library, Google Maps API, AJAX, DOM, Capybara library, Apache Solr, HTTP, HTML, MySQL, MVC

(Continues)

TABLE 9 (Continued)

Paper	RQ1 Objective	RQ2 Architecture	RQ3 Methodology	RQ4 Technological Components
[39]	Present evaluation data regarding the MAPIS3 architecture.	Layered architecture	Test-driven development	Mentorchat, Web services, PHP, MYSQL, JavaScript, Google visualization API, Google forms tool
[40]	Present an approach to the introduction of software architecting activities in an agile project course.	…..	SCRUM	…..
[41]	Efforts are made to review the state of the art of service-oriented technologies in smart home design.	SOA	Dynamic systems development method	Web services, HTTP, XML, Object Access Protocol (SOAP), DPWS
[42]	Present; (E-WAE), a lightweight UML extension for the modelling of these elements, which permits the inclusion of multitier, service-oriented architecture (SOA) and security design-level patterns in the models.	SOA	Model-driven engineering	JSF, WAE4x, ASP.NET MVC
[43]	Present a model for content enrichment in ubiquitous learning environments.	Component-based architecture	Prototyping	AJAX, Web services, JSON, HTTP, JavaScript, Stylesheet, MYSQL
[44]	Propose and implements framework for smart e-learning ecosystem.	Layered architecture	Model-driven engineering	Protégé
[45]	Proposes a solution to generate a platform-specific model (PSM) based on n-tier architecture from a platform independent model (PIM).	N-tier architecture	Model-driven engineering	UML, Spring MVC, Spring IOC, Hibernate, Jsp pages
[46]	Summarize Let's HPC project.	Client-server architecture	Dynamic systems development method	MongoDB, ExpressJS, Angular, Node
[47]	Design of a platform for the evaluation of physical education using data mining to help teachers provide individualised instruction.	Layered architecture	Model-driven engineering	ADO.NET, ASP.NET, Web services, XML, data mining technology, SQL server 2005
[48]	Support the assessment of collaborative activities in e-learning: a design founded on IMS-LD meta-model.	Layered architecture	Rapid application development	Web programming, Web services, ProcessMaker, WfMS database.
[49]	Propose a new mixed hardware-software architecture.	Layered architecture	Dynamic systems development method	Web-based client, laboratory WebServer, RLMS, interactive live-streaming platform

No information was found '…..'

Abbreviations: CRMS, Clinical Research Management System; DPWS, devices profile for web services; E-WAE, Enterprise Web Application Extension; GWT, Google Web Toolkit; HSPF, Health Science Process Framework; JSF, Java Server Faces; MVC, Model-view controller; SOA, service-oriented architecture; PSM, platform-specific model; PIM, platform independent model; RLMS, remote laboratory management system.

TABLE 10 Data extracted for RQ5

	RQ5			
Paper	Result Type	Modules	Evaluation	Limitations
[28]	Prototype open smart classroom	SPAW, WSWA and OSPG	The experiment involves seven students from Tsinghua and five from Kyoto. The experiment lasts about 140 min.	It is not suitable for real-time streaming application due to the limitation of the HTTP protocol, backtranslation is a heavy-load task, and Langrid does not have a good load-balancing mechanism so far.
[29]	Software infrastructure called CLPL	…..	Master's thesis courses at the Open University of Catalonia when developing complex software applications in the CSCL domain.	…..
[30]	Robots simulation	Matrix computation, robotics, remote operation and computer vision	To validate some of the system's capabilities, they have been used as teaching tools for university students in various robotics courses since 2009, with positive acceptance.	…..
[31]	Prototype of e-testing system	Password module, choice module, shuffle and random module, checking and scoring module, evaluation module	'C Programming Language' course students	…..
[32]	Ontology maintenance	…..	Participants from Athabasca University, Simon Fraser University, a private Canada-based company developing and offering technology and content for professional training and University of Belgrade. In total, 22 people (17 men and 5 women).	…..
[21]	Tell Us™	Questionnaire authoring, patient and family self-report, automated e-mail alerts to hospice, cross-sectional and longitudinal data display	…..	…..
[33]	EMS	Real-time data, manually collected data, spatial data types	…..	…..
[34]	Web3D educational stories	…..	…..	…..
[35]	Hardware and software architecture	Thermo-optical plant, DC motor and air heat exchanger, hydraulic system with coupled tanks	The evaluation was provided by 32 students participating in several courses (Integrated Control in Process Engineering, Process Control, Process Modelling and System Identification).	The main limitation of MHSA is the fact that it can be used only to develop laboratories based on PLCs.
[36]	E-learning-oriented software architecture design	User management, tests, learning content management, ask-answer, exercises, cooperation, memos and so on	Case study: Development of a web-based Multipurpose Hardware and Software Architecture (MHSA)	…..
[37]	Software prototype (hydrology)	View (My models, shared models and other models), upload, edit, run, visualization	Three case studies: case study 1 – educational application through model sharing (19 students in CE-54900: Computational Watershed Hydrology course); case study 2 – collaborative research through model sharing (a model created by user A is extended and/or reused by users B and C); case study 3 – simultaneous simulations of multiple SWAT models (multiple SWAT models)	…..

(Continues)

TABLE 10 (Continued)

	RQ5			
Paper	**Result Type**	**Modules**	**Evaluation**	**Limitations**
[38]	Information retrieval system	Ontology, scrapes, text world, semantic world, semantic annotation module, indexing module, application module	Its evaluation was developed in the context of the EU FP7 iTEC project, different locations around Europe with a total of 65 participants [21 Oulu (Finland), 9 Lisbon (Portugal), 9 Budapest (Hungary), 6 Vigo (Spain), 10 Bad Hofgastein (Austria) and 10 Vilnus (Lithuania)]	The main limitations of this research are about the evaluation process. The evaluation protocol is not fully formal described, and the methods used are neither formal nor systematic, another limitation of the proposed approach is that the interests of users are accumulated through time, as they navigate and bookmark resources.
[39]	Proposed MAPIS3 architecture	…..	Three case studies: case study 1: forum visualization and MAPIS3-based LD, 13 postgraduate students participated, duration 1 week; case study 2: forum visualization, 83 students of secondary education school for adults, duration 6 weeks; case study 3: orchestrating chat-forum tools, 39 students, duration 1 week	Limitations of the study include: (a) The small number of stakeholders with a teacher's role, (b) the relatively narrow background of the learners, (c) the data analysis method, which provided a percentage tendency which is by no means definitive and deserves further investigation and (d) the relatively low variety of interconnected tools in the case studies.
[40]	Agile project course	…..	…..	With respect to internal validity, a number of variables pose limitations to our conclusions and require attention (historical bias, teacher bias, selected case, agile theory bias, group composition, group context) and with respect to external validity of the experiment (cultural bias, educational context, age)
[41]	Smart home	Communication media manager, event manager, registry, resource manager, messaging system, stream manager, functional component module, device control, application and device control module manager	…..	The main limitation of UPnP is that one cannot access the service outside a local area network, another limitation is that it requires human intervention.
[42]	Enterprise applications	…..	The US Library of Congress Online Catalog (17 million records)	…..
[43]	Ubiquitous learning environments	Two modules: one for performing service requests and displaying graphical components, and another for fetching and storing system information	Case study with the Classroom eXperience platform (three school semesters with 121 undergraduate students)	The results cannot be generalised because they may be biased by the mastery of the courses studied (Computer Science and Information Systems).
[44]	Smart learning environment	…..	…..	…..
[45]	PSM Web model	…..	…..	…..
[46]	Summary of Let's HPC project	…..	Autumn semester 2017, a comprehensive survey was used to gather feedback from the 53 students who used the platform throughout the CS301 course for their assignments and projects	…..

TABLE 10 (Continued)

	RQ5			
Paper	Result Type	Modules	Evaluation	Limitations
[47]	Physical education assessment platform	Exam item management function module, exam type management function module, score analysis and statistics module, a platform administration function module and score management function module.	…..	…..
[48]	Collaborative assessment processes	…..	Case study summative assessment (a month with learners' groups of third year of licence, Computer Science)	Further experimentation is necessary to test the effectiveness of the proposed system in order to analyse its performances under a massive usage; in addition, use a survey to collect learners' opinions.
[49]	Propose a new mixed hardware–software architecture	…..	Technical evaluation, cost efficiency evaluation, scalability evaluation	…..

No information was found '…..'.

Abbreviations: CLPL, Collaborative Learning Purpose Library; EMS, environmental management system; SPAW, smart-platform-agent-webservice; OSPG, open-smart-platform-gateway; PSM, platform-specific model; WSWA, web-service-wrapper-agent.

TABLE 11 Bibliometric data extraction

Paper	Name	JCR	SJR	Year
[28]	IEEE Transactions on Knowledge and Data Engineering	Q1	Q1	2009
[29]	Journal of Systems and Software	Q2	Q1	2010
[30]	Robotics and Autonomous Systems	Q3	Q1	2011
[31]	Expert Systems with Applications	Q1	Q1	2011
[32]	IEEE Transactions on Learning Technologies	Q3	Q2	2011
[21]	Journal of Pain and symptom management	Q1	Q2	2011
[33]	International Journal of Sustainability in Higher Education	Q2	Q2	2012
[34]	Journal of Visual languages and Computing	Q3	Q2	2014
[35]	IEEE Transactions on Learning Technologies	Q3	Q2	2015
[36]	International Journal of Emerging Technologies in Learning	…..	Q3	2015
[37]	Environmental Modelling and Software	Q1	Q1	2016
[38]	Science of Computer Programming	Q3	Q2	2016
[39]	IEEE Transactions on Learning Technologies	Q3	Q2	2017
[40]	Journal of Systems and Software	Q1	Q1	2017
[41]	TEM Journal	…..	…..	2017
[42]	International Journal of Software Engineering and Knowledge Engineering	Q4	Q4	2017
[43]	Educational Technology and Society	Q2	Q1	2017
[44]	Computers in Human Behavior	Q1	Q1	2017
[45]	International Journal of Emerging Technologies in Learning	…..	Q3	2017
[46]	Journal of Parallel and Distributed Computing	Q2	Q2	2018
[47]	International Journal of Emerging Technologies in Learning	…..	Q3	2019
[48]	International Journal of Information and Learning Technology	…..	Q2	2019
[49]	IEEE Access	Q1	Q1	2019

Note: No information was found '…..'