IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 May 2024, accepted 4 June 2024, date of publication 14 June 2024, date of current version 27 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3414503

SURVEY

Exploring the Effectiveness and Trends of Domain-Specific Model Driven Engineering: A Systematic Literature Review (SLR)

AMINA ZAFAR, FAROOQUE AZAM, AFSHAN LATIF, MUHAMMAD WASEEM ANWAR, AND AON SAFDAR

Department of Computer and Software Engineering, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Islamabad 44000, Pakistan

Corresponding author: Afshan Latif (alatif.cse20ceme@student.nust.edu.pk)

This work was supported by the Higher Education Commission, Pakistan, through the National Research Program for Universities (NRPU) Multiple Representations of Embedded system Designs (MRED) Project under Grant 20-15651.

ABSTRACT Rapid technological advancements have resulted in increasingly complex software systems, posing challenges during development in terms of time and cost. Adopting domain-specific modeling (DSM) brings numerous benefits to software engineering, including enhanced efficiency, improved maintenance capabilities, higher software quality, reduced development time, and increased potential for cost-effective software solutions through improved reusability. Despite the proven effectiveness of DSM in various domains, a study summarizing recent advancements is hard to find in the state-of-the-art. Therefore, in this article, we present a comprehensive systematic literature review that examines the application of DSM in various domains (4). We selected 99 studies and classified those into four categories, i.e., meta-modeling (42), domain-specific languages (39), UML profiles (9), and general (9) based on the use of DSM approaches. We identified various tools from the selected studies, i.e., 21 existing and 91 proposed or developed. Moreover, model-driven engineering (MDE) techniques, including validation (12), simulation (5), verification (4), and software architectural modeling (3), are presented and analyzed. We further explained the type of model transformation employed in each study, i.e., model-to-text (49) and model-to-model (4). Finally, the regions participating in DSM's growth are also investigated. It is concluded that Ecore is the leading meta-modeling tool, Xtext is the often-used domain-specific tool, Sirius is graphical, and UPPAAL is the most utilized verification tool identified. Moreover, Validation is a frequently used MDE technique, and Model-to-text transformation with Acceleo is the most utilized transformation type in the selected studies. The comprehensive results of this research provide valuable guidance for DSM researchers and practitioners in choosing a suitable tool and technique that meets their specific requirements.

INDEX TERMS Model-driven engineering, domain-specific modeling, domain-specific language.

I. INTRODUCTION

The pursuit of efficient, dependable, and maintainable systems has been ongoing in software engineering. Traditional software development focuses on code-centric paradigms throughout the entire development life cycle. It relies heavily on manual coding and the specification of system behaviors [1]. However, as software systems grow in complexity and scale, the traditional development approaches have shown several limitations and challenges. Traditional systems often suffer significant challenges due to the extensive manual effort required in coding, implementation, integration, and testing processes [2]. These manual tasks consume valuable time and resources and introduce the risk of

The associate editor coordinating the review of this manuscript and approving it for publication was Leimin Wang.

errors and inconsistencies. Consequently, project schedules are frequently delayed, and costs escalate [3]. Moreover, traditional system development often lacks a clear bifurcation between the system's design and implementation, making it difficult to modify or adapt the system as requirements evolve. [4]

To overcome these challenges, Model-Driven Engineering (MDE) has emerged as a software development methodology led by the modeling standards of the object management group (OMG) [5]. Furthermore, Domain-Specific Modeling (DSM) is the central modeling aspect of MDE, facilitating the creation of abstract software models across diverse domains throughout the development process to capture system requirements, design, and behavior [6]. These models visually represent the software system and its components, facilitating enhanced comprehension and manipulation. DSM employs models as central artifacts, which promotes the separation of concerns and modularization. This modularization enhances system maintainability and extensibility, as changes to one part of the model can be propagated throughout the system. DSM also facilitates automatic code generation from models, eliminating the need for manual coding [7]. With the help of model transformation techniques, developers can generate executable code directly from the models, reducing human errors and ensuring consistency between the design and implementation. This code generation aspect of DSM significantly enhances productivity by freeing developers from low-level coding tasks [8], [9], [10]. Various DSM approaches, such as domain-specific modeling (DSML) and domain-specific language (DSL), have been developed to cater to specific aspects of system development. Due to their benefits, DSM approaches have been employed in many domains.

Certain SLR-based studies published before 2021 have extensively analyzed the effectiveness of DSM in multiple domains. A journal study from 2020 utilized systematic mapping to explore the models' development with DSM tools spanning the years 2012 to 2019 [11]. Consequently, a study is needed to analyze and summarize the latest trends in DSM, such as DSM tools and techniques, as well as the challenges involved in DSM research in the last two years. No study comprehensively analyzed and summarized the DSM tools and techniques and their applications in specific domains within the previous two years (2021-2023). It is crucial to assess the DSM's approaches, tools, and techniques to determine their effectiveness in different domains. As part of our research, we conducted a systematic literature review of studies published from January 2021 to August 2023 to provide valuable insight into DSM trends by addressing the following research questions:

RQ1: What notable studies significantly impact domain-specific modeling approaches between Jan 2021 and Aug 2023?

RQ2: In which major system domains have domain-specific modeling approaches been effectively employed and proven useful?

RQ3: What are essential, proposed, and utilized model-driven tools for software development covering graphical, transformational, and verification features?

RQ4: Which model-driven techniques have been primarily utilized in research conducted between 2021 and 2023?

RQ5: What are the prominent model transformation approaches employed within domain-specific modeling?

RQ6: Which regions have emerged as leading contributors to the growth of domain-specific modeling approaches?

RQ7: What are the significant challenges and their recommended future directions in the DSM research?

All the research questions, as mentioned earlier, are defined by thoroughly analyzing the current literature [11] on domain-specific modeling (DSM) tools; specifically, our questions are inspired by [12]. Each question is specified to explore essential aspects of recent advancements, applications, and challenges in the field. For instance, RQ1 and RQ4 aim to identify influential studies and standard techniques. On the other hand, RQ2 and RQ6 investigate the domains where DSM approaches have been practical and the geographic distribution of research contributions. RQ3 examines the landscape of model-driven tools, while RQ5 focuses on model transformation approaches within DSM. Finally, RQ7 emphasizes the importance of comprehending the significant challenges and outlining future research directions in DSM. Collectively, these questions provide a comprehensive framework to analyze the recent research developments and revolutionize the DSM research.

To find answers to the questions mentioned above, this study conducts a systematic literature review of 99 studies where domain-specific modeling (DSM) approaches are utilized for efficient systems development, published between Jan 2021 and Aug 2023. An overview of the SLR is depicted in Figure 1. Section II presents the research methodology employed for conducting this SLR. We defined four significant categories to categorize our selected studies (Section II-A) and developed a review protocol (Section II-B). We selected the studies published from January 2021 to August 2023 in well-known scientific databases, including IEEE, Elsevier, Springer, ACM, and Wiley, during our search process (Section II-B2) as mentioned in the inclusion and exclusion criteria (Section II-B1). We selected 44 studies from IEEE, 24 from Springer, 18 from ACM, 12 from Elsevier, and one from Wiley. We analyzed the quality of the selected studies (Section II-B3) and defined a template for data extraction and synthesis (Section II-B4). To simplify the data extraction process, we comprehensively analyzed the selected studies and categorized them into our defined categories: Meta-modeling, Domain-specific language, UML profile, and general categories. Among 99 selected studies, 42 belong to the meta-modeling category, 39 fall under domain-specific languages, 9 are related to the UML profile, and 9 are added to the general category (Section III-A). Our analysis identified the 4 major domains where DSM approaches have been employed extensively (Section III-B). It identified 21 utilized and 91 proposed tools (Section III-C), 4 MDE techniques (Section III-D), and three model-transformation approaches frequently used in the DSM research (Section III-E). It also revealed the regions actively involved in the DSM research. We analyzed the significant challenges in the DSM research and provided their recommended future directions. This analysis led to the answer to the defined research questions (Section IV). Although we have followed the standard guidelines to conduct this SLR, it is still limited in some ways (Section V). We have concluded this SLR with recommendations for future work to improve it (Section VI).

II. RESEARCH METHODOLOGY

Our research methodology follows standard guidelines for conducting a Systematic Literature Review (SLR) [13]. This section focuses on two main aspects: categorizing research studies based on their alignment with domain-specific modeling approaches (Section II-A) and designing the review protocol to carry out the SLR (Section II-B). By categorizing the studies and adhering to the review protocol, we ensure a rigorous and systematic approach to the research methodology.

A. CATEGORIES DEFINITION

Domain-specific modeling (DSM) approaches comprise two modeling notations: graphical and textual. Domain-Specific Modeling Language (DSML) is a particular variant of DSM characterized by graphical notation, while Domain-Specific Language (DSL) employs textual notation [14]. These DSM approaches are based on the Model Driven Architecture (MDA) vision led by the Object Management Group (OMG). This framework has four modeling levels: M3, M2, M1, and M0. However, the abstract models are created at the M2 level, and the concrete models are created at the M1 level as instances of M2-level meta-models [5]. However, DSML consists of two graphical representations: meta-modeling and UML profiles. Therefore, the selected studies are classified into four categories based on DSM approaches, descriptively defined below:

1) META-MODELING

Meta-modeling provides an abstract representation of models, focusing on visually depicting models and creating the instance model of the M1 level. Similar to the structure of a UML class diagram, a meta-model is created by dragging meta-modeling elements like classes, attributes, and relationships following the structure and semantics of modeling languages to provide a foundation for DSM [15], [16]. For example, authors in [17] and [18] developed an ecore and adoxx meta-model in the domain of embedded systems to incorporate vehicle route prediction and digital-twin cross-platform solutions, respectively. All such studies creating similar meta-models fall under the meta-modeling category.

2) DOMAIN SPECIFIC LANGUAGES

Domain-specific languages (DSLs) have a concrete syntax of models, textual syntax, and programming language similarity. The use of DSL development aids in promoting the syntax highlighting of textual structures, incorporating the grammar rules to tailor the domain-specific concepts. DSL grammar can be designed by defining the concepts, keywords, relationships, and attributes by following the syntax rules of specific modeling languages to express valid expressions [19]. For example, the authors in [20] have developed a framework using Xtext to encounter security vulnerability issues in database systems. Furthermore, the study of [21] developed a framework using the concepts of agent-based modeling language and JetBrains MPS to support participatory modeling in the healthcare domain. All such studies creating the textual models fall under the DSL category.

3) UML PROFILE

UML Profile is a UML-packaged stereotyped profile that provides a lightweight extension mechanism to extend the UML models of the M1 level. UML profile is designed by defining numerous constructs, such as stereotypes, packages, constraints, and tagged values at the M2 level [22], [23]. Like metamodeling, UML profiles facilitate the development of visual editors by expressing domain-specific concepts and their notations. For example, the authors in [24] developed a UML profile using the papyrus tool to encounter the domain-level complexity of business process modeling notations. Hence, such research studies utilizing UML Profiles are placed under this category.

4) GENERAL

There are few research studies where multiple DSM approaches (e.g., meta-modeling, DSL, and UML profile) are simultaneously utilized to develop a full-fledged domain-specific modeling environment. For example, the study of [25] developed an ecore meta-model and Xtext DSL to provide a basic knowledge of model-driven engineering by specifying the concepts of IoT systems. Therefore, such research studies are placed under the General category.

B. REVIEW PROTOCOL

The review protocol is comprised of six activities: 1) background and research questions, 2) inclusion and exclusion criteria, 3) search process, 4) quality assessment, and 6) data extraction and synthesizing. The first two activities have already been performed in Section I. The remaining activities are performed in the subsequent sections.

1) INCLUSION AND EXCLUSION CRITERIA

The third and foremost activity of the review protocol is to define the inclusion and exclusion criteria for selecting and

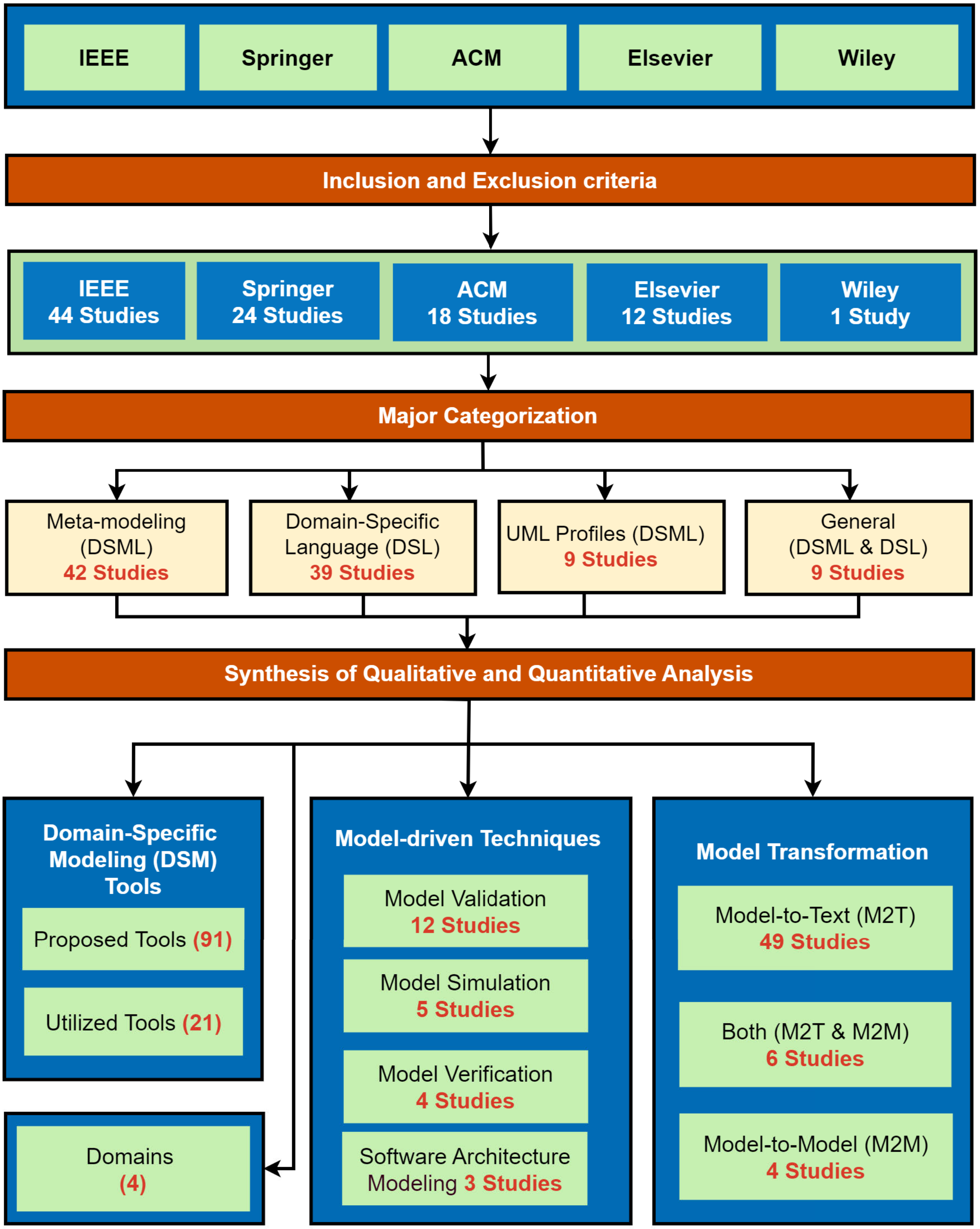

FIGURE 1. Overview of systematic literature review.

rejecting the selected studies. To analyze research studies, we describe five selection parameters as presented below: -

a) **Subject**: Research studies are only selected if they are highly relevant to the DSM and support the answers to our seven defined research questions. The selected studies must belong to one of our pre-defined categories. *Description*: Various research papers have been published in the context of MDE. Only those studies that are relevant to the defined categories are selected. All other studies are rejected.

b) **Publication Year**: The research studies are selected from Jan 2021 to Aug 2023.

Description: We extracted the latest research studies using a filter of publication year, i.e., Jan 2021 to Aug 2023. All research studies published before Jan 2021 are discarded.

c) Publisher: Research papers are selected only if they fall under five scientific databases, i.e., IEEE, Elsevier, Springer, ACM, and Wiley.
Description: The research papers are published in different scientific databases. We specified five scientific databases globally used for research publications for the selection criteria. We selected the research studies published in any of the specified scientific databases; otherwise, we rejected them.

d) Application Research: Only those research studies are selected in which any DSM framework, technique, or tool is proposed or utilized.
Description: In the context of MDE, various grey literature-based research papers have been published, i.e., theoretical frameworks, review papers, position papers, etc. [26]. For instance, the study of [27] is classified as a position paper in healthcare systems. Therefore, all such research studies are discarded where vague statements or studies of grey literature are specified, and a genuine framework or tool is not proposed or utilized.

e) Repetition: Comprehensive analysis is performed to discard research articles with similar content.
Description: Various research studies have similar outcomes—particularly research studies listed in different scientific databases. First, several studies with related research contents are analyzed, and then the study with the most trustworthy content is selected.

2) SEARCH PROCESS

We initiated the search process by selecting five databases (as shown in Table 3) and utilizing various search terms (as shown in Table 1) to extract relevant studies based on our inclusion and exclusion criteria (Section-B). We performed the search process with operators: "AND" and "OR". However, such operators return extensive research studies that are unfeasible to analyze exhaustively. Therefore, we performed the optimum search process with advanced search options provided by the databases, e.g., "where the title or abstract contains". After applying such advanced options or filters, we fetched the relevant research studies that could be thoroughly analyzed. Using search terms like "Model driven engineering," we obtained 123 results from the IEEE database, as given in Sr. #1 of Table 1, for the publication year range 2021 to 2023. We analyzed the fetched research studies and identified that a few keywords are closely correlated with the "Model driven engineering" keyword, which could be utilized to find more relevant research studies. For example, we identified terms like "meta-modeling", "Domain Specific language", "Domain Specific modeling, and "UML profile". Therefore, we utilized these terms to perform an advanced search process, as given in Sr. # 2 to 5 of Table 1. These advanced search terms helped us to identify the relevant research studies, which are hard to identify with the basic search terms. Lastly, we selected 99 research papers, as shown in Figure 2 by following the specific rules:

TABLE 1. Search terms and their corresponding filtered results.

Sr.#	Keywords	IEEE	Elsevier	Springer	ACM	Wiley
1	Model driven engineering	123	325	239	473	63
2	meta-modeling	245	102	178	84	29
3	Domain Specific language	174	94	125	167	28
4	Domain Specific modeling	9	47	88	91	19
5	UML profile	8	5	3	6	1

TABLE 2. Year-wise distribution of selected studies.

Sr.#	Publication Year	Studies	Total
1	2021	[28], [29], [17], [30], [25], [31], [32], [24], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52], [53], [54], [55], [56], [57], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67], [68], [69], [70], [71], [72], [73], [74], [75], [76], [77]	53
2	2022	[78], [79], [80], [81], [82], [83], [84], [85], [86], [87], [88], [21], [89], [90], [91], [92], [93], [94], [95], [96], [97], [98], [99], [100], [101], [102], [103], [104], [105], [106], [107], [108], [109], [110], [111], [112], [113], [114], [115], [116]	40
3	2023	[20], [117], [118], [119], [120], [18]	6

- Overall, we analyzed the initial 2726 research results. Initially, we rejected the 1500 research papers by only analyzing their titles.
- In the next step, we rejected the 757 research studies by analyzing the abstracts of the 1226 remaining research studies.
- In the next step, we analyzed the remaining 469 studies and rejected 311 research studies that violated the application research.
- After analyzing the remaining 158 research studies, we selected 99 studies that fully complied with our selection criteria.

3) QUALITY ASSESSMENT

We defined inclusion and exclusion criteria that supported the selection of high-impact studies and ensured the reliability of

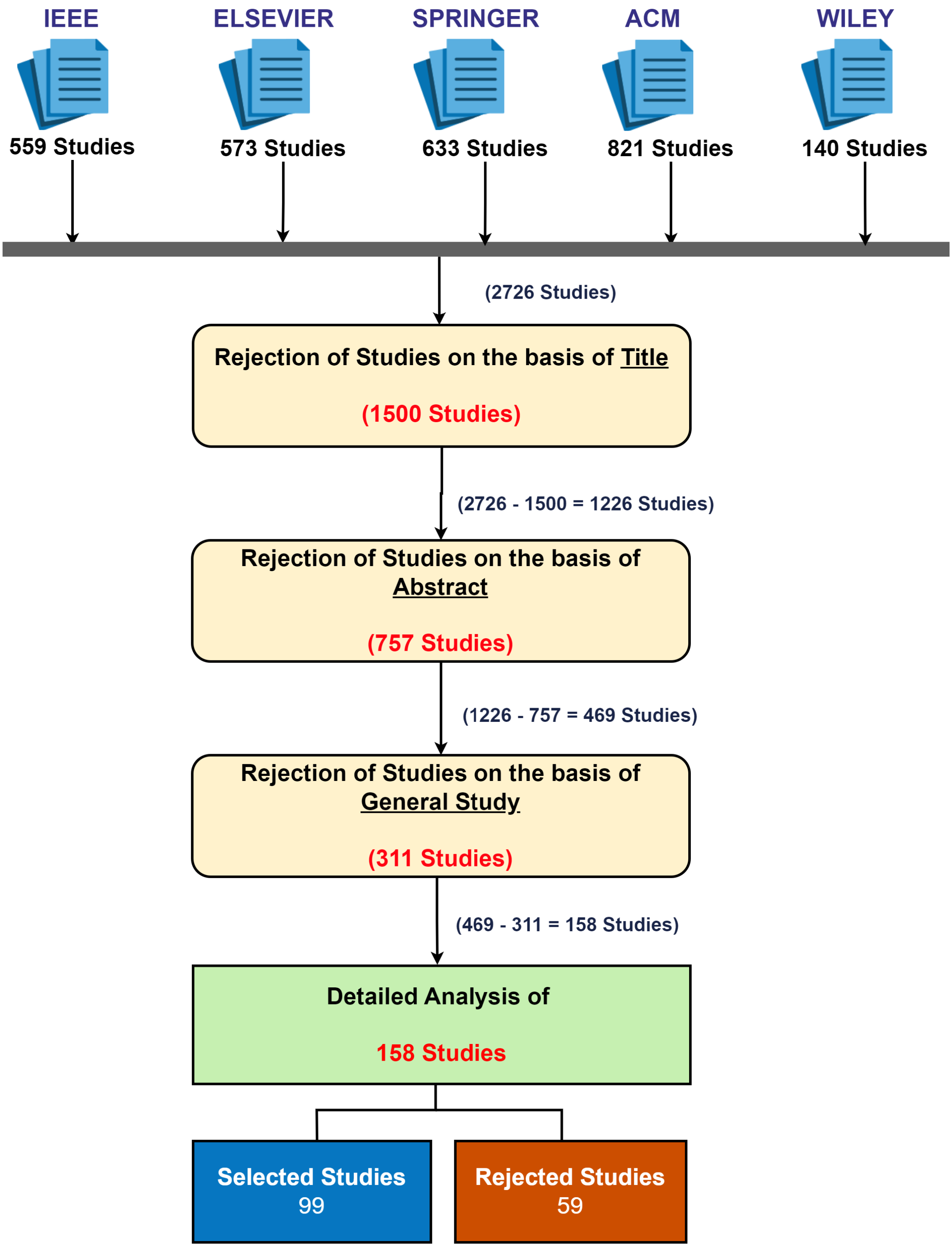

FIGURE 2. Search process.

the SLR. The fourth parameter ensures the selection of studies where a framework or tool is proposed or utilized. In the context of MDE, domain-specific modeling (DSM) tools are used to carry out the design and development process. Furthermore, the second parameter determines the selection of the latest research studies. It leads to identifying current MDE trends to support efficient system development. The year-wise distribution of the selected research studies is given in Table 2. We identified 53, 40, and 6 research studies published in 2021, 2022, and 2023, respectively.

TABLE 3. Summary of selected research studies w.r.t scientific databases and publication type.

Sr.#	Scientific Database	Type	Studies	Total
1	IEEE	Journal	[41], [42], [44], [45], [91], [54]	6
		Conference	[29], [17], [81], [80], [30], [25], [31], [82], [32], [83], [84], [85], [24], [33], [34], [35], [86], [36], [37], [88], [38], [39], [89], [43], [49], [50], [94], [95], [58], [67], [69], [70], [71], [72], [73], [75], [76], [77]	38
2	Springer	Journal	[117], [118], [47], [92], [55], [97], [98], [56], [103], [59], [60], [104], [66], [112], [113], [74], [116], [120], [18]	19
		Conference	[87], [90], [96], [100], [102],	5
3	ACM	Journal	Nil	0
		Conference	[21], [51], [53], [99], [57], [61], [105], [62], [106], [107], [108], [64], [65], [109], [110], [111], [114], [115]	18
4	Elsevier	Journal	[78], [79], [28], [20], [40], [46], [119], [48], [52], [63], [68]	11
		Conference	[101]	1
5	Wiley	Journal	[93]	1
		Conference	Nil	0

We considered five (5) high-quality scientific databases for selecting research studies to comply with the third parameter of the selection criteria (Section II-B1). The summary of the selected studies distribution is presented in Table 3. We selected 44, 24, 18, 12, and 1 research study (Journal + Conference) from IEEE, Springer, ACM, Elsevier, and Wiley, respectively. It is observed that IEEE is the significant scientific database in which most of the research studies have been published.

Moreover, the publication type for each selected research study is also analyzed. Although we tried to find as many journal studies as possible, we only found 37 that complied with the selection criteria. Consequently, we identified 63% (62 research studies) of the overall research articles published as conference publications and 37% of the overall studies (37 research studies) published as journal articles. We identified a study [102] that is published as a book chapter and considered it under the category of conference publication.

4) DATA EXTRACTION WITH SYNTHESIS

We have performed data extraction to find answers to the seven research questions (Section I) by first defining a data extraction template to extract the required information from the selected studies, as in Table 4. The bibliographic information is mainly extracted to analyze each selected study's main elements, as given in Sr. #1 of Table 4. Furthermore, the core concepts of each selected study are extracted and required to answer the RQs, as shown in Sr. #2 to Sr. #7 of Table 4. Finally, a comprehensive analysis is carried out to provide answers to the research questions.

III. RESULTS AND ANALYSIS

This section presents the comprehensive results obtained from the conducted SLR. The results are organized and presented according to the predefined inclusion and exclusion criteria, ensuring the relevance and quality of the selected studies. Table 5 provides a detailed categorization of the selected studies based on the established criteria. This categorization helps us to understand various aspects of the domain.

Table 6 highlights the system domains effectively explored through domain-specific modeling activities. This information will help identify the application areas where domain-specific modeling has been widely adopted.

In Tables 7, 8, 9, 10 and 11, an overview of the tools and their specific applications in the context of domain-specific modeling is provided. Additionally, Tables 12 and 13 shed light on the model-transformation types and offer valuable information on the techniques used to enhance software development through model-driven practices. Lastly, Table 14 highlights the global impact of MDE through the involvement of various regions in this research area.

A. CATEGORIZATION

Section II-A defined four categories based on domain-specific modeling (DSM) approaches to classify the selected studies. These categories include 1) Meta-modeling, 2) Domain-Specific Language, 3) UML Profile, and 4) General. In this section, the selected studies are classified based on the implementation languages of the categories defined in Section II-A. The results of the classification are presented in Table 5.

The first and second columns of Table 5 represent the main categorization and sub-categorization, respectively. It can be seen from the table that DSM can be classified into two types: graphical and textual modeling. Graphical modeling includes meta-modeling and UML Profiles, while textual modeling is based on DSL techniques. Thirdly, a general category combines both graphical and textual modeling.

During the analysis, we identified 42 studies related to meta-modeling (DSM), 39 on Domain-specific languages (DSLs), 9 studies targeting UML Profile, and 9 on the general category dealing with both DSM and DSL. Our findings indicate that Ecore is the most used modeling language in the meta-modeling category, appearing in 39 selected studies. Xtext, on the other hand, emerges as one of the most utilized textual DSL, employed in 25 studies. In the general category, we found five more studies using textual (Xtext) and graphical (Ecore) approaches.

TABLE 4. Elements of data extraction & synthesis.

Sr.#	Data	Details	Purpose
1	Bibliographic Information	Publication year, Publisher, and publication type (i.e., Conference or Journal) are analyzed.	Documentation
		Data Extraction with Synthesis	
2	Categories	Classification of Selected research studies. Results are placed in Table 5.	RQ1
3	Domains	Analysis of research studies is carried out to determine the target domains. Results are presented in Table 6.	RQ2
4	Tools	Analysis of research studies determines the utilized and proposed tools. The results are summarized in Section-C (Utilized tools are given in Table 7), and (proposed tools are given in Table 8, 9, 10, and 11).	RQ3
5	Techniques	Analysis of research studies is carried out to explore the leading MDE techniques. Results are presented in Table 12.	RQ4
6	Model Transformations	Model Transformations approaches utilized in selected research studies. Results are presented in Table 13.	RQ5
7	Regions	Analysis of research studies is conducted to determine the participating regions. Results are presented in Table 14.	RQ6
8	Challenges	Research studies are analyzed to identify the challenges involved in the DSM research.	RQ7

TABLE 5. Categorization of selected studies based on DSM approaches and their supported languages - frameworks.

Sr.#	Category	Sub-category	Reference	Total
1	Meta-modeling (DSM-Graphical modeling)	Ecore	[78], [79], [28], [17], [81], [31], [82], [32], [83], [84], [33], [87], [36], [37], [38], [39], [41], [89], [90], [43], [44], [91], [47], [119], [92], [50], [97], [99], [56], [57], [103], [106], [63], [67], [112], [70], [74], [120], [75]	39
		Adoxx	[118], [18]	2
		OCCI	[42]	1
2	Domain-Specific Language (DSL-Textual modeling)	Xtext	[20], [30], [88], [46], [51], [52], [53], [94], [95], [55], [96], [98], [101], [58], [61], [107], [108], [64], [65], [66], [109], [110], [71], [72], [115]	25
		MPS	[21], [117], [49], [54], [105], [68], [69], [114]	8
		PyEcore	[85], [34], [77]	3
		Epsilon	[116], [48]	2
		Droid	[111]	1
3	UML Profile (DSM-Graphical modeling)	-	[29], [24], [45], [102], [59], [60], [104], [62], [113]	9
4	General modeling (DSM & DSL – Both graphical & Textual modeling)	Ecore & Xtext	[25], [35], [93], [73], [76]	5
		Ecore & MPS	[80], [100]	2
		Ecore & UML Profile	[86]	1
		Xtext & UML Profile	[40]	1

These findings highlight the prevalence of Ecore and Xtext as prominent languages for graphical and textual modeling, respectively, in the context of DSM. This categorization provides valuable insights into utilizing specific languages and their applications across the selected studies.

B. DOMAIN AREAS

We analyzed the selected studies to determine the system domains where the DSM approaches are appropriately exploited. Table 6 presents an overview of the system domains. Overall, we identified 4 major domains by simply classifying the selected studies into the various subdomains where DSM approaches had been employed. These four significant domains are Embedded systems, Information technology systems, ERP, and general systems. Analysis revealed that the embedded system is a primary leading domain in which DSM has significantly contributed to 57 research studies. Embedded systems incorporate hardware and software to perform real-time operations, enabling the development of a large-scale architecture with microprocessor controllers, sensors, actuators, etc. A cyber-physical system is one of the leading subdomains of

TABLE 6. System domains.

Sr.#	Major Domain	Subdomain	Reference	Total
1	Embedded System	Cyber-physical system	[29], [17], [35], [40], [43], [45], [96], [98], [102], [60], [104], [105], [106], [63], [64], [66], [69], [70], [71], [72], [73], [116]	22
		IoT System	[30], [25], [82], [86], [38], [41], [117], [118], [91], [119], [54], [103], [61], [18]	14
		Mobile Application	[37], [89], [44], [47], [56], [108]	6
		Automotive systems	[90], [92], [93], [53], [94], [99]	6
		HealthCare System	[32], [83], [84], [21], [120]	5
		Digital Electronics System	[88], [65]	2
		Gamified Simulation System	[114], [76]	2
2	Information Technology System	Cybersecurity	[20], [33], [36], [39], [46], [49], [67]	7
		Data Analytics Systems	[79], [81], [85], [52], [75], [77]	6
		Web Application System	[92], [51], [59], [107]	4
		Cloud Computing Systems	[42], [58], [110]	3
		Geographical Information System	[97], [62], [74]	3
		Recommender System	[109], [111]	2
		Telecommunication System	[28], [34]	2
		Scientific Computing Application	[55]	1
		Quantum Information System	[113]	1
3	General	-	[80], [31], [87], [50], [95], [100], [57]	7
4	Enterprise Resource Planning (ERP) System	Business Process System	[24], [48], [101], [68], [112], [115]	6

the embedded system, wherein 22 research studies have mostly incorporated autonomous-based systems. Further, the Information technology system is the second leading domain that employs a combination of hardware, software, data resources, and networks to perform computations, analyze the data for decision-making, and securely store confidential information, resulting in 29 domains. Seven research studies have underscored cyber-security systems as a significant subdomain of information technology systems, incorporating advanced methods to secure confidential information. On the other hand, general systems are one of the third leading domains in which seven research studies have validated the proposed tool by considering the generic systems, representing the subdomain with a dash sign in Table 6. Lastly, six research studies have shown that ERP systems can provide a mechanism to manage business operations, such as supply chain, logistics, etc. Further domain analysis is specified in Table 6.

C. MODEL-DRIVEN TOOLS

This section summarizes the model-driven tools utilized and proposed by the researchers in the selected studies to support the domain-specific modeling (DSM) approaches. The tools used by the researchers in the studies chosen for implementing model-driven approaches/techniques are presented in Section III-C1. Whereas the tools proposed in the selected studies are given in Section III-C2.

1) TOOLS UTILIZED IN VARIOUS STUDIES

It is essential to identify the existing tools utilized in the selected studies to support various model-driven activities. This will help the researchers choose the right tool per their requirements. We identified 21 tools from the selected studies as presented in Table 7 with the five parameters provided below: **1) Approach** is the primary function of the tool **2) Tool Name** is the name of the tool **3) Availability** shows that either tool is publicly available or not **4) Reference** of the studies where the tool is utilized, is provided for further details **5) Total** represents the number of studies in the tool is utilized.

It is analyzed that among 21 tools, 5 are based on the meta-modeling approach, 5 are DSL tools, one is based on a UML profile, 3 are graphical editor tools, and 7 are verification tools. Meta-modeling tools identified from the selected studies are Eclipse-EMF, Obeo Designer, Adoxx, OCCI Aware Studio, and Exeed. However, the table shows that Eclipse-EMF is the leading meta-modeling-based tool utilized in 27 studies, and Obeo Designer is the second most used tool in 13 studies. We identified 5 DSL tools from the selected studies, including Xtext, JetBrains MPS, Pyecore, Droid, and Epsilon. Xtext and JetBrains MPS are the most frequently utilized DSL tools, with 31 and 10 studies, respectively.

Furthermore, we analyzed the selected studies to identify the graphical editors. We identified three graphical editors

TABLE 7. Summary of utilized tools in the selected studies.

Sr.#	Approach	Tool Name	Availability	Reference	Total
1	Meta-modeling	Eclipse-EMF	EPL	[78], [79], [28], [81], [80], [25], [86], [87], [38], [41], [43], [44], [119], [92], [93], [99], [56], [100], [103], [106], [87], [63], [67], [70], [73], [75], [76]	27
		Obeo-Designer	EPL	[17], [31], [82], [32], [83], [84], [33], [36], [37], [39], [91], [97], [57],	13
		Adoxx	Proprietary	[118], [18]	2
		OCCI Aware Studio	EPL	[42]	1
		Exeed	EPL	[47]	1
2	DSL	Xtext	EPL	[20], [30], [25], [35], [88], [40], [46], [93], [51], [52], [53], [94], [95], [55], [96], [98], [101], [58], [61], [107], [108], [64], [65], [66], [109], [110], [71], [72], [73], [115], [76]	31
		JetBrains MPS	EPL	[80], [21], [117], [49], [54], [100], [105], [68], [69], [114]	10
		Pyecore	EPL	[85], [34], [77]	3
		Epsilon	EPL	[48], [116]	2
		Droid	EPL	[111]	1
3	UML Profile	Papyrus	EPL	[29], [24], [86]. [40], [45], [59], [60], [104]	8
4	Graphical	Sirius	EPL	[78], [28], [17], [25], [31], [82], [32], [83], [84], [33], [35], [36], [37], [88], [38], [39], [89], [90], [43], [91], [119], [97], [57], [103], [106], [67], [112] [72], [73], [74], [120]	31
		Eugenia	EPL	[41], [48], [56]	3
		Picto	EPL	[92], [73]	2
5	Verification	Upaal	EPL	[69], [70]	2
		Rodin	EPL	[29].	1
		NumsV	EPL	[104]	1
		ProB	EPL	[61]	1
		Ocra & Kratos	EPL	[99]	1
		Alloy	EPL	[46]	1

from the selected studies: Sirius, Eugenia, and Picto. However, Sirius is the most utilized tool in 31 studies, Eugenia is the second leading tool in 3 studies, and Picto is used in only two. We also identified seven verification tools in a few research studies: Uppaal, Rodin, Prob, NusmV, Ocra (Othello Contracts Refinement Analysis), Kratos, and Alloy. Specifically, [29] utilized the Rodin tool, and [69], [70] utilized the Uppaal for the verification of automotive systems. Similarly, [99] employed Kratos and Ocra for the code verification of the automotive systems. The study of [61] used the Prob tool for verifying the health-pandemic systems. Similarly, [46] utilized the Alloy tool to verify embedded systems. Also, the study of [104] used the NuSMV tool to verify security applications.

2) TOOLS PROPOSED IN VARIOUS STUDIES

Various tools have been proposed in the selected studies to support DSM approaches. We comprehensively analyzed the selected studies to identify the proposed tools. These tools are summarized in Tables 8, 9, 10, and 11 with the following parameters: **1) Tool Name 2) Frameworks** based on which the tool is proposed **3) Open Source** or not **4) Empirical Evaluation** conducted or not **5) Usability** addressed or not **6) Reference** of the study in which the tool is proposed.

Some research studies have not defined their tool names, so those studies are illustrated with a dash sign. Firstly, open source represents whether it is a public project (represented by the tick sign) or private project (represented by the cross sign), or no information about its availability is provided (represented by the dash sign) in the research study. Secondly, empirical evaluation demonstrates whether the tools are rigorously evaluated with real-time participants (represented by the tick sign) or not (represented by the cross sign). Thirdly, the usability parameter demonstrates whether the tool has been comprehensively addressed with supporting documents such as manuals and documentation (represented by the tick sign) or not (represented by the cross sign). We classified each tool and devised four categories based on DSM approaches. The description of each DSM-categorized tool is defined in the subsequent section:

a) **Metamodeling based Tools:** The metamodeling tools are visual editors or graphical frameworks, like Eclipse Modeling Framework (EMF), that can define the abstract syntax through metamodeling constructs (relationships, attributes, enumerations) with a drag-and-drop mechanism and enable the concrete visualization of M1-level models with graphical iconic representation [15]. Such metamodeling tools build the foundation

TABLE 8. Meta-modeling-based Tools.

Sr.#	Tool Name	Meta-modeling Frameworks					Open source	Empirical Evaluation	Usability	Ref
		Obeo Designer	Eclipse EMF	Exeed	Adoxx	OCCI Aware Studio				
1	COMORP		✓				×	✓	×	[78]
2	-		✓				×	×	×	[79]
3	-		✓				×	×	×	[28]
4	AVRP	✓					×	×	×	[17]
5	EMM		✓				✓	×	×	[81]
6	-	✓					-	×	×	[31]
7	-	✓					-	×	×	[82]
8	MBN	✓					×	×	×	[32]
9	Optic Ally	✓					-	×	×	[83]
10	MD-BA	✓					-	×	×	[84]
11	MDAP	✓					×	×	×	[33]
12	-	✓					-	×	×	[36]
13	-	✓					-	×	×	[37]
14	DSL4ESP		✓				-	×	×	[38]
15	-	✓					×	×	×	[39]
16	Simulate IoT		✓				-	×	×	[41]
17	MODEMO					✓	✓	×	×	[42]
18	-		✓				×	×	✓	[43]
19	X-IOT				✓		✓	×	✓	[118]
20	MBUID		✓				-	×	×	[44]
21	Monitor-IOT	✓					-	✓	×	[91]
22	MAndroid			✓			×	✓	×	[47]
23	SI4IOT		✓				✓	✓	✓	[119]
24	MORE		✓				✓	×	✓	[92]
25	SUIC	✓					-	×	×	[97]
26	-		✓				-	×	×	[99]
27	ALBA		✓				✓	✓	✓	[56]
28	BOMO	✓					×	×	×	[57]
29	HealMA		✓				-	✓	×	[103]
30	MoDLF		✓				✓	×	✓	[106]
31	RoBMEX		✓				✓	×	✓	[63]
32	Dual-XACML		✓				-	×	✓	[67]
33	-		✓				-	×	×	[70]
34	DTMN				✓		✓	×	✓	[18]
35	-		✓				-	×	×	[75]
36	PSS		✓				✓	×	✓	[87]

of DSM and can assist the modelers in creating meta-models with modular designs, ensuring extensible and consistent architecture through constraint validation [121]. By analyzing the selected studies, we identified 36 meta-modeling tools proposed in the selected studies as presented in Table 8. This table provides insight into various tools proposed by researchers based on one of the five meta-modeling frameworks, including Obeo Designer, Eclipse EMF, Exeed, Adoxx, and OCCI Aware Studio. However, the researchers utilize Obeo Designer and Eclipse EMF to propose new tools in 14 and 18 studies. It is essential to mention that some research studies did not provide any information about the meta-modeling framework to achieve their implementation objective. For example, R. Veloso et al. [74] utilized the ecore meta-modeling language to support geographical information systems. Similarly, B. V, Acker, et al. [50] utilized the ecore meta-modeling language for the design manufacturing of general systems. Neither of these research studies has provided any information about the utilized meta-modeling framework. Therefore, information about

TABLE 9. Domain-specific language (DSL) based tools.

Sr.#	Tool Name	DSL Frameworks					Open Source	Empirical Evaluation	Usability	Ref
		Xtext	JetBrains MPS	Droid	Pyecore	Epsilon				
1	MDSL	✓					✓	×	✓	[20]
2	UCM4IOT	✓					-	×	×	[30]
3	-				✓		×	×	×	[34]
4	DSysML	✓					-	×	×	[88]
5	ECLang		✓				-	✓	×	[21]
6	-		✓				✓	✓	✓	[117]
7	-	✓					×	×	×	[46]
8	CEPChain					✓	×	×	×	[48]
9	DIDoc		✓				-	×	×	[49]
10	-	✓					✓	×	✓	[51]
11	SWQL	✓					✓	✓	✓	[52]
12	-	✓					×	×	×	[53]
13	AdaptiveVLE		✓				×	✓	×	[54]
14	MoSt	✓					✓	×	✓	[94]
15	-	✓					✓	×	×	[95]
16	-	✓					×	×	×	[55]
17	-	✓					×	×	×	[96]
18	ML-Quadrat	✓					✓	✓	✓	[98]
19	EventB2PDDL	✓					-	×	×	[101]
20	MDEForgeWL	✓					✓	×	✓	[58]
21	-	✓					×	×	×	[61]
22	DRESS-ML		✓				✓	×	✓	[105]
23	-	✓					✓	×	✓	[107]
24	EGen	✓					✓	✓	✓	[108]
25	ALIA	✓					-	×	×	[64]
26	-	✓					-	×	×	[65]
27	AsyncAPI	✓					✓	✓	✓	[66]
28	LEV4REC	✓					✓	✓	✓	[109]
29	IoTECS	✓					✓	×	✓	[110]
30	-			✓			✓	×	✓	[111]
31	-		✓				-	×	×	[68]
32	-		✓				✓	×	×	[69]
33	-	✓					✓	×	×	[71]
34	MDE4ROS	✓					-	×	×	[72]
35	ViMoTest		✓				✓	×	✓	[114]
36	ModAL	✓					✓	✓	✓	[115]
37	ATLAS					✓	✓	✓	✓	[116]
38	-				✓		×	×	×	[85]
39	-				✓		✓	×	✓	[77]

the meta-modeling framework used for such research studies is not provided in Table 8.

b) **DSL Based Tools:** The DSL-based tools are textual editors that allow DSL development using language invention patterns by following the sophisticated syntax of various grammar constructs. The DSL tools enable syntax highlighting and custom data validation for static error checking [15], [122]. Few DSL tools have embedded support of the EMF (Eclipse-Modeling Framework), while others have a projectional editor. EMF-based DSLs (e.g., Xtext) can generate the meta-models by incorporating meta-definitions of Extended Backus-Naur Form (EBNF) grammar at the M2 level and developing a language-instance model at the M1 level. On the other hand, projectional editors (e.g., JetBrains MPS) merely support the generation

TABLE 10. UML profile-based tools.

Sr.#	Tool Name	Papyrus	Open Source	Empirical Evaluation	Usability	Ref
1	-	✓	-	×	×	[29]
2	UMLPBPMN	✓	-	×	×	[24]
3	IoTsecM	✓	×	×	×	[45]
4	-	✓	-	×	×	[59]
5	IQCPSoS	✓	✓	×	✓	[60]
6	MBIPV	✓	✓	✓	×	[104]
7	-	✓	✓	×	×	[113]

of language-instance models [121]. From the selected studies, we identified 39 tools from the selected studies proposed by the researchers based on DSL frameworks, including Xtext, JetBrains MPS, Droid, Pyecore, and Epsilon, as given in Table 9. However, in 25 studies, researchers proposed Xtext-based tools. Some researchers performed implementation using the DSL frameworks and incorporated various validators. Just like M. Latifaj et al. [96] proposed a blended modeling tool using Xtext and validated it with the Papyrus and Exeed editor to visualize the textual and graphical notations.

c) **UML Profile Tools:** UML Profiles tools, like Papyrus, facilitate the creation of customizable UML models with advanced notations, such as meta-classes and stereotypes, through the use of drag-and-drop functionality [123]. These tools enable graphical visualization and constraint validation, similar to meta-modeling tools that define the semantics and structure of modeling concepts. In 7 research studies, researchers proposed tools based on UML Profile by utilizing the Papyrus tool as presented in Table 10. It is analyzed from the selected studies that most studies did not provide any information about the modeling tool for implementation objectives. For example, L. Douglas et al. [102] modeled the safety scenarios of embedded systems based on UML Profiles. Similarly, Z. Hayat et al. [62] performed the UML profile-based modeling of the geographical information system. However, since neither research study provided any information about the modeling tool for the implementation objectives, they are not presented in Table 10.

d) **General Modeling Tools:** The tools combine DSM approaches, including both DSML and DSL, enabling users to perform all operations in a single modeling environment, making them more versatile and powerful. From the selected studies, we identified 9 studies where researchers proposed new tools by utilizing two or more frameworks based on Meta-modeling, DSLs, or UML Profiles as provided in Table 11. Among 9 studies, four proposed tools based on Eclipse EMF and Xtext. Therefore, it is concluded that EMF and Xtext are the frameworks most utilized by researchers to propose tools based on meta-modeling and DSLs.

D. MDE TECHNIQUES

We selected four model-driven techniques representing the primary aspects of domain-specific modeling (DSM) research and development. These techniques are validation, verification, simulation, and software architectural modeling. Validation techniques ensure that the models serve the intended purpose and influence the intended applications. On the other hand, verification techniques ensure the DSM models align with the system requirements and specifications. Simulation techniques are used to test DSM models dynamically and provide valuable insights into the system's behavior under numerous conditions. Finally, software architectural modeling techniques are crucial in designing and organizing DSM models to simplify system development and maintenance. Our objective is to comprehensively estimate the utilization and effectiveness of different model-driven approaches in DSM by focusing on these techniques. This approach provides a comprehensive understanding of how these techniques are applied, including their strengths, limitations, and impact on DSM research and practice over the past two years. In the subsequent section, these techniques are described in more detail and presented in Table 12.

1) VALIDATION TECHNIQUE

The validation technique includes a standard mechanism for checking the model's conformance against consistency rules to ensure the model's correctness. Various validation languages can be employed to ensure the development of a well-formed model by integrating the constraint validation rules [124], [125]. Analysis of the selected studies revealed that validation is one of the most utilized MDE techniques employed in 12 studies. Furthermore, we analyzed the selected studies to identify the languages used for model validation and identified two validation languages: OCL (Object Constraint Language) and EVL (Epsilon Validation Language). The Eclipse platform supports OCL, whereas the Epsilon platform supports the EVL. It is observed from Table 12 that OCL is the most utilized language for model validation in 9 studies, whereas EVL is used in merely three studies. For example, the study of [78] designed a database-cost modeling framework and utilized the OCL constraints to validate the model with the semantics of MathML. Similarly, the author of study [79] used OCL constraints to validate the ontology-based system to support data integration. Both studies of [25] and [41] developed a well-structured IoT system by leveraging the OCL constraints for remote learning and simulation environments. Similarly, in [92], the authors used the EVL to specify constraints for checking the conformance of models against architectural

TABLE 11. General modeling-based tools.

Sr.#	Tool Name	Utilized Meta-Modeling Tools		Utilized UML Profile Tool	Utilized DSL Tools		Open source	Empirical Evaluation	Usability	Ref
		Obeo Designer	Eclipse EMF	Papyrus	Xtext	JetBrains MPS				
1	-		✓			✓	✓	×	×	[80]
2	-		✓		✓		✓	✓	✓	[25]
3	CARES		✓		✓		✓	×	✓	[35]
4	UMLPDI3OT	✓		✓			-	×	×	[86]
5	-			✓	✓		✓	✓	✓	[93]
6	CaMCOA		✓		✓		×	×	×	[73]
7	-		✓		✓		✓	×	✓	[76]
8	-			✓	✓		×	✓	✓	[40]
9	-		✓			✓	✓	×	✓	[100]

TABLE 12. Model-driven techniques utilized in the selected studies with their supported languages.

Sr.#	Techniques	Supported language	References	Total
1	Validation Techniques	OCL	[78], [79], [28], [25], [41], [90], [56], [60], [76]	9
		EVL	[48], [92], [73]	3
2	Simulation Technique	Agent-Based modeling Language	[21], [120]	2
		Hardware-In-The-Loop Simulation	[53]	1
		Functional-Mockup Interface	[35]	1
		Discrete Event Simulation	[28]	1
3	Verification Techniques	Event-B	[29], [61]	2
		Alloy formal model	[46]	1
		Z-Notation	[95]	1
4	Software architectural Modeling	Business Process Modeling Language	[24], [112]	2
		UML-RT Profiles	[96]	1

styles. Another study [73] used the EVL to detect errors in the early development phases of engine control systems.

2) SIMULATION TECHNIQUE

Simulation is a computational modeling technique used to generate the behavior of complex systems without manipulating the models. It provides a testing environment by simulating the model under different scenarios to identify errors through various simulation techniques [18]. In this SLR, we identified a limited number of research studies where simulation was performed, i.e., in 5 studies. Each research study employed a different simulation technique, including agent-based modeling language, discrete-event simulation, hardware-in-the-loop simulation, and functional-mockup interface (FMI) co-simulation.

In [21] and [120], the authors leveraged an agent-based modeling language to enrich the collaborative environments and bolster the traceability of models within the health domain. The authors of [53] utilized the hardware-in-the-loop simulation technique to simulate the behavior of automotive systems. Similarly, the study of [35] used the functional-mockup interface (FMI) co-simulation technique with their desired functions, i.e., doStep(), end (), etc., for simulating the heterogeneous embedded systems. Furthermore, the study of [28] developed a routing-based simulation framework with a discrete-event simulation technique and designed a well-formed model with the OCL constraint rules.

3) VERIFICATION TECHNIQUE

Verification is a technique that utilizes mathematical and logical methods to verify a system's correctness before deployment. The objective is to thoroughly detect any vulnerabilities in the model and ensure its reliability within the context of state-event transitions [126], [127]. In this SLR, we identified three formal verification techniques utilized in 4 different studies as presented in Table 12. These techniques, including Event-B, Alloy, and Z-notation, are used for verification scenarios. For example, the authors of [29] designed a formal verification approach using Event-B semantics for verifying the safety & security aspects of the automotive systems. Similarly, the study of [61] conducted verification of healthcare systems using Event-B semantics. They used a Prob model checker to verify the models through Event-B semantics. Furthermore, the study of [46] verified security systems against vulnerabilities using the semantics of the Alloy analyzer. The study of [95] used the Z-notation language by defining the requirements in natural language to support the formal modeling of complex software systems.

4) SOFTWARE ARCHITECTURAL MODELING

Software architectural modeling models the software architecture using the architecture description languages (ADLs).

TABLE 13. Model Transformations employed in the selected studies.

Sr.#	Transformation Type	Language used	References	Total
1	M2T	Acceleo	[79], [28], [17], [80] [82], [32], [84], [33], [35], [87], [37], [88], [38], [39], [41], [89], [90], [47], [119], [97], [103], [59], [106], [63], [109], [67]	26
		Xtend	[20], [30], [40], [46], [93], [53], [95], [55], [98], [101], [108], [64], [66], [110], [71], [72], [115]	17
		Epsilon Generation Language (EGL)	[48], [56], [65]	3
		TextGen	[117], [105]	2
		AdoScript Language	[118]	1
2	M2M	ATL	[51], [112]	2
		Epsilon Transformation Language (ETL)	[92]	1
		QVT-Operational	[96]	1
3	General (M2M & M2T)	Epsilon Transformation Language (ETL) & Epsilon Generation Language (EGL)	[73], [116]	2
		Henshin Language, ATL, Xtend	[25]	1
		ATL & Xpand	[44]	1
		ATL & Acceleo	[75]	1
		Henshin & QVT-operational & Xpand	[76]	1

Architectural description languages design the system blueprints through general-purpose modeling languages, such as UML [128]. However, UML-RT is also utilized to design the architecture of embedded systems [129]. From the selected studies, we identified two architectural description languages. In one study, the authors of [24] and [112] used business process modeling language (BPML) to model complex business processes and workflows. Secondly, the UML-RT (UML for Real-Time Systems) profile is utilized to attain the blended modeling framework for embedded systems [96]. Overall, BPMN is employed in 2 studies, while UML-RT is used in merely one research study.

E. MODEL TRANSFORMATION APPROACHES

Model transformation is one of the core concepts in model-driven engineering (MDE) that converts a model from one representation into another. Model transformation is crucial as it automates various tasks within the software development process, such as generating code and documentation from models or converting them into another representation for better understanding and visualization for multiple stakeholders [130]. We analyzed the selected studies to identify the model-transformation approaches in MDE. Analysis revealed that model transformation is of two types, i.e., Model-to-Text (M2T) and Model-to-Model (M2M). It is observed from Table 13 that M2T is the most utilized transformation approach in the selected studies as compared to the M2M transformation approach. Among 99 studies, 49 utilized the M2T approach, while 4 used the M2M approach.

1) MODEL-TO-TEXT (M2T) TRANSFORMATION APPROACH

The M2T approach transforms models into human-readable text or code. It is one of the most popular transformation approaches in MDE, as shown in Table 13. One of the core reasons for M2T being more prevalent among researchers and developers is that it shortens the software development lifecycle and reduces manual effort by automating the generation of code and documents from models, thereby ensuring consistency among software artifacts. Various languages have been developed to perform M2T transformation. We identified five M2T languages, including Acceleo, Xtend, EGL (Epsilon Generation Language), TextGen, and AdoScript, from the analysis of selected studies. The main difference between these languages is that Acceleo is a template-based language supported by OMG, while AdoScript is a command-based structure supported by the Adoxx tool. On the other hand, Xtend and TextGen have a Java-based structure within an Xtext and JetBrains MPS plugin, and EGL is a template-based language within an Epsilon plugin. Our Analysis of the selected studies revealed that Acceleo and Xtend are the two most utilized languages for generating textual artifacts from the model. Among 49 studies using the M2T transformation approach, 26 performed model transformation through Acceleo, while 17 employed Xtend to generate the desired textual artifacts. For example, Safdar et al. [106] proposed the automatic code generation from the Ecore meta-modeling using Acceleo for the embedded systems. Similarly, Riegler et al. [20] automatically generated Java code using Xtend to support database information systems.

2) MODEL-TO-MODEL (M2M) TRANSFORMATION APPROACH

The M2M approach supports the automatic transformation of models from one representation to another, allowing us to synchronize and evolve models at different abstraction levels, later facilitating code generation and documentation, etc., from models. M2M transformation is performed with the help of different languages. We analyzed the selected studies in this SLR to identify the M2M transformation languages. Consequently, we identified three languages: ATL, EGL, and QVT-operational. ATL and QVT-operational are entirely based on the Eclipse environment, whereas EGL is based on the Epsilon environment. It is observed from Table 13 that ATL is the most utilized M2M transformation language as it is employed in two studies. In contrast, both ETL and QVT-operational are used in merely one study.

3) GENERAL (M2T & M2M)

In this SLR, we identified a few studies (6 studies) where more than one M2T, M2M, or both M2T and M2M transformation approaches are utilized to generate an efficient system, as given in Table 13. The author of [73] proposed various model transformation techniques to validate the engine control systems. They used EGL to generate the code by complying with the Rolls-Royce coding standard. Further, they used ETL to test the behaviors of the engine controlling system in a simulation environment and generate the XML and Excel models to test the deployment environment. Similarly, the study of [116] used ETL to support the configuration of variations in health systems automatically. They also used EGL to generate algorithm code. Furthermore, the authors of [25] utilized the ATL to employ the out-place model transformations. Also, the authors used Henshin to conduct the in-place model transformation and utilized Xtend to generate the application code. Furthermore, [44] utilized ATL and Xpand to generate a concrete user-interface model and concrete-level code. Similarly, the authors of [75] performed the model-to-model transformation using ATL to demonstrate the machine-learning techniques and used Acceleo to generate the code of ubiquitous applications. Moreover, the authors of [76] performed the model transformation using the Henshin transformation rules to simulate the system behavior and utilized Xpand to generate code in two diverse languages. A few studies conducted the model transformations using different programming languages but did not specify any language. For example, L. Addazi et al. [40] proposed a blended modeling tool and performed the M2T and M2M using the Xtend and an unspecified language. Similarly, [54] generated the code using an unspecified programming language. Such studies where model transformations (i.e., M2T or M2M) are performed with software programming or undefined languages are excluded from Table 13.

F. REGION OF RESEARCH

We examined each selected study in detail to identify the regions that devoted significant efforts to utilize domain-specific modeling (DSM). A summary of the region-wise distribution is presented in Table 14, which provides insights into the global growth rate of DSM research. We have identified eight regions that significantly contribute to elevating the DSM research.

It can be seen from the table that Europe is a leading region and contributor to DSM research with 52 research studies. South Asia is the second emerging region, with 17 studies conducted overall. The Middle East is the 3rd contributing region on the list, where nine research studies were performed, and Northern America is the fourth contributing region, with five research studies. It is clear from the table that DSM approaches have been significantly explored in Europe during the last two years.

TABLE 14. Region-wise distribution of studies.

Sr.#	Region	Reference	Total
1	Europe	[79], [81], [80], [87], [40], [90], [92], [93], [53], [96], [100] [29], [35], [42], [46], [94], [72], [61], [63], [68], [21], [120], [54], [73], [116], [118], [18], [99], [58], [109], [60] , [41], [48], [52], [66], [111], [113], [98], [64], [114], [76], [115], [38], [50], [20], [25], [85], [34], [51], [74], [107], [49]	52
2	South Asia	[31], [32], [84], [24], [33], [86], [36], [37], [39], [95], [57], [62], [106], [83], [59], [55], [108]	17
3	Middle East	[89], [67], [47], [56], [103], [44], [82], [17], [77]	9
4	South America	[91], [119], [28], [117], [105], [112]	6
5	North America	[30], [65], [110], [45], [102]	5
6	North Africa	[88], [97], [78], [75], [101]	5
7	East Asia	[43], [104], [69], [70]	4
8	Siberia	[71]	1

IV. ANSWERS TO RESEARCH QUESTIONS

After conducting extensive research on the selected studies, we have obtained the necessary results, which are presented in Section III. These results allow us to provide comprehensive and valid answers to the research questions. Let's examine the answers to each research question:

RQ1: What notable studies significantly impact domain-specific modeling approaches between Jan 2021 and Aug 2023?

Answer: Overall, 99 studies relevant to MDE are identified and analyzed from five well-known databases, i.e., IEEE, Springer, ACM, Elsevier, and Wiley. The distribution of selected studies according to the scientific databases and the publication type is presented in Table 3. The distribution of the selected studies according to the publication year (2021-2023) is shown in Table 2. Of the 99 studies, 53 have been published in 2021, 40 in 2022, and 6 in 2023. From Table 2 and 3, it is analyzed that most of the studies were published in IEEE as conference papers in 2021. In addition, the categorization of the selected studies in the context of domain-specific modeling approaches is presented in Table 5. These categories include meta-modeling, domain-specific language (DSL), UML profile, and a general category for studies based on multiple approaches. Meta-modeling has been analyzed as a widely adapted domain-specific modeling approach followed by DSLs, utilized in 42(43%) and 39 (39%) research studies. These approaches can be performed through different frameworks. Each framework offers varying levels of tooling support and is suitable for various requirements. Table 5 shows that among 42 studies in the meta-modeling category, 39 studies (93%) are based on the Ecore modeling framework, whereas 25 studies (64%) in the DSL category are based on Xtext.

RQ2: In which major system domains have domain-specific modeling approaches been effectively employed and proven useful?

Answer: Domain-specific modeling approaches have been successfully applied in various domains. Overall, we identified 4 major system domains where DSM approaches have effectively confronted complex issues. The distribution of domains is specified in Table 6. Based on the results, the embedded system is the leading system domain where domain-specific modeling approaches have been utilized the most, with 57 studies. In contrast, Information technology systems are the 2nd leading domain with 29 research studies. Further details are given in Section III-B.

RQ3: What are essential, proposed, and utilized model-driven tools for software development covering graphical, transformational, and verification features?

Answer: Overall, we identified 21 tools utilized in the selected studies, as presented in Table 7. It is analyzed that among 21 tools, 5 are based on a meta-modeling approach, 5 are DSL tools, one is based on a UML profile, 3 are graphical editor tools, and 7 are verification tools. Analysis revealed that adequate tools are available in the existing literature to support domain-specific modeling and verification approaches. The results also show that Ecore is the most utilized meta-modeling tool, Xtext is the most common DSL tool, and Sirius is the leading graphical modeling tool used in most studies. Uppaal is the most frequently used verification tool. Further details are given in Section III-C. We also identified 91 tools proposed by the researcher in the selected studies. Among 91 tools, 36 tools are proposed based on meta-modeling approaches, 39 on domain-specific languages, and seven are associated with UML profile as presented in Table 8, 9, and 10 respectively. It can be seen from Table 11 that 9 tools are proposed by utilizing all three approaches, i.e., meta-modeling, DSL, and UML profile, or by combining any two of them. Analysis of the proposed tools has revealed that most are not open source. As a result, researchers and practitioners cannot customize and evaluate them further, considerably reducing their actual benefits. Further details are presented in Section III-C2.

RQ4: Which model-driven techniques have been primarily utilized in research conducted between 2021 and 2023?

Answer: We identified four model-driven techniques used in selected studies. The categorization of the selected studies based on these techniques is presented in Table 12. These techniques are Validation, Verification, Simulation, and Software architectural modeling. It is analyzed that 12 studies are based on validation techniques, four on verification, five on simulation, and three on software architecture modeling. These techniques are often performed through some formal languages. Analysis revealed that most studies have performed model validation through OCL (Object Constraint Language). OCL allows developers to define constraints and rules that models within a system must satisfy. These constraints range from simple data validation rules to complex business logic requirements. It is concluded that model validation through OCL is one of the most essential model-driven techniques, as models are the primary artifacts in model-driven engineering. Therefore, model validation is critical to ensure the correctness and quality of software models, improve communication among stakeholders, minimize rework, and ensure compliance with standards and regulations.

RQ5: What are the prominent model transformation approaches employed within domain-specific modeling?

Answer: We identified two prominent model transformation approaches in the selected studies, including Model-to-Model (M2M) and Model-to-Text (M2T), as in Table 13. M2M transformation involves converting one model representation into another, while M2T transformations focus on generating textual outputs, such as code or documentation, from models. It is analyzed that 49 studies (83%) utilized the Model-to-Text transformation (M2T) approach, and four studies (7%) are associated with the Model-to-Model (M2M) transformation approach. We also identified six studies (10%) that have used M2M and M2T approaches for the model transformation. The selected studies indicate that specific programming languages are commonly used for these model transformations. For M2T transformation, Acceleo is the widely used programming language with 26 studies (53%), followed by Xtend with 17 studies (35%). For M2M transformation, ATL is the most used programming language. It is concluded that M2T transformation is one of the significant model-driven approaches being utilized due to its ability to automate code generation, increase productivity, and reduce manual coding efforts.

RQ6: Which regions have emerged as leading contributors to the growth of domain-specific modeling approaches?

Answer: Overall, we identified eight regions that have contributed their efforts to the growth of domain-specific modeling approaches as represented in Table 14. It is analyzed that most of the studies (53%) are conducted in Europe, followed by South Asia (17%) and the Middle East (9%). The summary of contributing regions based on their contributions to the growth of DSM approaches is given in Table 14.

RQ7: What are the significant challenges and their recommended future directions in the DSM research?

Answer: Through our analysis of the selected studies, we identified four major challenges in the DSM research:

1) **Tools-Availability**: From the analysis of proposed tools in the selected studies, it is observed that 44% (40) of proposed tools are deployed as open source for accessibility. At the same time, the rest of the 24% (22) are publicly unavailable. On the other hand, 32% (29) of the proposed tools have no information available on the Internet, including their project information and source code. This unavailability becomes a prominent challenge for students, researcher collaborators, and

practitioners, hindering the validation and extendibility of those valuable tools. Therefore, these tools should be publicly available to promote their validation and extendibility.
2) **Collaborative Environment**: Our analysis revealed that Europe is the leading contributor where individuals and groups work together cooperatively to investigate, explore, and solve problems related to DSM. On the contrary, South Asia is the 2nd leading region where DSM research is being performed but not cooperatively. Instead, authors individually explore and conduct DSM research. Collaborative environments can lead to increased productivity, creativity, and overall success as they harness the collective capabilities of individuals. Therefore, researchers should work together collaboratively to improve the overall development of DSM research.
3) **Limited Empirical Evaluation**: Selected research studies have generally focused on developing an efficient DSM tool. However, most studies did not empirically evaluate their proposed tool, considering its learnability, processing time, etc. Analysis of proposed tools revealed that only 22% (20) had been empirically evaluated, indicating a significant gap in evaluating the DSM tool's efficacy. The empirical evaluation is an important criterion for ensuring the model's quality and a crucial measure for determining its usefulness to the public. Therefore, the concerned authors must evaluate the tool with at least a limited group of learners to evaluate their entire feature utilization.
4) **Usability**: Although the proposed tools have streamlined the development activities, they encountered some usability issues. Several research studies proposed tools based on advanced technologies requiring significant technical expertise to understand the integration of these technologies with DSM approaches and guide novice users in properly utilizing the tools. An analysis of these proposed tools revealed that only 38% (35) of them were accompanied by some supporting documentation, such as manuals or tutorials. Therefore, incorporating these tools into their workflows presents a daunting challenge for researchers. To mitigate this challenge, concerned authors must provide a comprehensive tool manual including all technical intricacies related to tool setup and usage for better tool understanding and utilization.

V. LIMITATIONS OF THE RESEARCH

Although we have followed the standard guidelines for conducting the systematic literature review (SLR), there are still some limitations present in this study:

A. LANGUAGE LIMITATION

The study focused on research studies published only in English. It is possible that relevant studies published in other languages, such as Spanish or Chinese, were not included in the analysis.

B. DATABASE SELECTION

We primarily utilized well-known databases such as IEEE, Elsevier, Springer, ACM, and Wiley to gather the research studies for the SLR. However, relevant studies could be published in other scientific databases, such as Scopus and PubMed, that were not included in the review.

C. SEARCH TERM SELECTION

Although we employed a comprehensive set of search terms and thoroughly analyzed the results, specific search terms returned many research studies that could not be thoroughly investigated, so we excluded a larger number of research studies where the paper title was not reflecting the content of the study. This may have resulted in the unintentional omission of some relevant studies.

Despite these limitations, this SLR's ultimate results are legitimate for the following reasons: 1) It is rare to find relevant studies in languages other than English. 2) We used the six most reliable databases, often publishing peer-reviewed, high-quality papers. As a result, even if a few relevant articles are missing from other databases, the general findings of this SLR are reliable and do not alter considerably.

VI. CONCLUSION AND FUTURE WORK

This study conducts a systematic literature review to analyze the latest studies to identify which model-driven techniques and domain-specific modeling tools are leveraged to resolve complex issues in various domains. This SLR analyzed the 99 studies published during the last two years, i.e., 2021 to 2023. It leads to the main categorization of the selected studies based on domain-specific modeling approaches, i.e., meta-modeling (42 studies), domain-specific language (39 studies), UML profile (9 studies), and general modeling (9 studies). Further, each selected study is analyzed to identify the target domains (4 major system domains) with the leading embedded system domain where approaches of domain-specific modeling have been utilized. Moreover, the identification of used tools is presented, i.e., meta-modeling tools (5), domain-specific modeling tools (5), UML profile (1), graphical instance modeling tools (3), and seven verification tools (7). Ecore is the leading meta-modeling tool, Xtext is the most used domain-specific tool, Sirius is graphical, and UPPAAL is a frequently utilized verification tool identified. We also identified 91 tools proposed by the researcher, i.e., meta-modeling tools (36), domain-specific languages (39), UML profiles (7), and general (9). In addition to this, model-driven techniques, including validation (12), simulation technique (5), verification (4), and software architectural modeling (3) have been identified. Validation is the most used MDE technique in the selected studies. The target type of model transformation employed in each selected study is also presented, i.e., model-to-text (49) and model-to-model (4) with their supported languages. Model-to-text transformation with Acceleo is the most utilized transformation type in the selected studies. Additionally, the identification of 8 contributing regions is also presented.

This research can be expanded in several ways in the future. One way to proceed would be to analyze the techniques identified in this SLR comprehensively. It is identified that the OCL is the most often utilized technique for model verification. It would be enlightening to explore other approaches for model verification. Several modeling and testing tools have also been identified in this SLR. So, a thorough comparison of these tools would be required to show their pros, cons, and applicability in a specific situation. This comparison will help researchers and practitioners choose the right language and tool for their needs.

REFERENCES

[1] M. Afonso, R. Vogel, and J. Teixeira, "From code centric to model centric software engineering: Practical case study of MDD infusion in a systems integration company," in *Proc. 4th Workshop Model-Based Develop. Comput.-Based Syst. 3rd Int. Workshop Model-Based Methodol. Pervasives Embedded Softw. (MBD-MOMPES)*, 2006, p. 10.

[2] M. Conrad, I. Fey, and S. Sadeghipour, "Systematic model-based testing of embedded automotive software," *Electron. Notes Theor. Comput. Sci.*, vol. 111, pp. 13–26, Jan. 2005.

[3] Mousami, "Model driven software engineering," *Int. J. Eng. Res. Technol.*, vol. 2, no. 3, Jul. 2018.

[4] R. Wuyts et al., "A logic meta-programming approach to support the co-evolution of object-oriented design and implementation," Ph.D. thesis, Program. Technol. Lab., Dept. Comput. Sci., Vrije Universiteit Brussel, Ixelles, Belgium, 2001.

[5] M. Lachgar and A. Abdali, "Modeling and generating the user interface of mobile devices and web development with DSL," *J. Theor. Appl. Inf. Technol.*, vol. 72, no. 1, pp. 124–132, 2015.

[6] R. Pohjonen and S. Kelly, "Domain-specific modeling," *Dr. Dobb's J.*, vol. 27, no. 8, 2002.

[7] R. Pohjonen, "Boosting embedded systems development with domain-specific modeling," *RTC Mag.*, pp. 57–61, 2003.

[8] J. Kärnä, J.-P. Tolvanen, and S. Kelly, "Evaluating the use of domain-specific modeling in practice," in *Proc. 9th OOPSLA Workshop Domain-Specific Modeling*, 2009.

[9] P. Mohagheghi and V. Dehlen, "Where is the proof?—A review of experiences from applying MDE in industry," in *Model Driven Architecture—Foundations and Applications*, Berlin, Germany: Springer, Jun. 2008, pp. 432–443.

[10] O.-P. Puolitaival and T. Kanstrén, "Towards flexible and efficient model-based testing, utilizing domain-specific modelling," in *Proc. 10th Workshop Domain-Specific Modeling*, Oct. 2010, pp. 1–6.

[11] A. Iung, J. Carbonell, L. Marchezan, E. Rodrigues, M. Bernardino, F. P. Basso, and B. Medeiros, "Systematic mapping study on domain-specific language development tools," *Empirical Softw. Eng.*, vol. 25, no. 5, pp. 4205–4249, Sep. 2020.

[12] A. W. Muzaffar, M. Tahir, M. W. Anwar, Q. Chaudry, S. R. Mir, and Y. Rasheed, "A systematic review of online exams solutions in E-learning: Techniques, tools, and global adoption," *IEEE Access*, vol. 9, pp. 32689–32712, 2021.

[13] B. Kitchenham, "Procedures for performing systematic reviews," *Keele, U.K., Keele Univ.*, vol. 33, pp. 1–26, Jul. 2004.

[14] T. Kos, T. Kosar, and M. Mernik, "Development of data acquisition systems by using a domain-specific modeling language," *Comput. Ind.*, vol. 63, no. 3, pp. 181–192, Apr. 2012.

[15] R. F. Paige, D. S. Kolovos, and F. A. C. Polack, "A tutorial on metamodelling for grammar researchers," *Sci. Comput. Program.*, vol. 96, pp. 396–416, Dec. 2014.

[16] H. Cho and J. Gray, "Design patterns for metamodels," in *Proc. Compilation Workshops DSM, TMC, AGERE!, AOOPES, NEAT, VMIL*, Oct. 2011, pp. 25–32.

[17] A. Nawaz, A. U. Rehman, T. M. Ali, F. Azam, Y. Rasheed, and M. W. Anwar, "A novel model driven approach for trajectory-based technique of autonomous vehicle route prediction," in *Proc. 16th Int. Conf. Emerg. Technol. (ICET)*, Dec. 2021, pp. 1–6.

[18] F. Corradini, A. Fedeli, F. Fornari, A. Polini, and B. Re, "DTMN a modelling notation for digital twins," in *Enterprise Design, Operations, and Computing*, T. P. Sales, H. A. Proper, G. Guizzardi, M. Montali, F. M. Maggi, and C. M. Fonseca, Eds. Cham, Switzerland: Springer, 2023, pp. 63–78.

[19] E. Syriani, D. Riegelhaupt, B. Barroca, and I. David, "Generation of custom textual model editors," *Modelling*, vol. 2, no. 4, pp. 609–625, Nov. 2021.

[20] M. Riegler, J. Sametinger, M. Vierhauser, and M. Wimmer, "A model-based mode-switching framework based on security vulnerability scores," *J. Syst. Softw.*, vol. 200, Jun. 2023, Art. no. 111633.

[21] T. Godfrey, "Participatory agent-based modelling in healthcare: A domain-specific modelling language approach," in *Proc. 25th Int. Conf. Model Driven Eng. Lang. Syst., Companion*, Oct. 2022, pp. 209–212.

[22] L. Fuentes-Fernández and A. Vallecillo-Moreno, "An introduction to UML profiles," *UML Model Eng.*, vol. 2, nos. 6–13, p. 72, 2004.

[23] D. A. Ferreira and A. Silva, "UML to OWL mapping overview an analysis of the translation process and supporting tools," in *Proc. 7th Conf. Portuguese Assoc. Inf. Syst.*, 2013.

[24] A. Amjad, S. U. Haq, M. Abbas, and M. H. Arif, "UML profile for business process modeling notation," in *Proc. Int. Bhurban Conf. Appl. Sci. Technol. (IBCAST)*, Jan. 2021, pp. 389–394.

[25] D. Bork, A. Fend, D. Scheffknecht, G. Kappel, and M. Wimmer, "From in-person to distance learning: Teaching model-driven software engineering in remote settings," in *Proc. ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst. Companion (MODELS-C)*, Oct. 2021, pp. 702–711.

[26] M. Nahotko, "Some types of grey literature: A polish context," *Grey J.*, vol. 4, no. 3, pp. 151–157, Jan. 2008.

[27] H. Khalajzadeh, J. Grundy, and J. McIntosh, "Vision: Developing collaborative model-driven apps for personalised care plans," in *Proc. 25th Int. Conf. Model Driven Eng. Lang. Syst.*, Oct. 2022, pp. 929–933.

[28] M. J. Blas and S. Gonnet, "Computer-aided design for building multipurpose routing processes in discrete event simulation models," *Eng. Sci. Technol., Int. J.*, vol. 24, no. 1, pp. 22–34, Feb. 2021.

[29] M. Quamara, G. Pedroza, and B. Hamid, "Multi-layered model-based design approach towards system safety and security co-engineering," in *Proc. ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst. Companion (MODELS-C)*, Oct. 2021, pp. 274–283.

[30] P. Boutot, M. R. Tabassum, and S. Mustafiz, "UCM4IoT: A use case modelling environment for IoT systems," in *Proc. ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst. Companion (MODELS-C)*, Oct. 2021, pp. 767–776.

[31] S. Khalid, U. Rasheed, and M. Abbas, "A model driven framework for standardizing requirement elicitation by quantifying software quality factor," in *Proc. Int. Conf. Innov. Comput. (ICIC)*, Nov. 2021, pp. 1–6.

[32] Z. Anwaar, M. Rashid, F. Azam, M. W. Anwar, Y. Rasheed, and M. Humdani, "A model based neurorehabilitation (MBN) framework using Kinect," in *Proc. IEEE Int. Syst. Conf. (SysCon)*, Apr. 2021, pp. 1–6.

[33] M. U. Farooq, M. Rashid, F. Azam, Y. Rasheed, M. W. Anwar, and Z. Shahid, "A model-driven framework for the prevention of DoS attacks in software defined networking (SDN)," in *Proc. IEEE Int. Syst. Conf. (SysCon)*, Apr. 2021, pp. 1–7.

[34] N. Petrović, I. Al-Azzoni, and J. Blank, "Model-driven multi-objective optimization approach to 6G network planning," in *Proc. 15th Int. Conf. Adv. Technol., Syst. Services Telecommun. (TELSIKS)*, Oct. 2021, pp. 223–226.

[35] L. Salmon, P.-Y. Pillain, G. Guillou, and J.-P. Babau, "CARES, a framework for CPS simulation : Application to autonomous underwater vehicle navigation function," in *Proc. Forum Specification Design Lang. (FDL)*, Sep. 2021, pp. 1–8.

[36] S. S. B. Shirazi, M. Rashid, F. Azam, Y. Rasheed, and M. W. Anwar, "A model-driven framework for early analysis of kill chain attacks," in *Proc. Nat. Comput. Colleges Conf. (NCCC)*, Mar. 2021, pp. 1–5.

[37] A. Ali, M. Rashid, F. Azam, Y. Rasheed, and M. W. Anwar, "A model-driven framework for Android supporting cross-platform GUI development," in *Proc. Nat. Comput. Colleges Conf. (NCCC)*, Mar. 2021, pp. 1–6.

[38] B. Karaduman and M. Challenger, "Model-driven development for ESP-based IoT systems," in *Proc. IEEE/ACM 3rd Int. Workshop Softw. Eng. Res. Practices IoT (SERP4IoT)*, Jun. 2021, pp. 9–12.

[39] M. Abbas, M. Rashid, F. Azam, Y. Rasheed, M. W. Anwar, and M. Humdani, "A model-driven framework for security labs using blockchain methodology," in *Proc. IEEE Int. Syst. Conf. (SysCon)*, Apr. 2021, pp. 1–7.

[40] L. Addazi and F. Ciccozzi, "Blended graphical and textual modelling for UML profiles: A proof-of-concept implementation and experiment," *J. Syst. Softw.*, vol. 175, May 2021, Art. no. 110912.

[41] J. A. Barriga, P. J. Clemente, E. Sosa-Sánchez, and Á. E. Prieto, "SimulateIoT: Domain specific language to design, code generation and execute IoT simulation environments," *IEEE Access*, vol. 9, pp. 92531–92552, 2021.

[42] Y. Al-Dhuraibi, F. Zalila, N. Djarallah, and P. Merle, "Model-driven elasticity management with OCCI," *IEEE Trans. Cloud Comput.*, vol. 9, no. 4, pp. 1549–1562, Oct. 2021.

[43] H. Xiao, Z. Li, Y. Yang, J. Deng, and S. Wei, "An extended meta-model of problem frames for enriching environmental descriptions," in *Proc. IEEE 29th Int. Requirements Eng. Conf. Workshops (REW)*, Sep. 2021, pp. 428–434.

[44] L. B. Ammar, "An automated model-based approach for developing mobile user interfaces," *IEEE Access*, vol. 9, pp. 51573–51581, 2021.

[45] P. J. Escamilla-Ambrosio, D. A. Robles-Ramírez, T. Tryfonas, A. Rodríguez-Mota, G. Gallegos-García, and M. Salinas-Rosales, "IoTsecM: A UML/SysML extension for Internet of Things security modeling," *IEEE Access*, vol. 9, pp. 154112–154135, 2021.

[46] Q. Rouland, B. Hamid, and J. Jaskolka, "Specification, detection, and treatment of STRIDE threats for software components: Modeling, formal methods, and tool support," *J. Syst. Archit.*, vol. 117, Aug. 2021, Art. no. 102073.

[47] M. Derakhshandi, S. Kolahdouz-Rahimi, J. Troya, and K. Lano, "A model-driven framework for developing android-based classic multiplayer 2D board games," *Automated Softw. Eng.*, vol. 28, no. 2, p. 7, Nov. 2021.

[48] J. Boubeta-Puig, J. Rosa-Bilbao, and J. Mendling, "CEPchain: A graphical model-driven solution for integrating complex event processing and blockchain," *Expert Syst. Appl.*, vol. 184, Dec. 2021, Art. no. 115578.

[49] P. Harnoš and L. Dedera, "DSL to support the process of creating security documentation for classified systems," in *Proc. Commun. Inf. Technol. (KIT)*, Oct. 2021, pp. 1–6.

[50] B. Van Acker, J. Denil, A. De Cock, H. Vangheluwe, and M. Challenger, "Knowledge base development and application processes applied on product-assembly co-design," in *Proc. ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst. Companion (MODELS-C)*, Oct. 2021, pp. 327–335.

[51] A. Bragança, I. Azevedo, N. Bettencourt, C. Morais, D. Teixeira, and D. Caetano, "Towards supporting SPL engineering in low-code platforms using a DSL approach," in *Proc. 20th ACM SIGPLAN Int. Conf. Generative Program., Concepts Exper.*, Oct. 2021, pp. 16–28.

[52] X. Guzmán-Guzmán, E. R. Núñez-Valdez, R. Vásquez-Reynoso, A. Asencio, and V. García-Díaz, "SWQL: A new domain-specific language for mining the social web," *Sci. Comput. Program.*, vol. 207, Jul. 2021, Art. no. 102642.

[53] M. N. Zafar, W. Afzal, and E. Enoiu, "Towards a workflow for model-based testing of embedded systems," in *Proc. 12th Int. Workshop Automating TEST Case Design, Selection, Eval.*, Aug. 2021, pp. 33–40.

[54] S. Meacham, V. Pech, and D. Nauck, "AdaptiveSystems: An integrated framework for adaptive systems design and development using MPS JetBrains domain-specific modeling environment," *IEEE Access*, vol. 9, pp. 127973–127984, 2021.

[55] A. S. Anand, K. Sayani, and R. K. Shyamasundar, "Fortress abstractions in X10 framework," *Int. J. Parallel Program.*, vol. 49, no. 6, pp. 911–933, Dec. 2021.

[56] M. Gharaat, M. Sharbaf, B. Zamani, and A. Hamou-Lhadj, "ALBA: A model-driven framework for the automatic generation of Android location-based apps," *Automated Softw. Eng.*, vol. 28, no. 1, pp. 1–45, May 2021.

[57] N. Gull, M. Rashid, F. Azam, Y. Rasheed, and M. W. Anwar, "A block-chain oriented model driven framework for handling inconsistent requirements in global software development," in *Proc. 10th Int. Conf. Softw. Comput. Appl.*, Feb. 2021, pp. 105–111.

[58] A. Indamutsa, J. Di Rocco, D. Di Ruscio, and A. Pierantonio, "MDE-ForgeWL: Towards cloud-based discovery and composition of model management services," in *Proc. ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst. Companion (MODELS-C)*, Oct. 2021, pp. 118–127.

[59] S. I. Ahmad, T. Rana, and A. Maqbool, "A model-driven framework for the development of MVC-based (web) application," *Arabian J. Sci. Eng.*, vol. 47, no. 2, pp. 1733–1747, Feb. 2022.

[60] M. Gharib, P. Lollini, and A. Bondavalli, "IQCPSoS: A model-based approach for modeling and analyzing information quality requirements for cyber-physical system-of-systems," *J. Data Semantics*, vol. 10, nos. 3–4, pp. 267–289, Dec. 2021.

[61] A. Idani, "A lightweight development of outbreak prevention strategies built on formal methods and xDSLs," in *Proc. 2nd Eur. Symp. Softw. Eng.*, Nov. 2021, pp. 85–93.

[62] Z. Hayat, M. Rashid, F. Azam, Y. Rasheed, and M. W. Anwar, "Extension of interaction flow modeling language for geographical information systems," in *Proc. 10th Int. Conf. Softw. Comput. Appl.*, Feb. 2021, pp. 186–192.

[63] M. Ladeira, Y. Ouhammou, and E. Grolleau, "RoBMEX: ROS-based modelling framework for end-users and experts," *J. Syst. Archit.*, vol. 117, Aug. 2021, Art. no. 102089.

[64] C. Wolschke, S. Marksteiner, T. Braun, and M. Wolf, "An agnostic domain specific language for implementing attacks in an automotive use case," in *Proc. 16th Int. Conf. Availability, Rel. Secur.*, Aug. 2021, pp. 1–9.

[65] K. Fall and E. Syriani, "Evolving domain-specific languages depending on external libraries," in *Proc. 36th Annu. ACM Symp. Appl. Comput.*, Mar. 2021, pp. 1247–1255.

[66] A. Gómez, M. Iglesias-Urkia, L. Belategi, X. Mendialdua, and J. Cabot, "Model-driven development of asynchronous message-driven architectures with AsyncAPI," *Softw. Syst. Model.*, vol. 21, no. 4, pp. 1583–1611, Aug. 2022.

[67] A. Fatemian, B. Zamani, M. Masoumi, M. Kamranpour, B. T. Ladani, and S. K. Rahimi, "Automatic generation of XACML code using model-driven approach," in *Proc. 11th Int. Conf. Comput. Eng. Knowl. (ICCKE)*, Oct. 2021, pp. 206–211.

[68] I. Yuskevich, A. M. Hein, K. Amokrane-Ferka, A. Doufene, and M. Jankovic, "A metamodel of an informational structure for model-based technology roadmapping," *Technol. Forecasting Social Change*, vol. 173, Dec. 2021, Art. no. 121103.

[69] D. Du, J. Chen, M. Zhang, and M. Ma, "Towards verified safety-critical autonomous driving scenario with ADSML," in *Proc. IEEE 45th Annu. Comput., Softw., Appl. Conf. (COMPSAC)*, Jul. 2021, pp. 1333–1338.

[70] B. Chen and T. Li, "Formal modeling and verification of autonomous driving scenario," in *Proc. IEEE Int. Conf. Inf. Commun. Softw. Eng. (ICICSE)*, Mar. 2021, pp. 313–321.

[71] A. Bastrykina, V. Zyubin, and A. Rozov, "Developing reflex IDE kernel with Xtext framework," in *Proc. IEEE 22nd Int. Conf. Young Professionals Electron Devices Mater. (EDM)*, Jun. 2021, pp. 511–514.

[72] M. Trezzy, I. Ober, I. Ober, and R. Oliveira, "Leveraging domain specific modeling to increase accessibility of robot programming," in *Proc. IEEE Int. Workshop Electron., Control, Meas., Signals Appl. Mechatronics (ECMSM)*, Jun. 2021, pp. 1–9.

[73] J. Cooper, A. De la Vega, R. Paige, D. Kolovos, M. Bennett, C. Brown, B. S. Piña, and H. H. Rodriguez, "Model-based development of engine control systems: Experiences and lessons learnt," in *Proc. ACM/IEEE 24th Int. Conf. Model Driven Eng. Lang. Syst. (MODELS)*, Oct. 2021, pp. 308–319.

[74] R. Veloso, J. Araujo, and A. Rodrigues, "A WebGIS interface requirements modeling language," in *Proc. Int. Conf. Res. Challenges Inf. Sci.* Cham, Switzerland: Springer, 2021, pp. 265–273.

[75] M. N. Tabbiche, M. F. Khalfi, and R. Adjoudj, "A smart modeling tool for model-driven engineering of ubiquitous applications: COVID-19 contact-tracer," in *Proc. Int. Conf. Recent Adv. Math. Informat. (ICRAMI)*, Sep. 2021, pp. 1–6.

[76] M. Fuksa and S. Becker, "Mini programming worlds: Teaching MDSD via the hamster simulator," in *Proc. ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst. Companion (MODELS-C)*, Oct. 2021, pp. 696–701.

[77] I. Al-Azzoni, N. Petrovic, and A. Alqahtani, "A utility to transform CSV data into EMF," in *Proc. 8th Int. Conf. Softw. Defined Syst. (SDS)*, Dec. 2021, pp. 1–6.

[78] A. Ouared, M. Amrani, and P.-Y. Schobbens, "COMORP: Rapid prototyping for mathematical database cost models development," *J. Comput. Lang.*, vol. 73, Dec. 2022, Art. no. 101173.

[79] A. Hussain, W. Wu, and Z. Tang, "An MDE-based methodology for closed-world integrity constraint checking in the semantic web," *J. Web Semantics*, vol. 74, Oct. 2022, Art. no. 100717.

[80] M. W. Anwar and F. Ciccozzi, "Blended metamodeling for seamless development of domain-specific modeling languages across multiple workbenches," in *Proc. IEEE Int. Syst. Conf. (SysCon)*, Apr. 2022, pp. 1–7.
[81] S. Idowu, D. Strüber, and T. Berger, "EMMM: A unified meta-model for tracking machine learning experiments," in *Proc. 48th Euromicro Conf. Softw. Eng. Adv. Appl. (SEAA)*, Aug. 2022, pp. 48–55.
[82] S. S. Alotaibi, M. Rashid, F. Azam, W. Abbas, Y. Rasheed, M. H. Sinky, and M. W. Anwar, "Integration of smart traffic fog nodes and homes through a model driven approach," in *Proc. 2nd Int. Conf. Comput. Inf. Technol. (ICCIT)*, Jan. 2022, pp. 104–109.
[83] L. Rana, A. U. Rehman, S. Javaid, and T. M. Ali, "A novel model-driven approach for visual impaired people assistance OPTIC ALLY," in *Proc. 3rd Int. Conf. Latest trends Electr. Eng. Comput. Technol. (INTELLECT)*, Nov. 2022, pp. 1–8.
[84] M. Naz, M. Rashid, F. Azam, M. Abbas, Y. Rasheed, M. H. Sinky, and M. W. Anwar, "A model driven framework for efficient bandwidth allocation in fog computing using priority algorithm," in *Proc. 2nd Int. Conf. Comput. Inf. Technol. (ICCIT)*, Jan. 2022, pp. 39–44.
[85] N. Petrovic, I. Al-Azzoni, D. Krstic, and A. Alqahtani, "Base station anomaly prediction leveraging model-driven framework for classification in Neo4j," in *Proc. Int. Conf. Broadband Commun. Next Gener. Netw. Multimedia Appl. (CoBCom)*, Jul. 2022, pp. 1–5.
[86] A. Amjad, F. Azam, and M. W. Anwar, "Device interoperability for industrial IoT using model-driven architecture," in *Proc. 24th Int. Multitopic Conf. (INMIC)*, Oct. 2022, pp. 1–6.
[87] M. W. Anwar, M. Latifaj, and F. Ciccozzi, "Blended modeling applied to the portable test and stimulus standard," in *Proc. 19th Int. Conf. Inf. Technol.-New Generat. (ITNG)*. Cham, Switzerland: Springer, 2022, pp. 39–46.
[88] Z. Lakhdara and N. Marir, "A formal modelling language for digital electronic systems design," in *Proc. Int. Conf. Adv. Aspects Softw. Eng. (ICAASE)*, Sep. 2022, pp. 1–6.
[89] S. Adib and B. Zamani, "A model-driven approach for automatic generation of Android tourism applications," in *Proc. 13th Int. Conf. Inf. Knowl. Technol. (IKT)*, Dec. 2022, pp. 1–6.
[90] M. W. Anwar, M. T. B. Shuaib, F. Azam, and A. Safdar, "A model-driven framework for design and analysis of vehicle suspension systems," in *Proc. Int. Conf. Inf. Softw. Technol.* Cham, Switzerland: Springer, 2022, pp. 197–208.
[91] L. Erazo-Garzón, P. Cedillo, G. Rossi, and J. Moyano, "A domain-specific language for modeling IoT system architectures that support monitoring," *IEEE Access*, vol. 10, pp. 61639–61665, 2022.
[92] A. Bucaioni, A. Di Salle, L. Iovino, I. Malavolta, and P. Pelliccione, "Reference architectures modelling and compliance checking," *Softw. Syst. Model.*, vol. 22, no. 3, pp. 891–917, Jun. 2023.
[93] A. Bucaioni, F. Di Silvestro, I. Singh, M. Saadatmand, and H. Muccini, "Model-based generation of test scripts across product variants: An experience report from the railway industry," *J. Softw., Evol. Process*, vol. 34, no. 11, Nov. 2022, Art. no. e2498.
[94] Y. Liu and J.-M. Bruel, "Modeling of natural language requirements based on states and modes," in *Proc. IEEE 30th Int. Requirements Eng. Conf. Workshops (REW)*, Aug. 2022, pp. 190–194.
[95] M. M. Awan, W. H. Butt, M. W. Anwar, and F. Azam, "Seamless runtime transformations from natural language to formal methods—A usecase of Z-notation," in *Proc. 17th Annu. Syst. Syst. Eng. Conf. (SOSE)*, Jun. 2022, pp. 375–380.
[96] M. Latifaj, F. Ciccozzi, M. W. Anwar, and M. Mohlin, "Blended graphical and textual modelling of UML-RT state-machines: An industrial experience," in *Software Architecture*, P. Scandurra, M. Galster, R. Mirandola, and D. Weyns, Eds. Cham, Switzerland: Springer, 2022, pp. 22–44.
[97] K. Chaoui, S. Bouzidi-Hassini, and Y. Bellik, "SUIL: A modeling language for spatial user interaction," *J. Reliable Intell. Environ.*, vol. 9, no. 2, pp. 161–181, Jun. 2023.
[98] A. Moin, M. Challenger, A. Badii, and S. Günnemann, "A model-driven approach to machine learning and software modeling for the IoT: Generating full source code for smart Internet of Things (IoT) services and cyber-physical systems (CPS)," *Softw. Syst. Model.*, vol. 21, no. 3, pp. 987–1014, Jun. 2022.
[99] A. Cimatti, S. Corfini, L. Cristoforetti, M. Di Natale, A. Griggio, S. Puri, and S. Tonetta, "A comprehensive framework for the analysis of automotive systems," in *Proc. 25th Int. Conf. Model Driven Eng. Lang. Syst.*, Oct. 2022, pp. 379–389.
[100] M. Latifaj, H. Taha, F. Ciccozzi, and A. Cicchetti, "Cross-platform blended modelling with jetbrains mps and eclipse modeling framework," in *Proc. 19th Int. Conf. Inf. Technol.-New Generat. (ITNG)*, S. Latifi, Ed. Cham, Switzerland: Springer, 2022, pp. 3–10.
[101] F. Fourati, M. T. Bhiri, and R. Robbana, "Validating event-B models using PDDL," *Proc. Comput. Sci.*, vol. 207, pp. 2638–2647, Jan. 2022.
[102] D. L. Van Bossuyt, N. Papakonstantinou, B. Hale, J. Salonen, and B. O'Halloran, "Model based resilience engineering for design and assessment of mission critical systems containing artificial intelligence components," in *Artificial Intelligence and Cybersecurity: Theory and Applications*. Cham, Switzerland: Springer, 2023, pp. 47–66.
[103] M. Mehrabi, B. Zamani, and A. Hamou-Lhadj, "HealMA: A model-driven framework for automatic generation of IoT-based Android health monitoring applications," *Automated Softw. Eng.*, vol. 29, no. 2, p. 56, Nov. 2022.
[104] T. Ye, Y. Zhuang, and G. Qiao, "MBIPV: A model-based approach for identifying privacy violations from software requirements," *Softw. Syst. Model.*, vol. 22, no. 4, pp. 1251–1280, Aug. 2023.
[105] L. Alves, J. D. Pereira, N. Aragão, M. Chagas, and P. H. Maia, "DRESS-ML: A domain-specific language for modelling exceptional scenarios and self-adaptive behaviours for drone-based applications," in *Proc. IEEE/ACM 44th Int. Conf. Softw. Eng., Softw. Eng. Soc. (ICSE-SEIS)*, May 2022, pp. 56–66.
[106] A. Safdar, F. Azam, M. W. Anwar, U. Akram, and Y. Rasheed, "MoDLF: A model-driven deep learning framework for autonomous vehicle perception (AVP)," in *Proc. 25th Int. Conf. Model Driven Eng. Lang. Syst.*, Oct. 2022, pp. 187–198.
[107] T. R. Silva, "Towards a domain-specific language to specify interaction scenarios for web-based graphical user interfaces," in *Proc. Companion ACM SIGCHI Symp. Eng. Interact. Comput. Syst.*, Jun. 2022, pp. 48–53.
[108] K. Boyalakuntla, M. Chinnakali, S. Chimalakonda, and K. Chandrasekaran, "EGEN: An energy-saving modeling language and code generator for location-sensing of mobile apps," in *Proc. 30th ACM Joint Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.*, Nov. 2022, pp. 1697–1700.
[109] C. Di Sipio, "Automating the design of recommender systems: From foundational aspects to actual development," in *Proc. 25th Int. Conf. Model Driven Eng. Lang. Syst.*, Oct. 2022, pp. 233–236.
[110] J. Li, S. Nejati, M. Sabetzadeh, and M. McCallen, "A domain-specific language for simulation-based testing of IoT edge-to-cloud solutions," in *Proc. 25th Int. Conf. Model Driven Eng. Lang. Syst.*, Oct. 2022, pp. 367–378.
[111] L. Almonte, E. Guerra, I. Cantador, and J. De Lara, "Building recommenders for modelling languages with droid," in *Proc. 37th IEEE/ACM Int. Conf. Automated Softw. Eng.*, Oct. 2022, pp. 1–4.
[112] A. Delgado, D. Calegari, F. García, and B. Weber, "Model-driven management of BPMN-based business process families," *Softw. Syst. Model.*, vol. 21, no. 6, pp. 2517–2553, Dec. 2022.
[113] R. Pérez-Castillo and M. Piattini, "Design of classical-quantum systems with UML," *Computing*, vol. 104, no. 11, pp. 2375–2403, Nov. 2022.
[114] M. Fuksa, "ViMoTest: A low code approach to specify ViewModel-based tests with a projectional DSL using JetBrains MPS," in *Proc. 25th Int. Conf. Model Driven Eng. Lang. Syst.*, Oct. 2022, pp. 189–194.
[115] J. Neugebauer and H. Kuchen, "Model-driven customizing of ERP systems: A case study," in *Proc. 37th ACM/SIGAPP Symp. Appl. Comput.*, Apr. 2022, pp. 1276–1279.
[116] J. Harbin, S. Gerasimou, N. Matragkas, T. Zolotas, R. Calinescu, and M. A. Santana, "Model-driven design space exploration for multi-robot systems in simulation," *Softw. Syst. Model.*, vol. 22, no. 5, pp. 1665–1688, Oct. 2023.
[117] I. Alfonso, K. Garcés, H. Castro, and J. Cabot, "A model-based infrastructure for the specification and runtime execution of self-adaptive IoT architectures," *Computing*, vol. 105, no. 9, pp. 1883–1906, Sep. 2023.
[118] F. Corradini, A. Fedeli, F. Fornari, A. Polini, B. Re, and L. Ruschioni, "X-IoT: A model-driven approach to support IoT application portability across IoT platforms," *Computing*, vol. 105, no. 9, pp. 1981–2005, Sep. 2023.
[119] D. Alulema, J. Criado, L. Iribarne, A. J. Fernández-García, and R. Ayala, "SI4IoT: A methodology based on models and services for the integration of IoT systems," *Future Gener. Comput. Syst.*, vol. 143, pp. 132–151, Jun. 2023.

[120] S. Zschaler and F. A. C. Polack, "Trustworthy agent-based simulation: The case for domain-specific modelling languages," *Softw. Syst. Model.*, vol. 22, no. 2, pp. 455–470, Apr. 2023.
[121] J.-M. Horcas, M. Pinto, and L. Fuentes, "A modular metamodel and refactoring rules to achieve software product line interoperability," *J. Syst. Softw.*, vol. 197, 2023, Art. no. 111579.
[122] E. Negm, S. Makady, and A. Salah, "Survey on domain specific languages implementation aspects," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 11, pp. 624–633, 2019.
[123] R. Wei, A. Zolotas, H. Hoyos Rodriguez, S. Gerasimou, D. S. Kolovos, and R. F. Paige, "Automatic generation of UML profile graphical editors for papyrus," *Softw. Syst. Model.*, vol. 19, no. 5, pp. 1083–1106, Sep. 2020.
[124] D. Allaki, M. Dahchour, and A. En-nouaary, "Building consistent uml models for better model driven engineering," *J. Digit. Inf. Manag.*, vol. 15, no. 5, pp. 289–300, 2017.
[125] M. Richters and M. Gogolla, "Validating UML models and OCL constraints," in *Proc. Int. Conf. Unified Modeling Lang.* Berlin, Germany: Springer, 2000, pp. 265–277.
[126] M. C. W. Geilen, "Formal techniques for verification of complex real-time systems," Ph.D. thesis, CompSOC Lab Predictable Composable Embedded Syst., Dept. Elect. Eng., Eindhoven Univ. Technol., 2002.
[127] A. Gargantini, E. Riccobene, and P. Scandurra, "Combining formal methods and MDE techniques for model-driven system design and analysis," *Int. J. Adv. Softw.*, vol. 3, no. 1, pp. 1–18, 2010.
[128] N. Medvidovic and R. N. Taylor, "A classification and comparison framework for software architecture description languages," *IEEE Trans. Softw. Eng.*, vol. 26, no. 1, pp. 70–93, Jan. 2000.
[129] T. S. Ivanovich, "Languages of architectural description in systems and software engineering," in *Proc. CEUR Workshop*, 2019, pp. 179–184.
[130] D. Cetinkaya and A. Verbraeck, "Metamodeling and model transformations in modeling and simulation," in *Proc. Winter Simul. Conf. (WSC)*, Dec. 2011, pp. 3043–3053.

AMINA ZAFAR received the bachelor's degree in software engineering from the Department of Software Engineering, University of Engineering and Technology (UET), Taxila, Pakistan, in 2021. She is currently pursuing the master's degree in software engineering with the Department of Computer and Software Engineering, College of Electrical and Mechanical Engineering, National University of Sciences and Technology (NUST). Her major research interests include model-driven software engineering, machine learning, deep learning, and natural language processing (NLP).

FAROOQUE AZAM is currently a Key Faculty Member with the Department of Computer and Software Engineering, College of Electrical and Mechanical Engineering, National University of Sciences and Technology (NUST), Pakistan. He has been involved in postgraduate teaching and research, since 2007. He had 138 international journals and conference publications, until April 2020. His research interests include model-driven software engineering, model-driven testing, model-driven embedded applications, model-driven web engineering, software design, and architecture. He is a regular member of Pakistan Engineering Council (PEC) evaluation committees and the Higher Education Commission's Technological Development Funding (HEC-TDF).

AFSHAN LATIF received the bachelor's degree in software engineering from the Department of Software Engineering, Mirpur University of Science and Technology (MUST), Mirpur, Pakistan, in 2019, and the master's degree in software engineering from the Department of Computer and Software Engineering, College of Electrical and Mechanical Engineering, National University of Science and Technology (NUST), Islamabad, Pakistan, in 2022. She is currently a Research Engineer with NUST. Her major research interests include model-driven software engineering, machine learning, deep learning, and natural language processing (NLP).

MUHAMMAD WASEEM ANWAR received the Ph.D. degree from the Department of Computer and Software Engineering, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Pakistan. He is currently a Postdoctoral Scholar with Mälardalen University, Västerås, Sweden. He is also a Senior Researcher and an Industry Practitioner in the field of model-based system engineering (MBSE) for embedded and control systems. His primary research interests include model-based system engineering (MBSE) for complex and large systems.

AON SAFDAR received the master's degree in computer software engineering from the College of Electrical and Mechanical Engineering (CEME), National University of Science and Technology (NUST), Islamabad, Pakistan. He is currently associated with the Model-Driven Software Engineering (MDSE) and Artificial Intelligence (AI) research groups, CEME, NUST. His research interests include model-driven software engineering, model-driven architecture, machine learning, deep learning, and computer vision.