ISBN: 978-628-95207-8-1. ISSN: 2414-6390. Digital Object Identifier: 10.18687/LACCEI2024.1.1.1336

Tools in Model-Driven Web Development: A Systematic Mapping Study

José Castillo[1], and Abraham Dávila[2]
[1]Universidad Nacional Agraria de la Selva, Perú, joseo.castillo@unas.edu.pe
[2]Pontificia Universidad Católica del Perú, Perú, abraham.davila@pucp.edu.pe

Abstract*– Model-driven development (MDD) is an application development paradigm that prioritizes the construction of models instead of code, offering several advantages. In this context, tools play an important role in supporting MDD more efficiently. Several tools have been proposed for model-driven web development (MDWD), and it is necessary to identify the characteristics of those tools. This study is aimed at identifying and classifying published articles related to MDWD tools. A systematic mapping study was thus performed using eight relevant digital databases. When searches were performed on specified strings, 1,654 articles were obtained. After the selection process, 79 articles were considered primary studies for addressing the nine research questions posed. The publication of articles on MDWD tools is consistent, indicating that research will continue in this field. Unified modeling language in the context of MDWD tools is the most reported notation type in the literature. Finally, code generation and productivity increase were reported to be the main benefits of MDWD tools.

***Keywords--* Model driven development, MDD, MDWD, systematic mapping study.**

I. INTRODUCTION

Model-driven web development (MDWD), which is a specialization in the web application development domain, is based on the model-driven development (MDD) paradigm [1]. MDD facilitates the development of applications by prioritizing the construction of models instead of code [2]. According to a few researchers, MDD is a solution for managing complex and evolving software systems as it promotes component reuse and the fast generation of quality code [3]. The main fields in which MDD has been applied successfully include mobile, simulation and verification, security, web engineering, and graphical user interface development [4].

The objective of MDWD is to provide tools and techniques for simplifying the design, development, and evolution of web applications [5]; thus, MDWD has become a relevant topic for a significant group of web application developers. Interest in web application development has increased significantly with the increase in the popularity of mobile devices such as smartphones [6], [7]. The main characteristic of the MDWD process is that it constitutes a transformation [8] to produce different models, views, or artifacts using another model based on a transformation pattern [8]. The modeling and manipulation of models allow for the abstraction level to become higher than the code level; this yields several significant advantages [9].

Digital Object Identifier: (only for full papers, inserted by LACCEI).
ISSN, ISBN: (to be inserted by LACCEI).
DO NOT REMOVE

Some authors have highlighted that the MDWD is important for software engineering because it facilitates the development of web software systems that are extensible, reusable, and easy to maintain [10], thus facilitating a good understanding of web application designs [11]. Other researchers [2], [12] have indicated that a significant amount of work is required to ensure that modern web applications are developed faster, more securely, and with fewer errors. It has been highlighted in [13] that low-code tools and technology constitute the evolution of MDD, in recent years, and there have been many secondary studies on MDD [2], [14], [15], [16], [17], [18], [19].

There are several publications regarding MDWD tools that are required to be identified and classified. These classified articles present the relevant characteristics of tools that are useful for MDWD practitioners and researchers. In this study, a systematic mapping study (SMS) was conducted on MDWD tools. This paper is organized as follows. Section 2 introduces the fundamental concepts applied in this study. In Section 3, the related work is summarized. Section 4 details the SMS process, and Section 5 consolidates the results. In Section 6, the results are discussed in relation to the research questions, and conclusions are drawn.

II. BACKGROUND

This section introduces the concepts of MDD and MDWD.

A. Model-driven development

MDD is a paradigm that addresses the challenges of integration and composition of a system [20]. It allows for the creation and use of business models rather than computational concepts [21], [22], thus enhancing the abstraction level of software development and bridging the gap between technology domains by enabling domain experts to design and build systems [20]. MDD models encompass the system requirements, business models, and technology implementations [23] and can represent systems at any level of abstraction ranging from business architectures to technology implementations [24], thereby contributing to the simplification of these architectures [25].

Relevant terms in the literature include model-driven architecture (MDA) [26], model-driven engineering (MDE) [2], [8], [25], and model-driven software engineering [27]. In addition, MDA combines domain-specific modeling languages, transformation engines, and generators [28].

In terms of thematic domain areas, this paradigm establishes the context and scope of the model [24]. High-level domains include telecommunications, healthcare, sales, manufacturing, transportation, defense, and government.

It is important to know the models used in MDD. According to [29], there are three models: i) the business or domain model wherein the computation independent model (CIM) is located; ii) logical system models in which the components of a system interact with each other and where the platform independent model (PIM) is located; iii) implementation models, which are closer to technology and where the platform-specific model (PSM) is located. The transformation between these models, also known as model-to-model (M2M) transformation, is the core process of MDD [29]. However, the classification of [24] involves three architectural layers instead of models.

B. *Model-driven web development*

MDWD comprises other approaches such as model-driven web engineering (MDWE), which is an MDD application in the context of web software development [30]. The concept of MDWE corresponds to MDE [1]. The continuous evolution of web technologies and platforms based on the application of independent models makes MDWD useful in circumventing various web application problems [30] related to content, navigation, process, and presentation [10].

III. RELATED WORK

The related studies have been identified as follows.

According to [8], approximately half of the studies on MDD are published in conferences and workshops, and the other half are published in prestigious journals. Furthermore, this number of studies increased significantly in 2011 and 2012.

In [31], a systematic literature review (SLR) was performed, in which 30 studies (2013–2018) were identified, wherein the MDD was used for mobile application development. This resulted in the identification of 11 model-based techniques, 21 tools, and eight modeling languages.

In [27], an SMS was conducted on rich internet applications (RIAs). It was found in this study that the few tools available for MDD are difficult to access, and their robustness level is yet to be experimentally validated.

In [32], it was determined that it is extremely challenging for MDE environments that result in improvements in software productivity and quality to appear in short and medium terms because current MDE technologies reflect deficiencies in the understanding of software modeling.

In [33], it was found that agile development and MDD can coexist and benefit from integration, thus improving productivity by up to five times, code generation by 93%, and code quality by up to 20% with a shorter turnaround time of deliverables.

In [2], a literature review was conducted, wherein it was found that (i) the architecture, domain modeling, and code generation are the most important aspects of MDD-based mobile application development; (ii) the two main benefits of applying MDD are increased productivity and increased automation in code generation; lastly, (iii) 22.3% of the primary studies on MDD pertain to academic case studies on its applications.

IV. SYSTEMATIC MAPPING STUDY

The literature review is generally performed using SLR [34] or SMS [35]. SLR [34] facilitates the identification and assessment of the papers' relevance and quality, and the selected studies are synthesized. Systematic mapping studies allow for the structuring of a research area and are useful in the identification of possible future research directions [35]. An SMS was performed in this study using the guidelines proposed in [36] and improved in [35]. In the guidelines of [36], the process steps established are definition of the research questions, conducting a search, screening of papers, keywording using abstracts, data extraction, and mapping process.

A. *Scope and research question*

The objective of this study is to identify and classify the research conducted on MDWD tools and their applications, which is similar to the objective in [27]; however, this study is more oriented towards specific M2M transformations, and the research questions (RQs) provided (Table I) have a wider scope, including other transformations and aspects of MDWD tools.

TABLE I.
RESEARCH QUESTIONS

ID	Research Question
RQ01	Which tools or instruments are used in MDWD?
RQ02	What type of modelling notation is used for tools or instruments in MDWD?
RQ03	What are the source and target models in a model transformation comprising the use of MDWD tools?
RQ04	Which IDEs are reported for the use of MDWD tools/instruments?
RQ05	Which programming languages are indicated for the use of MDWD tools?
RQ06	What are the benefits of developing MDWD-based applications?
RQ07	What are the issues and challenges of adopting MDWD for application development?
RQ08	What are the main domains/areas wherein MDWD is frequently practiced?
RQ09	How has research on MDWD tools evolved?

B. *Search string and selection criteria*

In this SMS, an automatic search was performed on eight databases of scientific relevance according to [37], while considering the possibility of access to full-text documents. The databases considered were IEEE Xplore, Web of Science, ACM DL, Scopus, Wiley, Ebsco, Science Direct, and Taylor & Francis. EI Compendex was not considered as it could not be accessed. No additional sources were considered for this study. The search string was defined using the population and intervention (PI) strategy, which is recommended used for SMS [35]. The population was MDWD and the intervention used included Tools; therefore, the search string was P AND I = (MDD OR MDSD OR MDE OR MDA OR "Model Driven") AND (Web) AND (tool). There were no restrictions on the start and end dates.

The inclusion and exclusion criteria applied in this study were as follows:

- IC.1. Primary studies that focus on MDWD are acceptable.
- IC.2. Primary studies with tools that have been used or tested are acceptable.
- EC.1. Duplicate papers are rejected.
- EC.2. Books, secondary studies, and summaries of conference proceedings are rejected.
- EC.3. Articles that are not written in English are rejected.
- EC.4. Articles that do not deal with MDWD tools are rejected.
- EC.5. Articles comprising unavailable content are rejected.
- The quality evaluation was omitted in the SMS because the articles were obtained from journals, conferences, or books selected by digital databases using quality criteria [35].

C. Data extraction and classification

In this SMS, the following five-step process was established:

- First stage: The metadata of the articles are obtained by running the search strings on the databases, thereby consolidating the results in a single file, and eliminating duplicate articles in compliance with criterion EC.1.
- Second stage: The titles and keywords are reviewed to exclude articles unrelated to MDWD and reject secondary studies in compliance with criteria IC.1 and EC.2.
- Third stage: Abstracts are reviewed to exclude articles not written in English in compliance with criterion EC.3.
- Fourth stage: Articles that have passed the previous stages are downloaded to ensure that the article is written entirely in English. In addition, a quick review is performed to ensure that the articles pertain to MDWD tools in compliance with criteria EC.3, EC4, and IC.2.
- Fifth step: An in-depth reading is performed to ensure that the article concerns tools that facilitate the application of MDWD and that these tools have been used and evaluated in compliance with criterion IC.2.

During the planning of the SMS, a pilot study was conducted to validate the search string and ensure that the RQs could be answered. In the pilot study, 14 articles were identified from the IEEE and ACM databases to validate the search string and determine if adjustments were required to address the RQs.

The first step was started in July 2023, and the second query round was conducted in January 2024. The metadata obtained from digital databases were stored and processed using Microsoft Excel and macros for the classification process. In the initial stage, two authors worked together. In the second and third stages, the first author performed tasks, and the second author reviewed the process randomly. In the remaining stages, the two authors worked together.

In the SMS process, 1,654 articles were obtained from eight digital databases using a search string. The partial results of the process are listed in Table II. At the end of the selection process, 79 articles remained as primary studies (Appendix A). As indicated in [35], the databases comprised selections from journals and conferences with a formal schema; therefore, no quality assessment was explicitly performed.

A spreadsheet was used to establish an extract data template during the data extraction process (Table III). Each field in the structure was used to extract data from the primary study to answer every RQ. Furthermore, to answer each RQ, a table or figure was prepared based on quantitative analysis, and narrative synthesis was performed [37].

TABLE II.
RESULTS OF THE STUDY SELECTION PROCESS

Database	Studies obtained	Studies remained
ACM	104	---
IEEE	183	---
EBSCO	128	---
Science Direct	85	---
Scopus	890	---
Web Of Science	231	---
Wiley	20	---
Taylor and Francis	13	---
Total	1,654	1,654
Duplicates excluded	---	1,329
Title Review	---	354
Abstract Review	---	166
Relevant full text	---	79

TABLE III.
TEMPLATE FOR DATA EXTRACTION

Data	Detail	RQ
Study identifier	Unique identifier of the study created for SMS	General
MDWD Tools	Name and author of MDD tools developed for web applications	RQ01
Notation for modelling	Type of modelling notation used by MDD tools to develop web applications	RQ02
Source and target models in a transformation	Source and target models of transformation using MDD tools to develop web applications	RQ03
IDE used	IDEs allowing the use of MDWD tools	RQ04
Programming languages	Programming languages designated for use in MDWD tools/instruments	RQ05
Benefits obtained	Benefits obtained by developing applications based on MDWD	RQ06
Problems and challenges	Issues and challenges of adopting MDWD	RQ07
Areas where MDWD is applied	Main domains/areas where MDWD is frequently practiced	RQ08
Evolution of research on MDWD tools	Evolution of MDWD tool research	RQ09

D. Classifiers considered in the study

The topic-specific classifiers are a group of classifiers used in the subject domain. The classifiers that were considered in this study are as follows:

- Modeling languages: Unified modeling language (UML), extensible markup language (XML), systems modeling language (SysML), and web modeling language (WebML) [38], [39].
- Programming languages: C#, Java, Python, and PHP [32], [39].
- Tools used: The list was constructed from primary studies.
- Source and target models in a transformation: CIM, PIM, PSM, and transformation to code [38],[26], [33].

E. Validity threat analysis

The validity threat analysis was based on the work performed in [40]. The validation aspects considered in this study were as follows.

- Validation of study selection. To ensure the selection of the largest number of relevant studies, the following steps were performed: (i) A pilot search was conducted iteratively to validate the search string, resulting in a sample of 14 articles; (ii) Data extracted from the selected articles were verified to ensure that they answer the RQs. Inclusion and exclusion criteria were applied in the article selection process. In terms of the decision to include or exclude studies, when in doubt, the decision was postponed to the next stage to realize a more informed resolution. The first and second authors have more experience in MDD and MSL, respectively. Both the authors worked together, and in some cases, the second author aided the first author in the selection process.
- Data validation. Only relevant digital databases [37] were considered with the already defined evaluation schemes. Therefore, no quality assessment was performed in the selection process, as indicated in [35]. One digital database was not considered owing to a lack of access.
- Research validation. To ensure the reliability of the process, two authors who were familiar with the research topic participated in it. The second author, who had extensive experience in secondary studies and research, reviewed the selection process randomly. This process is replicable because all the data collected during the research as well as the search strings were publicly available. In addition, the search process was defined without a start date or an end date.

V. Results and Discussion

Based on the primary studies, the following RQs were answered and discussed in some cases.

A. RQ01. Which tools or instruments are used in MDWD?

In terms of the MDWD tools (Fig. 1), it was found that the majority of the authors (96%) propose new names for these tools. Among these tools, some appear more than once as in the

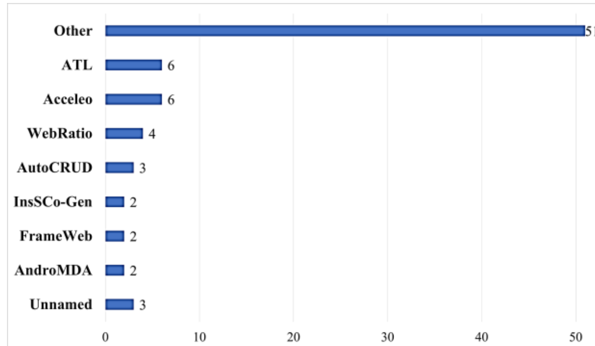

Fig. 1. Tools used in MDWD

case of the active template library (ATL) and Acceleo, which were reported in six primary studies; AutoCRUD, which was reported in three primary studies each; and InsSco-gen and FrameWeb, which were reported in two primary studies each. In the case of tools mentioned (i.e., ATL, AutoCRUD, WebRatio, InsSco-gen, and FrameWeb), some of the authors had authored more than one paper. Fig. 2 presents the names of the 58 MDWD tools reported as a text cloud.

In addition, 51 tools (64%) were grouped under the category "Other." These tools appeared only once in the primary studies, indicating a clear tendency to make new proposals instead of improving or evaluating the existing tools. This is in line with the findings of two previous studies, in which it was reported that 44% of the research efforts are focused on creating new tools [27] and that 40% of the papers propose a new tool [8]. In terms of support and documentation that facilitates the use of these tools without difficulty, Acceleo, WebRatio, ArgoUML, AutoCRUD, and ArgoUWE provide support and documentation on their respective web pages. Finally, three primary studies did not specify the name of the tool used.

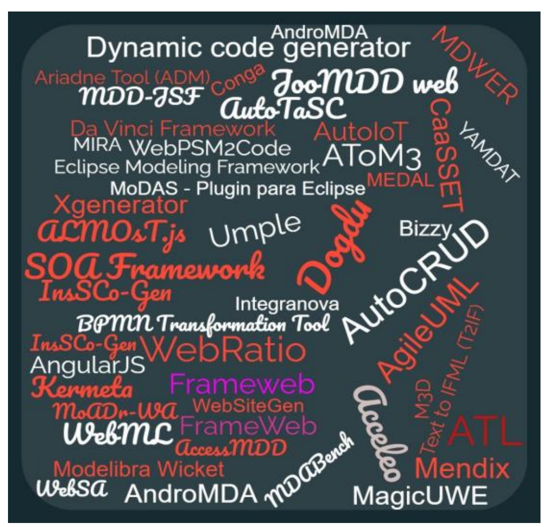

Fig. 2. Word cloud tools used in MDWD

B. RQ02. What type of modeling notation is used for tools or instruments in MDWD?

Among the primary studies, the most reported type of modeling notation was UML, which was identified in 42 (53%) studies (Fig. 3), followed by the interaction flow modeling language (IFML), which was reported in six studies. In the majority of the primary only one type of notation was reported, and in two papers reported two types. In addition, in seven primary studies, the modeling language was not specified. In agreement with our results, in [31], [33], and [8], it was found that UML was used for modeling in 50%, 44%, and 41.7% of the articles, respectively.

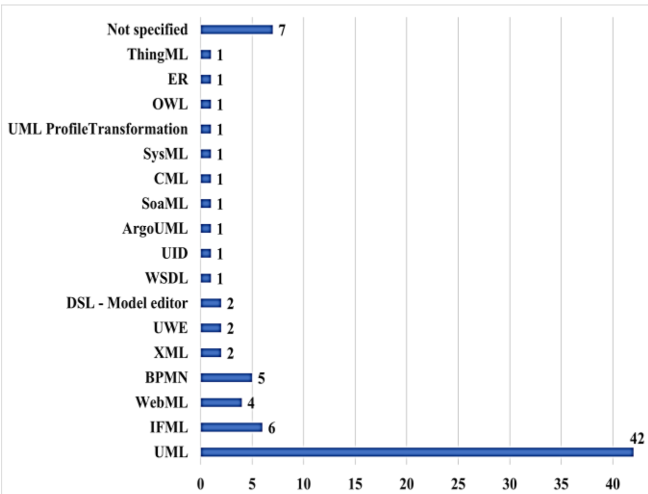

Fig. 3. Type of tool modeling notation

C. RQ03. What are the source and target models in a model transformation comprising the use of MDWD tools?

In terms of the source and target models in a model transformation (Fig. 4), it was found that 38 (48%) primary studies used PIM as their source model. This can be attributed to the fact that PIM can be implemented using UML diagrams [41], and as indicated in the response to the previous question, UML was reported in 52% of studies.

Furthermore, 37 primary studies (46%) reported tools that were intended for implementation at the source code level in a programming language. This percentage is lower than that reported in [8], which indicates that in 69% of reported articles, the target was expressed using a programming language.

In contrast, in 22 primary studies (27%), the model of origin or destination was not indicated. This result agrees with the findings of [8], which reported that 27.1% of articles did not specify the origin or destination model. However, this significantly differs from the report of [27], in which it was found that 6% of the articles did not consider any origin or destination model. These differences can be explained to some extent by the fact that the papers of [8] and [27] are oriented

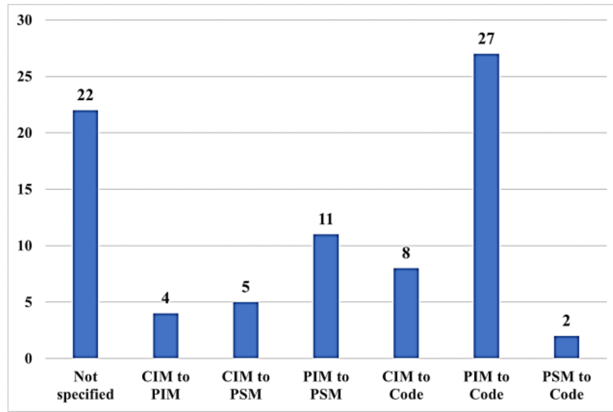

Fig. 4. Reported origin and destination models

toward MDD and RIA respectively, while our study is focused on MDWD tools.

D. RQ04. Which IDEs are reported for the use of MDWD tools/instruments?

Among the primary studies, only 32% specified the integrated development environment (IDE) used for the implementation of MDWD tools. Among these, 21% of the studies indicated Eclipse as the most reported IDE. This was followed by Visual Studio .NET and Android Studio, as indicated in 3% of the studies (Fig. 5). In contrast, in the SMS study [27], it was reported that 52% of the tools were developed using the Eclipse IDE, and 31% did not report the IDE used. This difference is explained by the fact that the work of [27] assesses transformation experiences, whereas our study is focused on tools.

E. RQ05. Which programming languages are indicated for the use of MDWD tools?

In terms of programming languages (Fig. 6a), 29% of the primary studies reported that the MDWD tools generated code

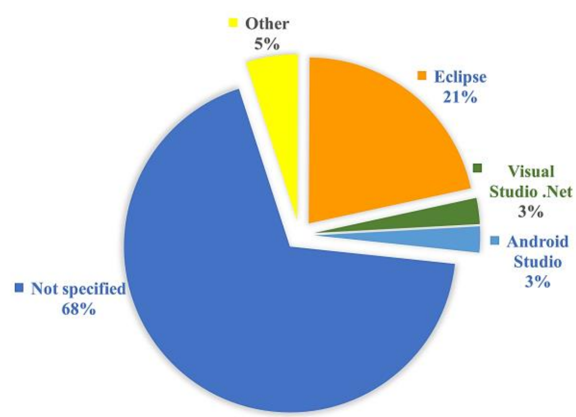

Fig. 5. IDE used for MDWD tool applications

from the Java programming language. Furthermore, in five studies published in the last 4 years (Fig. 6b), it was reported that the generated code was from Javascript, which may have been a result of the emergence of new frameworks incorporating Javascript. In related works, it was observed in [32], that MDD models produced significant portions of the code in languages such as Java, C++, and C.

Finally, 40% of the primary studies did not specify the programming language generated. This finding is related to some extent to the fact that transformation to source code is not always performed.

F. RQ06. What are the benefits of developing MDWD-based applications?

In terms of the benefits of using MDWD (Fig. 7), among the primary studies, 12 highlighted an increase in developer productivity, and 11 indicated a decrease in development time. This is in line with the findings of [33] of (a) increased productivity, (b) decreased development time, and (c) code generation facilitating reusability, validation, and compliance with requirements. In addition, 48% (38) of studies did not specify any benefits, and four primary studies reported code generation as a benefit, thus quantifying the benefit as the percentage of the code produced in the entire system, which was cited as 87% in [42] and 39% in [43].

G. RQ07. What are the issues and challenges of adopting MDWD for application development?

Among the primary studies, only 21 articles reported one or more problems or challenges in adopting MDWD, as shown in Fig. 8. Among these, 14% stated that the tools require further evaluation, and 19% indicated that the tools should be used in more complex projects. Others reported the issues that the results of using the tools cannot be generalized and that MDWD tools are difficult to use in the beginning. Similarly, in [27], it was highlighted that the tools required more evaluation; the tools were required to be applied in more complex projects, and there was a lack of documentation or manuals to facilitate tool usage.

H. RQ08. What are the main domains/areas where MDWD is frequently practiced?

In terms of the domains or areas identified in the primary studies, only 31% of the cases reported the domain. Among

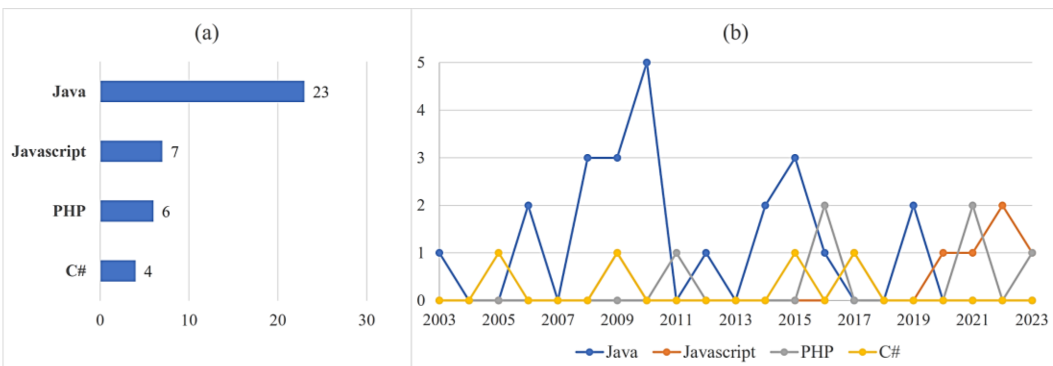

Fig. 6. Programming languages generated using MDWD tools: (a) cumulative to date and (b) annual evolution

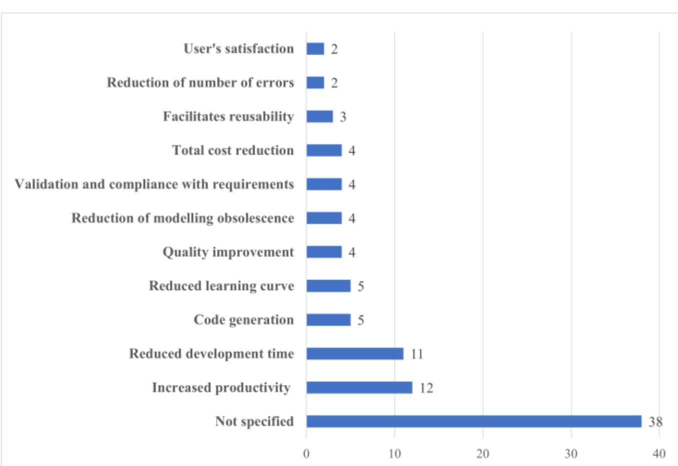

Fig. 7. Benefits realized by developing MDWD-based applications

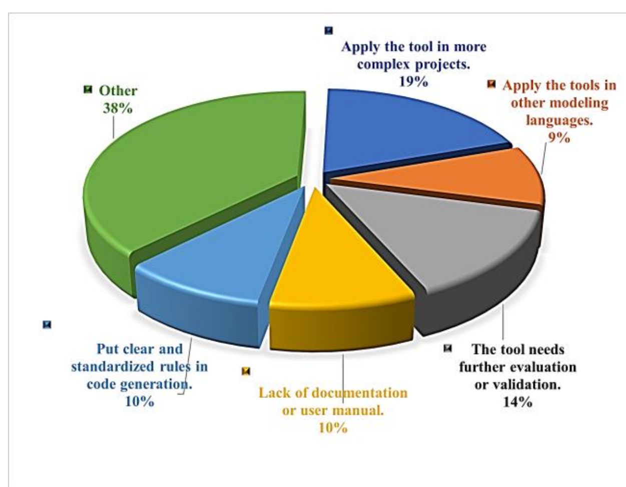

Fig. 8. Issues and challenges of adopting MDWD for application development

these (Fig. 9), 11% of the applications were developed for e-commerce and 8% for the education sector.

In the education sector, MDWD was applied for research and student training. This agrees with the report in [8], in which it was found that 64% of articles presented and developed examples for academic purposes. In the case of the business and industry sector [31], 16% of the articles reported MDWD tools as part of case studies.

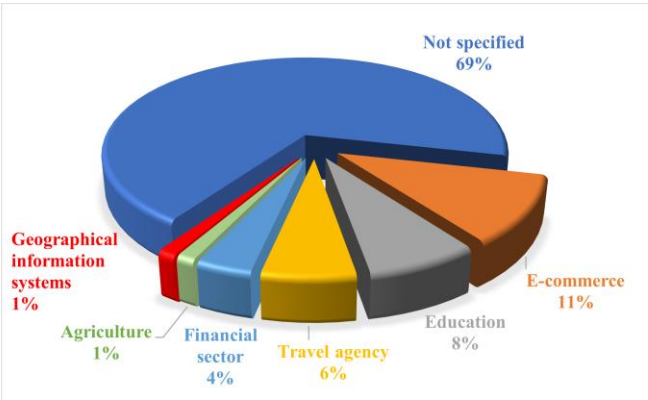

Fig. 9. Domains/areas wherein MDWD is frequently practiced

I. RQ09. How has research on MDWD tools evolved?

The evolution of articles on MDWD tools, as shown in Fig. 10, remained stable from 2003 to 2023, with an average of 3.8 articles per year. The publications in conferences and journals exhibited a similar trend. This level of publication is consistent with that reported in [31] and corresponds with the period of 2013–2018. In our study, we calculated an average of 3.5 articles per year, thus confirming the validity of MDWD as a research domain.

VI. Conclusions

In this study, an SMS was conducted on tools for developing web applications using the MDD paradigm. For this, 1,654 articles were obtained from relevant databases, and 79 primary studies were selected at the end of the process.

Based on the data obtained from the primary studies, nine research questions posed could be answered. The reported research interest in MDWD tools included the period from 2003 to date, and it was verified that this is an active research subject to date. In [2], it was noted that the interest in the MDD approach increased steadily in the period from 2005–2019.

Among the primary studies, the majority of the identified works reported tools only once (65%) with no single tool predominating in the reviewed papers. In addition, 4% of the studies did not refer to a particular tool by name because they were focused on explaining the technical component (academic purposes) or developing a tool with a specific purpose for their research. Finally, 32% of the studies referred to one of the seven MDWD tools considered.

In terms of the modeling language, transformation, IDE, and programming language used, the findings indicated that (a) UML (53%) was widely used as a notation for modeling; (b) 48% of studies relied on PIM as their source model, and coincidentally, 46% reported that the target model was the source code of some programming language; (c) Eclipse (21%) was the most reported IDE among the primary studies; and (d) Java (29%) was the most used programming language. However, other IDEs and programming languages were reported as alternatives by researchers.

The primary benefits included code generation and increased developer productivity, while the most reported challenges were conducting evaluations on MDWD tools and using these tools in more complex projects. According to the primary studies, the areas wherein MDWD tools were used mainly included e-commerce and education.

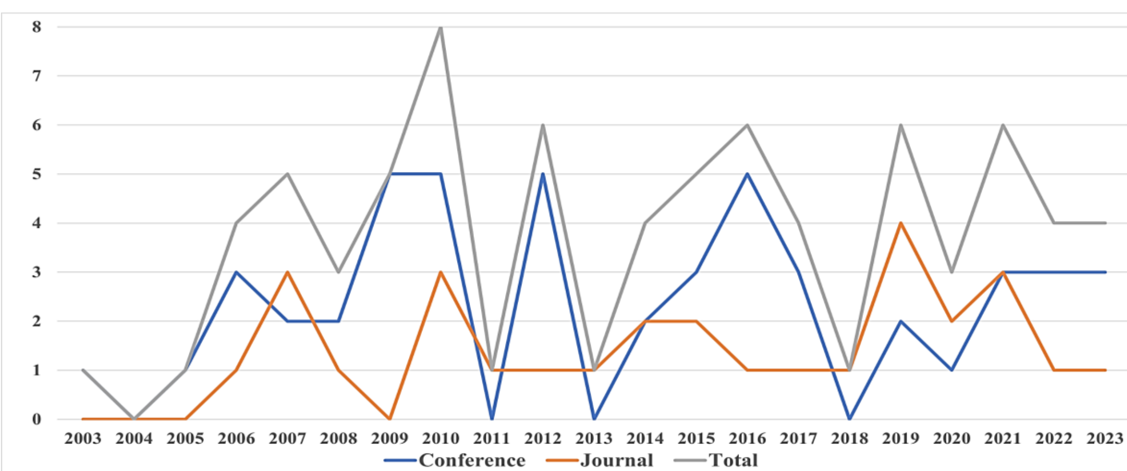

Fig. 10. Evolution of papers (publications per year)

The results of this research show that there exist opportunities for further research in terms of different aspects of MDWD tools from a technical perspective, such as their applicability in different contexts as well as their economic implications within the software industry.

ACKNOWLEDGEMENTS

This research was supported by Software Engineering Research & Development Group (GIDIS) – Pontificia Universidad Católica del Perú and Universidad Nacional Agraria de la Selva, as a part of Red Peruana de Universidades (RPU) under Project: gidis.RPU.002.PUCP.UNAS.2021.

Appendix A. List of primary studies. URL

REFERENCES

[1] Z. Morales *et al.*, *Computational Science and Its Applications – ICCSA 2016*, vol. 9790. in Lecture Notes in Computer Science, vol. 9790. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-42092-9.

[2] Md. Shamsujjoha, J. Grundy, L. Li, H. Khalajzadeh, and Q. Lu, "Developing Mobile Applications via Model Driven Development: A Systematic Literature Review," *Inf Softw Technol*, vol. 140, p. 106693, Dec. 2021, doi: 10.1016/j.infsof.2021.106693.

[3] T. Heineck, E. Gonçalves, A. Sousa, M. Oliveira, and J. Castro, "Model-driven development in robotics domain: A systematic literature review," *Proceedings of the 10th Brazilian Symposium on Components, Architectures and Reuse Software, SBCARS 2016*, pp. 151–160, 2016, doi: 10.1109/SBCARS.2016.12.

[4] G. Sebastián, J. A. Gallud, and R. Tesoriero, "Code Generation using Model Driven Architecture: A Systematic Mapping Study," *J Comput Lang*, vol. 56, no. Nov, p. 100935, Feb. 2020, doi: 10.1016/j.cola.2019.100935.

[5] G. Rossi, M. Urbieta, D. Distante, J. M. Rivero, and S. Firmenich, "25 Years of Model-Driven Web Engineering: What we achieved, What is missing," *CLEI Electronic Journal*, vol. 19, no. 3, 2016, doi: 10.19153/cleiej.19.3.1.

[6] F. Dias, L. Duarte, and R. Fortes, "AccessMDD: An MDD approach for generating accessible mobile applications," in *Proceedings of the 39th ACM International Conference on the Design of Communication: Building Coalitions. Worldwide, SIGDOC 2021*, 2021. doi: 10.1145/3472714.3473904.

[7] E. Sanchiz, M. Gonzalez, N. Aquino, and L. Cernuzzi, "Mobile cloud applications development through the model driven approach: A systematic mapping study," *Proceedings of the 2016 42nd Latin American Computing Conference, CLEI 2016*, 2017, doi: 10.1109/CLEI.2016.7833333.

[8] D. Ameller, X. Burgués, O. Collell, D. Costal, X. Franch, and M. P. Papazoglou, "Development of Service-Oriented Architectures using Model-Driven Development: A Mapping Study," *Inf Softw Technol*, vol. 62, no. 1, pp. 42–66, 2015, doi: 10.1016/j.infsof.2015.02.006.

[9] P. H. Nguyen, M. Kramer, J. Klein, and Y. Le Traon, "An extensive systematic review on the Model-Driven Development of secure systems," *Inf Softw Technol*, vol. 68, pp. 62–81, Dec. 2015, doi: 10.1016/j.infsof.2015.08.006.

[10] K. Wakil and D. N. A. Jawawi, "Model driven web engineering: A systematic mapping study," *E-Informatica Software Engineering Journal*, vol. 9, no. 1, pp. 107–142, 2015, doi: 10.5277/E-INF150106.

[11] J. A. Hincapié Londoño and J. Freddy Duitama, "Model-driven Web engineering methods: A literature review," *Revista Facultad de Ingenieria*, no. 63, pp. 69–81, 2012.

[12] Y. Martínez, C. Cachero, and S. Meliá, "MDD vs. traditional software development: A practitioner's subjective perspective," in *Information and Software Technology*, 2013. doi: 10.1016/j.infsof.2012.07.004.

[13] J. Cabot, "Positioning of the Low-Code Movement within the Field of Model-Driven Engineering," *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, MODELS-C 2020 - Companion Proceedings*, pp. 535–537, 2020, doi: 10.1145/3417990.3420210.

[14] H. Alfraihi and K. Lano, "Trends and Insights into the Use of Model-Driven Engineering: A Survey," *2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, pp. 286–295, 2023, doi: 10.1109/models-c59198.2023.00058.

[15] I. David *et al.*, "Blended modeling in commercial and open-source model-driven software engineering tools: A systematic study," *Softw Syst Model*, vol. 22, no. 1, pp. 415–447, 2023, doi: 10.1007/s10270-022-01010-3.

[16] A. A. Martinez-Garate, J. A. Aguilar-Calderon, C. Tripp-Barba, and A. Zaldivar-Colado, "Model-Driven Approaches for Conversational Agents Development: A Systematic Mapping Study," *IEEE Access*, vol. 11, no. July, pp. 73088–73103, 2023, doi: 10.1109/ACCESS.2023.3293849.

[17] K. Ordoñez, J. Hilera, and S. Cueva, "Model-driven development of accessible software: a systematic literature review," *Univers Access Inf Soc*, vol. 21, no. 1, pp. 295–324, 2022, doi: 10.1007/s10209-020-00751-6.

[18] J. Onunga, S. ; Mbugua, and S. Mbugua, "Model Driven Architecture a Review of Current Trends," *International Journal of Computer Science and Mobile Applications*, vol. 11, pp. 19–28, 2023, [Online]. Available: www.ijcsma.com

[19] C. Verbruggen and M. Snoeck, "Practitioners' experiences with model-driven engineering: a meta-review," *Softw Syst Model*, vol. 22, no. 1, pp. 111–129, Feb. 2023, doi: 10.1007/s10270-022-01020-1.

[20] K. Balasubramanian, A. Gokhale, G. Karsai, J. Sztipanovits, and S. Neema, "Developing Applications using Model-Driven Design Environments," *Computer (Long Beach Calif)*, vol. 39, no. 2, pp. 33–40, 2006, doi: 10.1109/MC.2006.54.

[21] B. Selic, "The pragmatics of model-driven development," *IEEE Softw*, vol. 20, no. 5, 2003, doi: 10.1109/MS.2003.1231146.

[22] M. Anastasopoulos, T. Forster, and D. Muthig, "Optimizing Model-driven Development by Deriving Code Generation Patterns from Product Line Architectures," *NetObject Days*, no. Jan, 2005.

[23] V. V. Graciano Neto, M. Guessi, L. B. R. Oliveira, F. Oquendo, and E. Y. Nakagawa, "Investigating the Model-Driven Development for Systems-of-Systems," in *Proceedings of the 2014 European Conference on Software Architecture Workshops*, New York, NY, USA: ACM, Aug. 2014, pp. 1–8. doi: 10.1145/2642803.2642825.

[24] OMG, "OMG MDA Guide rev. 2.0," *OMG Document ormsc*, vol. 2.0, no. June, pp. 1–15, 2014, [Online]. Available: http://www.omg.org/cgi-bin/doc?omg/03-06-01

[25] M. Hamdani, W. H. Butt, M. W. Anwar, and F. Azam, "A Systematic Literature Review on Interaction Flow Modeling Language (IFML)," in *Proceedings of the 2018 2nd International Conference on Management Engineering, Software Engineering and Service Sciences*, New York, NY, USA: ACM, Jan. 2018, pp. 134–138. doi: 10.1145/3180374.3181333.

[26] Y. Rhazali, Y. Hadi, and A. Mouloudi, "A model transformation in MDA from CIM to PIM represented by web models through SoaML and IFML," *Colloquium in Information Science and Technology, CIST*, vol. 0, pp. 116–121, 2016, doi: 10.1109/CIST.2016.7805027.

[27] D. Bonhaure, M. Gonzalez, N. Aquino, L. Cernuzzi, and C. Pons, "Model-to-Model Transformations for RIA Architectures: A Systematic Mapping Study," in *2016 XLII Latin American Computing Conference (CLEI)*, IEEE, Oct. 2016, pp. 1–11. doi: 10.1109/CLEI.2016.7833405.

[28] D. Schmidt, "Model-Driven Engineering," vol. 39, no. 2, pp. 25–31, 2006.

[29] S. Martínez, L. Lamonth, and N. Jacho, "Análisis de la Transformación de Modelo CIM a PIM en el Marco de Desarrollo de la Arquitectura Dirigida por Modelos (MDA)," *Revista Politécnica - Septiembre*, vol. 36, no. 3, 2015, [Online]. Available: https://www.revistapolitecnica.epn.edu.ec/images/revista/volumen36/tomo3/AnalisisdelaTransformaciondeModeloCIMaPIMenelMarco.pdf

[30] A. Kraus, A. Knapp, and N. Koch, "Model-driven generation of web applications in UWE," *CEUR Workshop Proc*, vol. 261, no. January, 2007.
[31] H. Tufail, F. Azam, M. W. Anwar, and I. Qasim, "Model-Driven Development of Mobile Applications: A Systematic Literature Review," *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON 2018*, pp. 1165–1171, 2019, doi: 10.1109/IEMCON.2018.8614821.
[32] R. France and B. Rumpe, "Model-driven development of complex software: A research roadmap," *FoSE 2007: Future of Software Engineering*, pp. 37–54, 2007, doi: 10.1109/FOSE.2007.14.
[33] H. Alfraihi and K. Lano, "The Integration of Agile Development and Model Driven Development: A Systematic Literature Review," *MODELSWARD 2017 - Proceedings of the 5th International Conference on Model-Driven Engineering and Software Development*, vol. 2017-Janua, no. Modelsward, pp. 451–458, 2017, doi: 10.5220/0006207004510458.
[34] B. Kitchenham and S. Charters, "Guidelines for Performing Systematic Literature Reviews in Software Engineering. Version 2.3," 2007.
[35] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Inf Softw Technol*, vol. 64, pp. 1–18, Aug. 2015, doi: 10.1016/J.INFSOF.2015.03.007.
[36] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic Mapping Studies in Software Engineering," in *12th International Conference on Evaluation and Assessment in Software Engineering (EASE)*, 2008, pp. 1–10. doi: 10.1142/S0218194007003112.
[37] A. Carrera-Rivera, W. Ochoa, F. Larrinaga, and G. Lasa, "How-to Conduct a Systematic Literature Review: A Quick Guide for Computer Science Research," *MethodsX*, vol. 9, 2022, doi: 10.1016/j.mex.2022.101895.
[38] N. Silega, M. Noguera, Y. I. Rogozov, V. S. Lapshin, and T. Gonzalez, "Transformation From CIM to PIM: a Systematic Mapping," *IEEE Access*, vol. 10, pp. 90857–90872, 2022, doi: 10.1109/ACCESS.2022.3201556.
[39] I. Zafar, F. Azam, M. W. Anwar, B. Maqbool, W. H. Butt, and A. Nazir, "A Novel Framework to Automatically Generate Executable Web Services From BPMN Models," *IEEE Access*, vol. 7, pp. 93653–93677, 2019, doi: 10.1109/ACCESS.2019.2927785.
[40] A. Ampatzoglou, S. Bibi, P. Avgeriou, M. Verbeek, and A. Chatzigeorgiou, "Identifying, categorizing and mitigating threats to validity in software engineering secondary studies," *Inf Softw Technol*, vol. 106, pp. 201–230, Feb. 2019, doi: 10.1016/j.infsof.2018.10.006.
[41] A. T. Rahmani, V. Rafe, S. Sedighian, and A. Abbaspour, "An MDA-based modeling and design of service oriented architecture," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2006. doi: 10.1007/11758532_76.
[42] W. Chansuwath and T. Senivongse, "A Model-Driven Development of Web Applications using AngularJS Framework," in *2016 IEEE/ACIS 15th International Conference on Computer and Information Science, ICIS 2016 - Proceedings*, 2016. doi: 10.1109/ICIS.2016.7550838.
[43] J. D. S. Saraiva and A. R. Da Silva, "Development of CMS-based web-applications using a model-driven approach," in *4th International Conference on Software Engineering Advances, ICSEA 2009, Includes SEDES 2009: Simposio para Estudantes de Doutoramento em Engenharia de Software*, 2009. doi: 10.1109/ICSEA.2009.79.