Information and Software Technology 125 (2020) 106316

Contents lists available at ScienceDirect

Information and Software Technology

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

Model composition in Model Driven Engineering: A systematic literature review

Anas Abouzahra[a,b,*], Ayoub Sabraoui[a], Karim Afdel[a]

[a] Laboratory of Computer Systems and Vision LabSIV, Ibn Zohr University, Agadir, Morocco
[b] Digital Engineering Department, Aeronautics, Space and Defense Division, Altran South West, France

ARTICLE INFO

Keywords:
Model composition
Model Driven Engineering
Systematic literature review

ABSTRACT

Context: Model Driven Engineering (MDE) aims to alleviate complexity and improve reusability in software development. The development of complex software implies to divide it into independent parts before then assembled. This is how the problem of model composition has become an interesting and stills an emerging topic in MDE.

Objective: Our goal is to analyze the current state of the art in model composition in the context of Model Driven Engineering.

Method: We use the systematic literature review based on the guidelines proposed by Biolchini et al., Brereton et al., and Kitchenham and Charters. We propose five research questions and six quality assessments.

Results: Of the 9270 search results, 56 have been considered relevant studies. These studies have resulted in 36 primary studies.

Conclusion: The evaluation shows that most of approaches allow more than two models as inputs of the composition, allow composing heterogeneous models and enable the tuning of the composition schema, while the important limitations are about the maturity of implementations and the lack on the management of future evolutions or backwards compatibility.

Contents

* Corresponding author.
E-mail address: anas.abouzahra@edu.uiz.ac.ma (A. Abouzahra).

https://doi.org/10.1016/j.infsof.2020.106316
Received 9 April 2019; Received in revised form 26 February 2020; Accepted 7 April 2020
Available online 8 May 2020
0950-5849/

1. Introduction

Since its emergence, Software Engineering (SE) aims to alleviate complexity and improve reusability. Indeed, Dijkstra [1] and Parnas [2] were among the precursors in the 70s to suggest the Separation of Concerns (SoC) principle to reduce the ever-growing complexity of systems by dividing them into units of behavior or units of function. Therefore, we need to assemble those units later using a composition processing technique. In fact, composing software units has always been a relevant topic in SE. Successive efforts have attempted to address the complexity of systems by the program composition. Thus, various approaches have been proposed in imperative programming, such as Chandy and Taylor [3], Vargas–Vera [4] and Martin and Lamport [5], arriving to Object Oriented Programming (OOP) which invites designers to decompose problems into functional units called objects. Unfortunately, according to Kiczales. [6] and Czarnecki et al. [7], OOP technologies were not sufficient to separate the main concern of a system from the non-functional and transversal concerns in many situations. However, the advent of Model Driven Engineering (MDE) [8], which principles are to upgrade the role of modeling and to increase the level of automation all along the SE development process, has substantially modified this landscape.

MDE is a model-centric development methodology [9] that raises the level of development to models instead of programs [10,11]. Models are system abstractions that allow developers better expressing their domain concerns. The centric role of models in MDE has directly influenced expressing composition between software units. Bernstein et al [12], Pottinger et al. [13], Kienzle et al. [14], Estublier et al. [15], Fleury et al. [16], Bézivin et al. [17], and Kolovos et al. [18] were among the first to be convicted that the application of MDE to larger and more complex systems contexts of necessity will involve the use of techniques for enabling flexible composition between models. Moreover, Mussbachar et al. in [19] have cited the model composition as one of the major fields of advancement relative to model transformation management in MDE.

Given the importance of composing software units in SE development, the promise of MDE to offer new possibilities to improve this operation, and that the software units handled in MDE are models, we consider it of great interest to perform a review of the literature regarding model composition in the context of MDE. Our goal is to discover how model composition is addressed, implemented and used in MDE and to identify possible areas of improvements.

In order to provide a comprehensive review of this topic, this work presents a Systematic Literature Review (SLR) [20–24] of model composition in the context of MDE. It aims to answer research questions such as "Which kind of correspondences between models are used in the current state of the art of MDE?" or “What level of automation is suggested for the creation of correspondences between models during the composition operation?”.

This paper is organized as follows: in Section 2, we give an overview of model composition in the context of MDE. Then, we describe the method we followed in Section 3. Next, we present results of the SLR in Section 4. Right after we answer research questions in Section 5. In Section 6, we discuss about the limitations of this SLR. Finally, in Section 7, we present our conclusions.

2. Background

Model composition is a growing research field in MDE that was inspired by works in Aspect Oriented Programming (AOP) [25], Generative Programming [7], Database Integration [26] and Model Transformation [27]. There are many open questions in this field and there are several proposals that address it. Nevertheless, there is not, yet, a standard vocabulary or a set of normalized definitions on model composition in MDE. That is why, for example, the Object Management Group (OMG) glossary of terms and acronyms [28] does not contains any appropriate definition relative to model composition. Thus, some consider model composition as an operation that is performed only on large modeling units and not at finer levels of granularity. While, some consider model composition as the fusion of two models or more without adding any new model elements. Others may consider model composition as the definition of relationships between elements of different models [29].

We define composition with its broader meaning as the act of creating some kind of links between modeling elements belonging to different models, regardless of whether they are heterogeneous (i.e. conforming to different metamodels) or homogeneous (i.e. conforming to the same metamodel). Furthermore, the links could be persisted in other models and, therefore, processed in different application scenarios (like tool interoperability or traceability) using any model processing technique (such as model transformation or model generation). Such a model composition scenario is represented by a very simplistic example in Fig. 1. Two models *Ma* and *Mb* are connected by an “extra” model *Mcomp* that represents links between *Ma* and *Mb*. The composition operation combines four petals of model *Ma* and a circle of model *Mb*. Thus, a model transformation *Mtrans* is produced by processing the model *Mcomp*. It maps petals and circle from source models *Ma* and *Mb* into a flower in the target model *Mc*.

We see in the aforementioned example that the composition links were collected in an “extra” model. However, they could be directly defined in the source models. Moreover, there are no clear similarities between composed elements, while sometimes a composition may be deduced using matching rules or heuristics. Similarly, many questions can be relevant to specify what a composition is and how it can be done; such as “Should all the information present in input models be present in the composed models?”, “Should the composition operation be performed with no duplicate information?”, “Should it be performed with no loss of information?”, “Should it produce some additional information?”, etc.

Furthermore, besides model composition, the second more used term in the literature is model merging which is actually a special case of model composition where all the information from the input models is preserved in the output models. Therefore, each choice and each terminology can lead to a different semantic, and a different representation of the composition operation. Our goal is obviously not to answer these questions or to take a position but to analyze how approaches that have addressed the subject of model composition in MDE have answered these

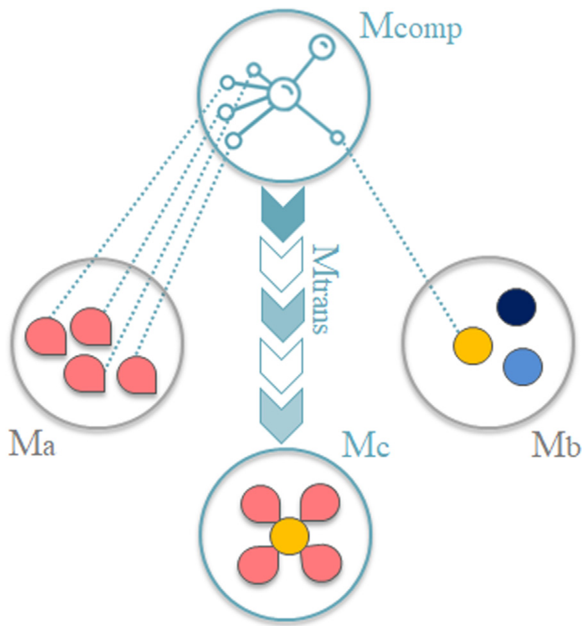

Fig. 1. An example of model composition in a MDE scenario.

questions. Therefore, these interrogations had been taken into consideration in the elaboration of this study. Especially when we have been focused on formulating research questions. However, we found it necessary to establish a set of definitions relating to the model composition operation. This allows giving an exact meaning to the terminology that we will use in the rest of this study.

To achieve this end, we relied on a study conducted by Bézivin et al. [30], as part of the project Modelware IST [31], where they proposed a canonical set of formal definitions regarding model composition, and a set of requirements for model composition frameworks. Those definitions and requirements aim, inter alia, to help comparing different model composition solutions. According to it, composition operation scenarios frequently imply:

- A way to capture correspondences between the models to compose.
- A way to specify which elements in the models to be composed can match and how they can match.
- A way to produce the composed model, whether it is a new model or not, virtual or persisted. A virtual model is a model whose “virtual” elements are proxies to elements contained in different models. A persisted model is a model which can be saved and whose “physical” elements are model elements representing relationships linking elements contained in different models.

Thereby, in the context of this review we consider the following definitions:

- *Match operation*: the match operation takes a set of models as input, searches for equivalences between their elements and produces a correspondence model as output.
- *Correspondence model*: a correspondence model represents links between elements of different models (which would correspond to the “extra model” of the aforementioned example).
- *Merge operation*: the merge operation takes a set of sources models and a correspondence model between them as input, and returns a model including all the information from source models, without duplicate information. The correspondence model can be created by the match operation to specify the elements that are going to be merged.
- *Compose operation*: the compose operation takes a set of models and a correspondence model between them as input and combines their elements into a new output model.

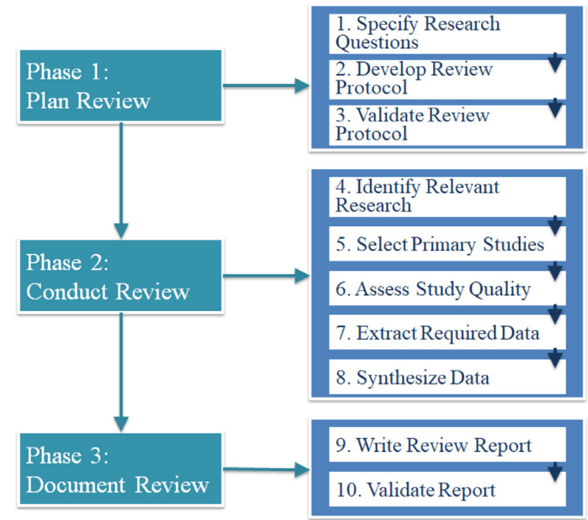

Fig. 2. The SLR process proposed by Brereton et al. in [22].

In addition, it is worth noting that even if merge is a special case of composition, we have considered approaches that addressed this issue, from the moment that the approach proposes a generic solution to merge models in the context of MDE.

3. Methodology

This study has been carried out by following Biolchini et al. [20], Brereton et al. [22] and Kitchenham and Charters [23] guidelines for conducting an SLR in SE. According to these guidelines, performing an SLR involves distinct activities, which can be grouped into three main phases: planning, conducting and reporting. Fig. 2 presents an overview of the ten steps review process that led to our study [22]. These steps give to the SLR process a sequential nature, but it is worth to note that many of them involve iterations. In particular, some steps are initiated during the protocol development phase and refined during the actual review [23]. Moreover, checkpoints can be added in order to improve the quality of the review and control its execution. Thereby, it is essential to evaluate the protocol before the second phase. If problems are found, it is necessary to return to the planning phase in order to review the protocol. Similar, if executing search to identify relevant researches gives unsatisfying, the study may be re-executed. Hence, we followed all these steps for the reported SLR as described in the following sections of this paper.

For the rest of this section, we will focus on the planning phase. This phase enables defining the research objectives and the way the review will be conducted. As shown in Fig. 3, the review protocol we elaborated consists of four successive processes: “Research Questions”, “Search, Assessments and Analysis Strategies”, "Primary Studies Selection" and “Data Extraction and Analysis”. A fifth process is conducted throughout all the protocol in order to record the results: "Packaging".

During the “Research Questions” process, research questions, which are arising from the main review subject, are defined, listed and explained, knowing that the broad objective of this review is to answer these questions. The “Search, Assessments and Analysis Strategies” process involves how we will search, filter, and evaluate studies; through two steps: “Specifying Data Sources and Search Strategy” and “Defining Assessments and Data Analysis Strategy”. Once strategies for searching, assessing and analyzing studies are fixed, the search is executed over the “Primary Studies Selection” process; through two steps: “Executing Search” and “Selecting Primary Studies”. It should be noted that a checkpoint is introduced to verify if the strategies defined in the previous process have been successful. Otherwise, the previous process is

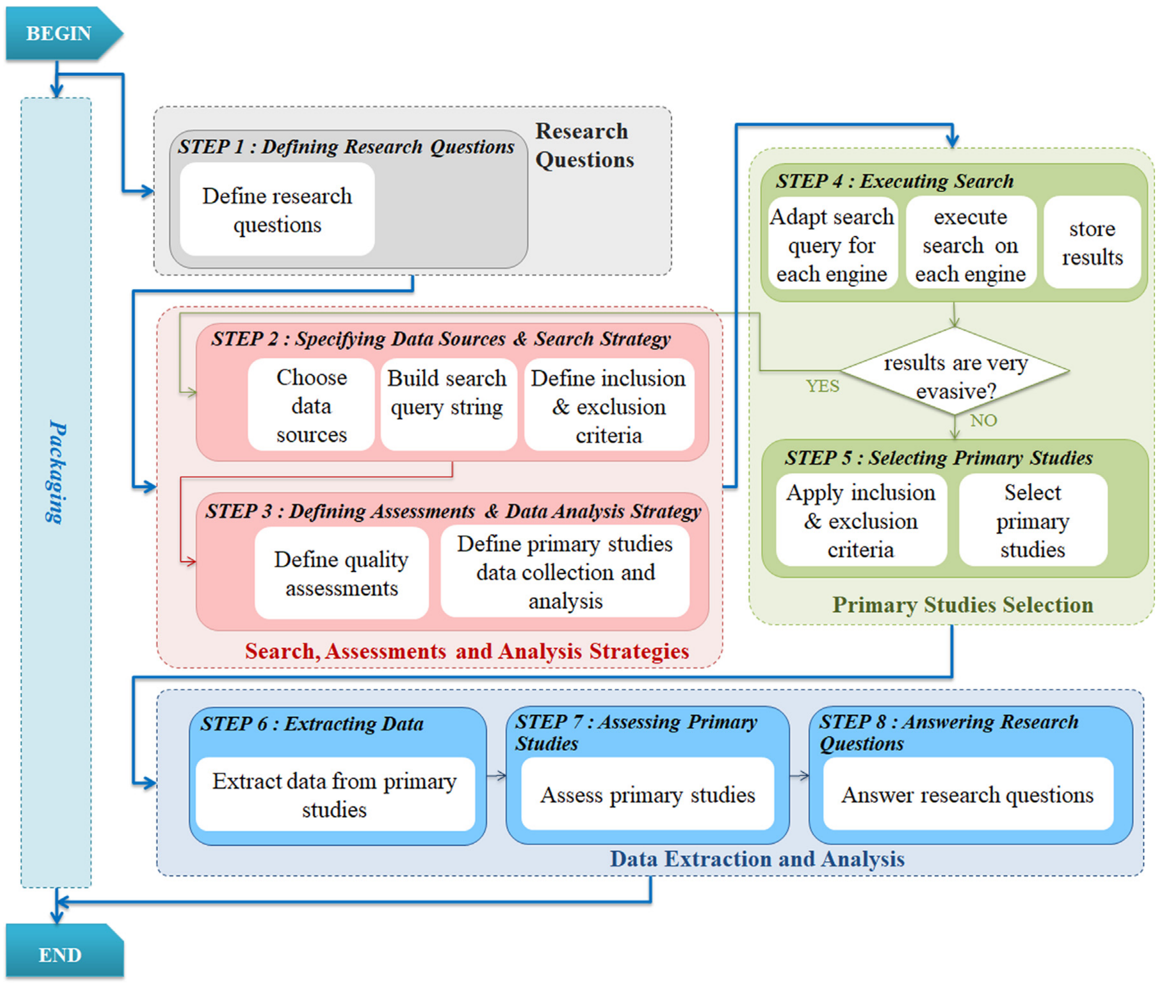

Fig. 3. The protocol followed to drive our SLR.

re-executed. As soon as we obtain primary studies, data are extracted, assessed, synthesized and analyzed during the “Data Extraction and Analysis” process; through three steps: "“Extracting Data"”, “Assessing Primary Studies” and “Answering Research Questions”. At the end of this process, we are able to answer research questions defined at the beginning of the review. Hence, the review reporting is achieved over the whole protocol through the “Packaging” process. This process involves a detailed record of the results and decisions made throughout the review. The final activity of this process is to format the results in order to circulate them to potentially interested parties.

Usually systematic reviews will be reported in at least two formats: (i) In a technical report or in a section of a PhD thesis; (ii) In a journal or conference paper. A journal or conference paper will normally have a size restriction.

In this paper, we followed the structure and contents of reports used in [24]. This structure is appropriate for technical reports and journals. As our project fits into a doctoral thesis, we kept a detailed record of decisions made throughout the review process in order to report them in a chapter of the PhD thesis manuscript.

Accordingly, the next subsections are organized as follows: Research Questions (Section 3.1), Data Sources (Section 3.2), Search Query String (Section 3.3), Inclusion and Exclusion Criteria (Section 3.4), Quality Assessments (Section 3.5) Data Collection and Analysis (Section 3.6).

3.1. Research questions

Specifying the Research Questions (RQs) is the most important activity at the planning stage but also the most important part of any SLR, because it drives the entire review methodology [23]. Considering that the goal of our SLR is to identify and analyze the state of the art in model composition in the context of MDE, we intend to answer the following research questions:

- *RQ1. Which kinds of correspondences (i.e. composition schema) between models are used in the current state-of-the art in model composition in the context of MDE?*

As afromentioned, authors of different proposals often use other terms than “composition” like “merging” or “weaving”. It corresponds to their specific interpretation of model composition. Therefore, we are interested to classify the different proposals about the semantic they give to the model composition operation, in light of the definitions listed in Section 2.

- *RQ2. What level of automation is suggested for the creation and the management of links between models during the composition operation?*

According to Bézivin et al. [30] one desirable and practical requirement for a model composition framework is the ability to provide the means for minimizing the effort expended by the developer to write composition or merge operations [29]. Thus, we are interested to know how the different proposals deal with certain tasks related to automate the composition process (e.g., by allowing composing rules to be inferred by metamodel structure or other matching strategies or by allowing composition expressions to be reused).

- *RQ3. What value can be drawn from the literature with regard to the use of model composition in the context of MDE?*

Table 1
Libraries used in the search.

Library	Source
ACM Digital Library [ACM]	http://dl.acm.org/
IEEE Xplore [IEEE]	http://ieeexplore.ieee.org/
Science Direct [SCD]	http://www.sciencedirect.com/
SpringerLink [SPL]	http://www.link.springer.com/

One of the main motivations of this systematic literature review is to find out whether the model composition in the context of MDE has succeeded to address the challenge mentioned in Section 2 that is the application of MDE into larger and more complex systems contexts. The critical properties of a complex system are size, distribution, and heterogeneity [17]. We therefore aim to determine which of them were addressed in the proposals. Furthermore, we are interested to know whether proposals have provided the user with any implementation, framework or technological support that has been largely deployed and applied to significant use cases.

- *RQ4. What are the limitations of state of the art in model composition in the context of MDE?*

The aim of this research question is to identify limitations and points of weakness when dealing with model composition proposals in the context of MDE. The goal is to evaluate whether there is space for improvement in the area and to assess the possibility of addressing them in future works.

- *RQ5. Are there favored journals or conferences where model composition in MDE proposals are published, and what is the quality of these forums?*

The aim of this research question is to identify where studies related to model composition in the context of MDE have been published in order to analyze whether there are any journals or conferences that specialize in this topic. In addition, we are interested in identifying the quality and importance of these forums based on scientometric indexes and quality assessments widely accepted by the academic community (e.g. the Computing Research and Education Association of Australasia (CORE) Conference Ranking and the SCImago Journal Rank (SJR) Indicator).

3.2. Data sources

We searched for studies, written in English, published online before December 2018 and available on digital libraries. Libraries were chosen according to characteristics such as, ease of use, accessibility of documents, spectrum of indexed documents (i.e. journals articles, conference proceedings, books and books chapters), ability of their search engines to take a complex textual query (i.e. using logical operators), accuracy of search (i.e. the engine respects the query as specified and does not modify it) and reproducibility of results. The selected digital libraries to carry out the search process of this review are presented in Table 1.

3.3. Search query string

All selected digital libraries have search engines based on keywords. The query string that was used in search engines was build according to the examination of the subject keywords of the SLR (Fig. 4).

On the one hand, besides ""“composition”, as we mentioned before, other terms are often mixed up in the literature to mean model composition operation in MDE. Consequently, the term “composition” is not enough to ensure complete and accurate results. Thus, we decided to add synonym terms as “merging"”, “weaving”, “linking”, “connecting”, “integrating” and “relating”; in plus of their nearby words. On the other hand, in addition to “Model Driven Engineering”, names of other model driven engineering approaches are used in the literature like “Model Driven Architecture” (MDA), “Model Driven Development” (MDD), “Model Driven Software Development” (MDSD) and “Model Driven Software Engineering” (MDSE). Thus, we added all this variations with their acronyms as part of the keywords of interest. Eventually, we used logical operators “OR” and “AND” in order to specify the search query string as described in Fig. 4. It is worth mentioning that the search engine of each digital library uses its own syntax. Therefore, the canonical search query string has been adapted to each one of them.

It is important to note that the final query string was adopted after several executions of other query strings that did not give satisfactory results during the “Primary Studies Selection” process (i.e. step 4). As described previously in the protocol of the SLR, we returned to the previous processes (i.e. step 2) and rebuilt a new string until obtained pertinent results.

3.4. Inclusion and exclusion criteria

As the guidelines of [23] suggest, we defined inclusion and exclusion criteria. They specify what makes a study among results of search in digital libraries to be retained or to be rejected. Accordingly, only studies fulfilling inclusion criteria and not excluded by exclusion criteria were retained and called Primary Studies (PSs).

An Inclusion Criterion (IC) was defined to test of the relevance and compatibility of each study with the review main goal. Studies applying to ICs are called henceforth Candidate Studies (CSs). We define two ICs:

- *IC1.* The study title must include the word "model" as well as the word "composition" or one of its synonyms or nearby words from those specified in Section 3.3.
- *IC2.* The lecture of the study abstract must lead to conclude that the main purpose of the study is model composition in the context of MDE.

An Exclusion Criterion (EC) was defined to refine the selection of CSs. Studies remaining after applying ECs are called henceforth Relevant Studies (RSs). We define three ECs:

- *EC1.* The CS acknowledges model composition as an important model operation but does not provide a model composition approach.
- *EC2.* The CS deals with model composition in a special use case without providing a general approach.
- *EC3.* The CS is a review of the literature in model composition.

In addition, one time the RSs obtained, if an RS have one or more authors in common with another RS and their ideas belong to the same research, we called them Interrelated Studies (ISs). We kept only the most detailed and complete one as PS.

3.5. Quality assessments

According to the guidelines proposed by Kitchenham and Charters [23], once we had selected PSs, we have to assess the quality of the researches they present and quantitatively compare them, by the mean of Quality Assessments (QAs). Therefore, we defined six QAs for our SLR:

- *QA1. Is the purpose of the study clearly stated?*
- *QA2. Is the interest and the usefulness of the work clearly presented?*
- *QA3. Is the study methodology clearly established?*
- *QA4. Are the concepts of the approach clearly defined?*
- *QA5. Is the work compared and measured with other similar work?*
- *QA6. Are the limitations of the work clearly mentioned?*

The scoring procedure used to evaluate each QA was:
Yes (Y) = 1, Partly (P) = 0.5 or No (N) = 0.

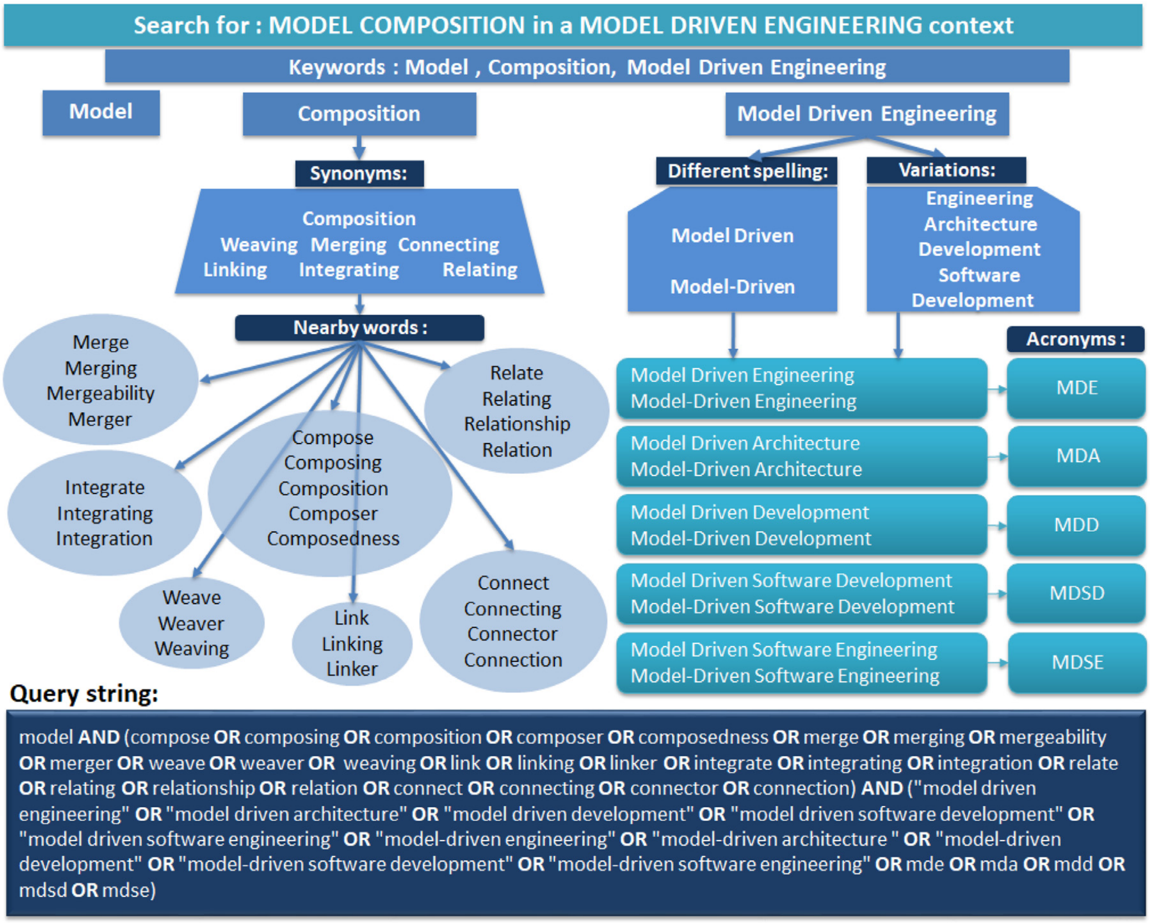

Fig. 4. The building of the search query string.

3.6. Data collection and analysis

A data extraction phase adapted from Kitchenham and Charters [23] was followed for each PS in order to collect all information needed to answer the SLR RQs. This phase has led to two kinds of data: objective data composed of basic data like title, authors, publication and publication year; and subjective data composed of information, extracted from each PS, relating to the SLR RQs. Objective data enabled to identify each PS and was used for their classification. When subjective data allowed carrying out an in-depth analysis of each PS. If the objective data are obvious to extract, the subjective ones require analysis. To simplify the analysis of these data we relied on the resulting requirements of the study made by Bézivin et al. [30]. Thereby, we extracted information from each PS (if specified) regarding a set of 12 Analysis Assessments (AAs). The scoring procedure used to evaluate each one of them was the same as for the QAs (i.e. Y = 1, P = 0.5, N = 0). The AAs questions specified in this SLR were:

- *AA1. Whether the proposal of the PS expresses means to carry out match operation.*

Scores: Y (Yes), the proposal defines and integrates the matching operation as a preliminary operation to the composition operation; P (Partly), it suggests to realize a matching before the composition operation and provides ideas to perform it; N (No), the proposal does not consider matching operation.

- *AA2. Whether the proposal of the PS integrates the use of the correspondence model in the composition process.*

Scores: Y (Yes), the proposal incorporates the use of a correspondence model to represent the links between the elements to be composed. This model is a full-fledged model that conforms to a metamodel and that can be persisted; P (Partly), the proposal uses a temporary model of correspondence (or a pseudo-model, e.g. schema of correspondence) that serves to assist the construction of the composition but that is not saved as a full-fledged model; N (No), the proposal does not consider correspondence model.

- *AA3. Whether the proposal of the PS covers the general case of composition operation.*

Scores: Y (Yes), the proposal enables to weave the composition freely and to give to possibility to adjust the composed model afterwards; P (Partly), the proposal gives the possibility to weave the composition but the result is a merging of the models. There is no possibility to adjust the composed model afterwards; N (No), the proposal does not make it possible to influence composition; it is automatically performed after a match operation.

- *AA4. Whether the proposal of the PS provides the means for minimizing the effort made to compose models elements.*

Scores: Y (Yes), the proposal furnishes automatic matching techniques to identify corresponding elements in the models that are to be composed (e.g. heuristic-based matching strategies); P (Partly), the proposal provides techniques to realize matching either semi-automatically (e.g. language to define matching rules); N (No), the proposal does not consider matching operation.

- *AA5. Whether the proposal of the PS provides means for traceability and navigability between composed model elements and their source elements in the original models.*

Scores: Y (Yes), once the composition is done, the proposal allows to easily navigate from the elements of the composed model to the source model elements in the original models; P (Partly), once the composition is done, the proposal furnishes ideas to implement means to access to the source model elements from the composed model; N (No), the proposal does not consider links between the elements of the composed model and their source elements in the original models after the composition is done.

- *AA6. Whether the proposal of the PS has been implemented.*

Scores: Y (Yes), the proposal provides a complete implementation (tool or framework) to support its methodological solution; P (Partly), the proposal includes only a prototype implementation or a partial implementation; N (No), the proposal is only a theoretical proposal, lacking implementation.

- *AA7. Whether the proposal of the PS has been adopted by a wide software community or supported by a software editor.*

Scores: Y (Yes), the proposal is adopted by an international consortium in the world of software engineering (e.g. OMG) or approved / integrated by a software community of high impact (e.g. Eclipse project) or supported by a renowned software company; P (Partly), the proposal is a project that interests an international consortium or a high impact software community or a renowned company in the world of software engineering but it stills in an incubation phase; N (No), the proposal has not generated much interest in the software community at the moment.

- *AA8. Whether the proposal of the PS has been the subject of significant use cases.*

Scores: Y (Yes), the proposal has been used in industrial projects in complex system contexts and proved its ability to achieve real scalability; P (Partly), the proposal has been used in several interesting and relevant use cases being published/communicated in recognized journals/conferences and which promises to achieve scalability; N (No), the proposal has not been the subject of use cases or has been used only in small demonstrations to illustrate the methodology.

- *AA9. Whether the proposal of the PS allows composing an unlimited number of input models.*

Scores: Y (Yes), the proposal does not restrict the number of models to be composed; P (Partly) - this choice is not relevant to this question; N (No), the proposal allows to compose only two models.

- *AA10. Whether the proposal of the PS allows composing heterogeneous models.*

Scores: Y (Yes), the proposal allows composing heterogeneous models; P (Partly), the proposal does not allow composing heterogeneous models but allows composing models conforming to different versions of the same metamodel; N (No), the proposal allows to compose only homogeneous models.

- *AA11. Whether the proposal of the PS enables the design of the composition metamodel.*

Scores: Y (Yes), the proposal gives means to design the schema of composition, i.e. the metamodel of correspondence model; P (Partly), the proposal furnishes a unique schema of composition. However, it is possible to extend it through external tools; N (No), the proposal furnishes a unique and rigid schema of composition.

- *AA12. Whether the proposal of the PS supports as well future extension as backwards compatibility for composed models.*

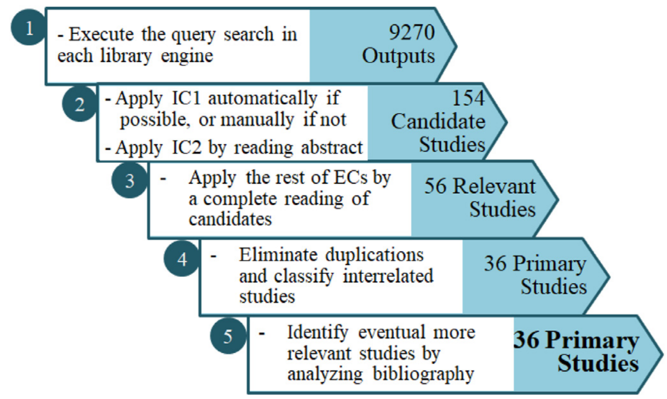

Fig. 5. Activities of the "Primary Studies Selection" process.

Scores: Y (Yes), the proposal automatically guarantees to support future extension and backwards compatibility for composed models; P (Partly), the proposal provides means to update the composition in order to support future extension or backwards compatibility for composed models; N (No), the proposal does not support future extension, nor it supports backward compatibility.

Evaluating these AAs for each PS allowed us to answer the four first RQs as follows:

- *AA1, AA2 and AA3 allowed answering RQ1.*
- *AA4 and AA5 allowed answering RQ2.*
- *AA6, AA7 and AA8 allowed answering RQ3.*
- *AA9, AA10, AA11 and AA12 allowed answering RQ4.*

In addition, the basic data extracted from each study enabled the categorizing of the studies according to their publication, which allowed answering RQ5.

4. Results

In this section, we present the main results of our SLR in the respect of the protocol defined in the previous section.

4.1. Search results and primary studies selection

The process “Primary Studies Selection” has started with the execution of the search in the search engine of each library and has brought to the selection of the PSs. It consists of five successive activities, which are mentioned in Fig. 5. This section details the execution of this process.

As mentioned previously, the string used as search query for each search engine was adapted to the engine search tips. In addition, each time it was possible in the search engine, we incorporated the inclusion criterion IC1 into the search query, limited the search to the common fields: title, abstract and keywords, and restricted the search to the scope of “Computer Science” or “Software Engineering”. Table 2 summarizes the different search configurations performed on each library and shows the number of obtained results. As a result, the number of outputs obtained was a total of 9270 results. This has been reduced to 1141 results by including IC1. Then, we proceeded to read the abstracts of the remaining results in order to apply IC2. This resulted in 154 CSs. It represents only 1.66% per cent of the 9270 initial outputs.

The next activity was to gather all outputs in order to apply ECs. We proceeded first by treating results of each digital library and obtained RSs. Once we gathered all the RSs from all digital libraries, we filtered ISs and eliminated duplications. It resulted in 56 RSs, which represents a little more than the third of the CSs.

Accordingly, Table 3 summarizes the number of results for each data source, those filtered as candidates and those remaining as relevant. Thus, column 2 shows the total of outputs recorded from each digital

Table 2
Adapted query strings for each library search engine.

Search engine	Common fields	Scope	Total outputs	+ IC1	Remaining outputs
ACM	Title, abstract and author keywords	Full-Text Collection	1058	Automatically	117
IEEE	Title text, abstract and indexing terms)	Conferences, Journals & Magazines	1002	Automatically	407
SCD	Title, abstract and keywords	Computer Science	981	Manually	27
SPL	All	Software Engineering	6229	Manually	590
All data sources	**1141**				

Table 3
Search results.

Search engine	Total Outputs	CSs	% of CSs	RSs	% of RSs	% of All RSs
ACM	1058	30	2.84	12	1.13	21.43
IEEE	1002	32	3.19	14	1.40	25.00
SCD	981	12	1.22	4	0.41	7.14
SPL	6229	80	1.28	26	0.42	46.43
All engines	**9270**	**154**	**1.66**	**56**	**0.60**	**100**

Table 4
Elimination of duplications and interdependent studies.

	Studies	Percentage
Relevant studies	56	100
Duplicated studies	3	5.36
interrelated studies	17	30.36
Primary studies	**36**	**64.29**

library. Then, column 3 shows the number of CSs found in each digital library. Whereas column 4 shows their percentage, e.g. 1.28% per cent of the studies returned by the SPL library were identified as CSs. Column 5 shows the number of RSs found in each digital library, whereas column 6 shows their percentage, e.g. 0.42% per cent of the studies returned by the SPL library were identified as RSs. Finally, column 6 shows the percentage of the RSs retained from each digital library regarding to the total number of RSs.

In this respect, we can note that the inclusion and exclusion criteria were used to screen a large number of results. The number of RSs represents only 0.60% per cent of the total number of studies returned by the different search engines. However, ICs have been much more effective. They eliminated a very large number of results; more than 98% per cent of initial results. This is primarily because many of the returned results were turned out to contain some strings of the search query, but were not related to the research questions of this review. The ECs refined the selection by eliminating two-thirds of the remaining candidate studies. Actually, an important part of them, when closely inspected, has been found to be dealing with model composition without providing a general approach in the context of MDE. They used model composition in a special use case, they focused only on models decomposition, or they assessed existing model composition approaches.

Overall, the IEEE and ACM libraries garnered the highest percentages of relevant studies. This is because they are less generalist than the two other libraries, which are more oriented Computer Science. Furthermore, we can note that most of final RSs were found in the SPL library. In fact, during the search activity we noticed that there are many studies dealing with model composition in the context of MDE published as chapters of some SPL book series like Lecture Notes in Computer Science (LECT NOTES COMPUT SC). Therefore, we decided to include chapters in the research criteria of this search engine. It resulted to a large number of outputs and allowed us to collect almost half of relevant studies.

Finally, after identifying all RSs, the final step led to the selection of PSs. Essentially, it consisted of two tasks: removing possible duplications and factoring sets of interrelated studies. Accordingly, Table 4 shows that the third of RSs were duplicated in different data sources or they belong to the same research. Therefore, we obtain the list of 36 PSs for our SLR. It is important mentioning that, after studying the bibliographies of the selected PSs, we have not identified any other relevant study different from those already recorded.

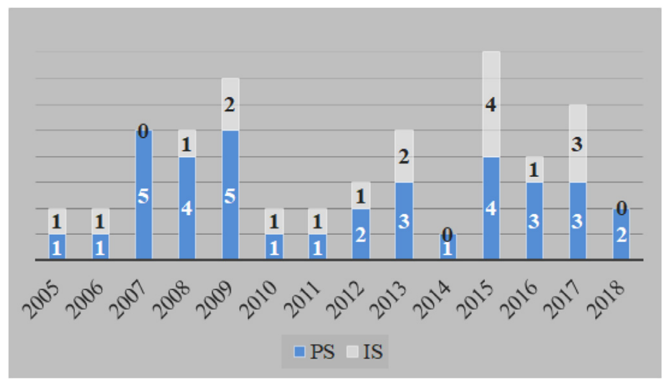

Fig. 6. The evolution of number of publications of PSs & ISs over year.

Table 5 contains the complete list of PSs according to their authors, title, year of publication and where they were published. If the PS is published in a journal following a previous communication at a conference, it is cited in parentheses. In addition, column 6 cites the eventual interrelated studies that were eliminated from PSs. We thought it important to trace these references as additional information.

Fig. 6 shows how the publications of PSs and their ISs are spread over time. Observing the left part of the chart, we can note a growth in publications until the year 2009 before an abrupt fall. This change in interest can be justified by the same trend of interest around the MDE, which had a slow period after 2009. However, it has since recovered. Driven by this new wave, a renewal of interest to the issue of models composition in MDE has appeared and persists in recent years. This is what we can note on the right part of the chart.

4.2. Quality assessment results

Once the PSs were selected, the next step was to assess them against the QA questions defined in Section 3.5. The score assigned to each PS for each QA is shown in Table 6. The results show that all PSs scored at least 3 points out of a maximum of 6 points. The last column ("% by max PS") presents the percentage obtained by each PS out of the maximum score (i.e., 6). The penultimate row ("% total score") presents the percentage of points attained by all the PSs relative for a QA question relative to the total number of points scored by all the PSs in all the QA questions. Finally, the last row ("% max QAs") shows the percentage of

Table 5
Primary studies.

ID	Author/s [ref.]	Title	Year	PS publication	IS publication (year) [ref.]
PS1	Jézéquel [32]	Model driven design and aspect weaving	2008	SOFTW SYST MODEL	
PS2	Fuentes and Sánchez [33]	Dynamic weaving of aspect-oriented executable UML models	2009	LECT NOTES COMPUT SC	LECT NOTES COMPUT SC (MODELS), 2008 [34]
PS3	Kramer, Klein, Steel, Morin, Kienzle, Barais and Jézéquel [35]	Achieving practical genericity in model weaving through extensibility	2013	LECT NOTES COMPUT SC (ICMT)	
PS4	Barais, Klein, Baudry, Jackson and Clarke [36]	Composing multi-view aspect models	2008	ICCBSS	
PS5	Marchand, Combemale and Baudry [37]	A categorical model of model merging and weaving	2012	MISE	
PS6	Fleurey, Baudry, France and Ghosh [16]	A generic approach for automatic model composition	2007	LECT NOTES COMPUT SC (MODELS)	
PS7	Kolovos, Paige and Polack [38]	Merging models with the epsilon merging language (EML)	2006	LECT NOTES COMPUT SC (MODELS)	
PS8	Boubakir and Chaoui [39]	A pairwise approach for model merging	2016	LECT NOTES COMPUT SC (MISC)	
PS9	Berg and Møller-Pedersen [40]	Metamodel and model composition by integration of operational semantics	2015	COMM COM INF SC (MODELSWARD)	MODELSWARD, 2015 [41]; LECT NOTES COMPUT SC (SAM), 2012 [42]; COMM COM INF SC (MODELSWARD), 2015 [43]
PS10	Anwar, Benelallam, Nassar and Coulette [44]	A graphical specification of model composition with triple graph grammars	2013	LECT NOTES COMPUT SC (MOMPES)	ICECCS, 2011 [45]
PS11	Westfechtel [46]	Merging of EMF models	2014	SOFTW SYST MODEL	ACMTE, 2013 [47]; SCI COMPUT PROGRAM (MODELSWARD), 2015 [48]; IWMCP, 2010 [49]
PS12	Estublier, Vega and Ionita [15]	Composing domain-specific languages for wide-scope software engineering applications	2005	LECT NOTES COMPUT SC (MODELS)	ASWEC, 2005 [50]
PS13	Haber, Look, Nazari, Perez, Rumpe, Völkel and Wortmann [51]	Composition of heterogeneous modeling languages	2015	COMM COM INF SC (MODELSWARD)	MODELSWARD, 2015 [52]
PS14	Meyers, Cicchetti, Guerra and De Lara [53]	Composing textual modelling languages in practice	2012	MPM	SOFTW SYST MODEL, 2013 [54]
PS15	Whittle, Jayaraman, Elkhodary, Moreira and Araújo [55]	MATA: a unified approach for composing UML aspect models based on graph transformation	2009	LECT NOTES COMPUT SC	
PS16	Mansoor, Kessentini, Langer, Wimmer, Bechikh and Deb [56]	MOMM: multi-objective model merging	2015	J SYST SOFTWARE	
PS17	Herrmann, Krahn, Rumpe, Schindler and Völkel [57]	An algebraic view on the semantics of model composition	2007	LECT NOTES COMPUT SC (ECMDA-FA)	
PS18	Yie, Casallas, Deridder and Wagelaar [58]	Deriving correspondence relationships to guide a multi-view heterogeneous composition	2010	LECT NOTES COMPUT SC (MODELS)	
PS19	Hovsepyan, Van Baelen, Berbers and Joosen [59]	Specifying and composing concerns expressed in domain-specific modeling languages	2009	LECT NOTES BUS INF (TOOLS EUROPE)	
PS20	Weisemöller and Schürr [60]	Formal definition of MOF 2.0 metamodel components and composition	2008	LECT NOTES COMPUT SC (MODELS)	
PS21	Strüber, Taentzer, Jurack and Schäfer [61]	Towards a distributed modeling process based on composite models	2013	LECT NOTES COMPUT SC (FASE)	LECT NOTES COMPUT SC (MODELS), 2009 [62]
PS22	Bartelt [63]	Consistence preserving model merge in collaborative development processes	2008	CVSM	
PS23	Sabetzadeh, Nejati, Easterbrook and Chechik [64]	A relationship-driven framework for model merging	2007	MISE	

(continued on next page)

Table 5 (*continued*)

ID	Author/s [ref.]	Title	Year	PS publication	IS publication (year) [ref.]
PS24	Schmidt, Wenzel, Kehrer, and Kelter [65]	History-based merging of models	2009	CVSM	
PS25	Defen Zhang, Shixian Li and Xianming Liu [66]	An approach for model composition and verification	2009	NCM	
PS26	Dam, Egyed, Winikoff, Reder and Lopez-Herrejon [67]	Consistent merging of model versions	2016	J SYST SOFTWARE	
PS27	Boronat, Carsí, Ramos and Letelier [68]	Formal model merging applied to class diagram integration	2007	ELECTRON NOTES THEOR COMPUT SCI	
PS28	Abouzahra, Sabraoui and Afdel [69]	A practical approach for extending DSMLs by composing their metamodels	2018	ADV SCI TECHNOL ENG SYST J	ECEEC, 2017 [70]
PS29	Bill, Neubauer and Wimmer [71]	Virtual textual model composition for supporting versioning and aspect-orientation	2017	SLE	
PS30	Rabbi, Lamo and Kristensen [72]	A model driven engineering approach for heterogeneous model composition	2017	COMM COM INF SC (MODELSWARD)	MODELSWARD, 2017 [73]
PS31	El Marzouki, Nikiforova, Lakhrissi and Mohajir [74]	Toward a generic metamodel for model composition using transformation	2017	PROCEDIA COMPUTER SCIENCE	IT4OD, 2016 [75]; WITS, 2017 [76]
PS32	Taylor and Sharma [77]	Model composition via object-role modeling	2018	SOUTHEASTCON	
PS33	Didonet Del Fabro and Valduriez [78]	Semi-automatic model integration using matching transformations and weaving models	2007	ACM SAC	SOFTW SYST MODEL, 2009 [79]; LECT NOTES COMPUT SC (ECMDA-FA), 2006 [30]
PS34	Roychoudhury, Gray and Jouault [80]	A model-driven framework for aspect weaver construction	2011	LECT NOTES COMPUT SC	
PS35	Bowles, Alwanain, Bordbar and Chen [81]	Matching and merging scenarios automatically with Alloy	2015	COMM COM INF SC (MODELSWARD)	
PS36	Debreceni, Ráth, Varró, De Carlos, Mendialdua and Trujillo [82]	Automated model merge by design space exploration	2016	LECT NOTES COMPUT SC (FASE)	

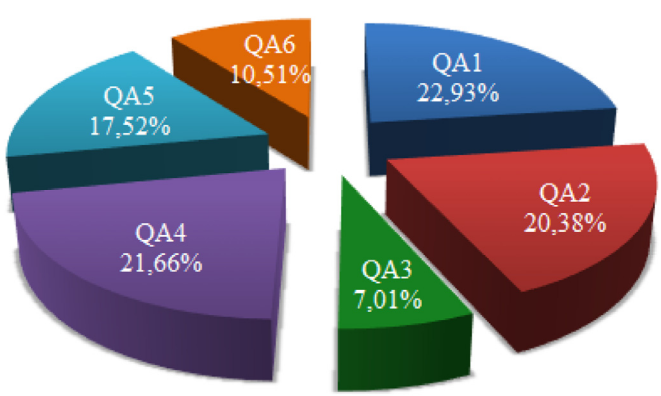

Fig. 7. The distribution of scores of QA questions.

points obtained by all the PSs for a given QA question out of the points that would be obtained if every PS obtained the highest score (i.e. 36). The arithmetic mean of the scores is 4.36 (i.e. 72.69% per cent) and the standard deviation is 0.76.

Fig. 7 presents a pie chart illustrating the distribution of scores of QAs. It shows that the questions Q1, Q2, Q4 and Q5 are around 20% per cent, while questions Q3 and Q6 (research methodology and limitations) represent a score close to 10% per cent. Accordingly, we can observe that the specification of the research methodology was a major handicap for the evaluated PSs, given that it was clearly presented by only six PSs, namely PS8, PS16, PS26, PS28, PS29 and PS36. The other qualitative aspect to be improved is the specification of limitations. Only a quarter of the PSs considered it. This lack reflected, in some respects, a gap in the internal validation of a study. It worth important to note that the three PSs which specified a clear research methodology and presented limitations of their approaches, obtained the maximum score of 100% per cent (i.e. PS16, PS29 and PS36).

In view of these results, we can conclude that in general the quality of PSs is good given that all of them have obtained a quality score higher than or equal to 50% per cent.

4.3. Data extraction results

Once the PSs have been assessed with QAs, we proceeded to the evaluation of the AA questions defined in section 3.6 and got scores for each PS. The scores are presented in Table 7. The penultimate row ("% total score") shows the percentage of points obtained by all the PSs for a given AA with regard to the total number of points obtained by all the PSs in all AAs. The last row ("% Max AAs") indicates the percentage of points reached by a given AA over the points that would be scored if every PS got the highest score (i.e. 36). Fig. 8 shows a pie chart depicting the distribution of scores for AAs ("% total score") and Fig. 9 shows a spider chart depicting the percentage of scores for each AA question ("% Max AAs").

By analyzing these results, we noted that all the PSs have addressed issues of AA questions. In addition, except for questions AA4, AA7 and AA8, the "% total score" is close to 10% per cent where the "% Max AAs" is close to 50% per cent. These results prove the relevance of the AA questions as they illustrate the variability of how the PSs treat the issue of model composition in the context of MDE. In contrast, the AA4, AA7 and AA8 question scores give the impression that existing solutions for modeling MDE models have low maturity. Because these AAs were defined inter-alias to evaluate the maturity and the level of industrialization of solutions brought by the PSs. However, we do not rush into hasty statements. The AAs results analysis step in the next section allow us to better address this question and answer the RQs.

Table 6
Quality assessment of PSs.

ID	QA1	QA2	QA3	QA4	QA5	QA6	Total score	% by Max PS
PS1	Y	Y	N	Y	N	Y	4	66.67
PS2	Y	P	P	P	Y	Y	4.5	75.00
PS3	Y	P	P	Y	Y	P	4.5	75.00
PS4	Y	Y	N	Y	Y	N	4	66.67
PS5	Y	Y	N	Y	Y	N	4	66.67
PS6	Y	Y	N	P	P	Y	4	66.67
PS7	Y	Y	P	Y	N	N	3.5	58.33
PS8	Y	Y	Y	Y	P	N	4.5	75.00
PS9	Y	P	P	Y	Y	N	4	66.67
PS10	Y	Y	N	Y	Y	N	4	66.67
PS11	Y	Y	N	Y	Y	Y	5	83.33
PS12	Y	Y	N	P	Y	N	3.5	58.33
PS13	Y	Y	N	Y	Y	N	4	66.67
PS14	Y	Y	N	Y	Y	N	4	66.67
PS15	Y	Y	N	Y	Y	Y	5	83.33
PS16	Y	Y	Y	Y	Y	Y	6	100.00
PS17	Y	P	N	Y	Y	P	4	66.67
PS18	Y	Y	N	Y	Y	N	4	66.67
PS19	Y	P	P	Y	P	N	3.5	58.33
PS20	Y	Y	N	Y	P	Y	4.5	75.00
PS21	Y	Y	P	Y	Y	N	4.5	75.00
PS22	Y	Y	N	Y	N	Y	4	66.67
PS23	Y	Y	P	P	N	P	3.5	58.33
PS24	Y	P	N	Y	P	N	3	50.00
PS25	Y	Y	P	Y	Y	N	4.5	75.00
PS26	Y	Y	Y	Y	Y	P	5.5	91.67
PS27	Y	Y	P	Y	Y	Y	5.5	91.67
PS28	Y	Y	Y	Y	Y	N	5	83.33
PS29	Y	Y	Y	Y	Y	Y	6	100.00
PS30	Y	Y	N	Y	Y	N	4	66.67
PS31	Y	N	N	Y	P	Y	3.5	58.33
PS32	Y	Y	N	Y	P	Y	4.5	75.00
PS33	Y	Y	N	Y	Y	Y	5	83.33
PS34	Y	Y	P	Y	P	P	4.5	75.00
PS35	Y	Y	N	Y	P	N	3.5	58.33
PS36	Y	Y	Y	Y	Y	Y	6	100.00
Total	**36.00**	**32.00**	**11.00**	**34.00**	**27.50**	**16.50**	**157.00**	**Mean = 72.69%**
% total score	**22.93**	**20.38**	**7.01**	**21.66**	**17.52**	**10.51**	**100.00**	
% by Max QAs	**100.00**	**88.89**	**30.56**	**94.44**	**76.39**	**45.83**		

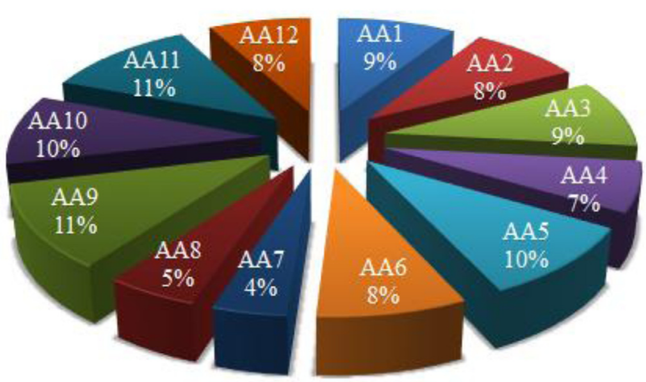

Fig. 8. The distribution of scores of AA questions.

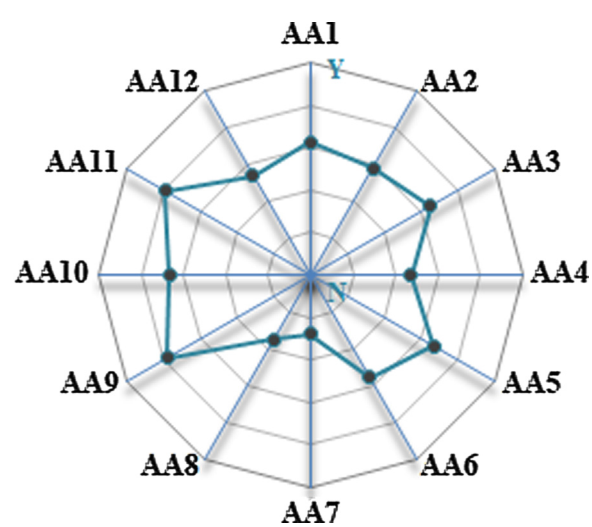

Fig. 9. The % Max score of each AA question.

5. Discussion

This section discusses how the data extracted from the PSs allowed us to answer our SLR RQs. By investigating the five RQs, we aim to provide a synthesized overview of the literature on model composition in the context of MDE.

5.1. PSs families

Before answering the RQs, we found important to note that, during the data extraction activity, we noticed that the PSs approaches presented some redundant areas of inspiration. We could classify them in four major families. By consequence, we decided to highlight them to better analyze the AAs scores and to make richer answers for our RQs by considering them from the point of view of these families. It is important to point out that this classification, as well as the pathway towards its manifestation, is not part of our "systematic" method. We have not

Table 7
Analysis assessment of PSs.

ID	AA1	AA2	AA3	AA4	AA5	AA6	AA7	AA8	AA9	AA10	AA11	AA12	Total Score
PS1	Y	Y	P	P	N	Y	Y	P	Y	N	Y	P	8
PS2	Y	P	Y	P	Y	Y	N	P	Y	N	Y	P	8
PS3	Y	Y	Y	Y	P	P	N	P	Y	P	Y	P	8.5
PS4	Y	Y	Y	Y	N	N	N	N	N	N	Y	N	5
PS5	P	Y	Y	P	P	N	P	N	Y	Y	Y	N	7
PS6	Y	N	P	P	N	N	N	N	Y	N	Y	N	4
PS7	Y	N	P	P	Y	Y	Y	Y	Y	Y	P	P	9
PS8	Y	N	N	Y	N	P	N	N	Y	P	N	P	4.5
PS9	N	P	Y	N	Y	P	N	N	Y	Y	Y	Y	7
PS10	Y	Y	P	P	P	Y	N	P	Y	Y	Y	Y	9
PS11	Y	P	N	Y	N	P	N	N	Y	Y	P	P	6
PS12	N	Y	Y	N	P	N	P	P	Y	Y	Y	Y	7.5
PS13	N	N	Y	N	Y	Y	N	Y	Y	Y	P	P	7
PS14	N	Y	Y	N	P	P	P	N	Y	Y	Y	N	6.5
PS15	Y	Y	P	P	Y	Y	P	Y	Y	Y	Y	P	10
PS16	Y	N	N	Y	N	P	P	P	Y	P	N	Y	6
PS17	N	N	Y	N	P	P	P	P	N	Y	Y	N	5
PS18	Y	Y	P	P	Y	P	N	N	Y	Y	Y	P	8
PS19	N	Y	Y	N	Y	N	N	N	Y	Y	Y	P	6.5
PS20	N	Y	Y	N	Y	P	N	N	Y	Y	Y	P	7
PS21	N	Y	P	N	Y	Y	P	P	Y	N	N	Y	6.5
PS22	Y	N	N	Y	Y	P	P	N	N	P	N	Y	5.5
PS23	P	P	Y	N	P	Y	P	P	Y	Y	Y	N	7.5
PS24	Y	N	N	Y	Y	P	P	P	N	N	N	Y	5.5
PS25	N	Y	Y	N	P	N	N	N	N	Y	Y	P	5
PS26	Y	P	N	P	Y	P	P	P	Y	N	P	Y	7
PS27	P	P	Y	P	Y	Y	Y	Y	N	N	Y	P	8
PS28	N	N	Y	Y	Y	P	N	N	Y	Y	Y	P	7
PS29	N	P	Y	N	Y	Y	P	P	Y	Y	Y	P	8
PS30	P	Y	P	P	Y	N	N	P	Y	Y	Y	P	7.5
PS31	Y	Y	Y	P	Y	P	N	N	N	Y	Y	P	7.5
PS32	Y	Y	N	P	Y	N	N	N	Y	Y	P	P	6.5
PS33	P	Y	Y	P	Y	Y	P	Y	Y	Y	Y	P	10
PS34	Y	N	P	P	N	P	N	N	Y	Y	Y	P	6
PS35	Y	N	P	P	N	P	N	P	Y	N	Y	P	5.5
PS36	Y	P	P	Y	Y	Y	P	P	N	N	Y	Y	8
Total	22.5	21	23.5	17	24	20	10	12.5	28	24	28.5	19.5	250.5
% total score	**8.98%**	**8.38%**	**9.38%**	**6.79%**	**9.58%**	**7.98%**	**3.99%**	**4.99%**	**11.18%**	**9.58%**	**11.38%**	**7.78%**	**100%**
% by Max AAs	**62.5%**	**58.3%**	**65.3%**	**47.2%**	**66.7%**	**55.6%**	**27.8%**	**34.7%**	**77.8%**	**66.7%**	**79.2%**	**54.2%**	

defined a research question or assessment analysis elements in this direction. This is more of a "subjective" analysis. The families we identified are as follows:

***Studies oriented Model Driven Development (MDD)** and centred on model weaving and model transformation*: this 1st family gathers studies PS7, PS8, PS9, PS10, PS11, PS20, PS21, PS22, PS23, PS25, PS27, PS30, PS31, PS33 and PS35. These studies focus on the composition as a separate operation that has its place and its proper role in an MDE development approach. If these approaches generally produce a merge of composed models by the way of a model transformation, they advocate establishing prior correspondences between elements of the input models. They often talk about model weaving. One part of these approaches proposes to rely on a language of comparison to facilitate or produce this weaving. Another part proposes a language of rules to produce a matching. A last part offers a domain specific language (graphic or syntactic) that helps describing a weaving between elements of the input models.

***Studies inspired by Aspect Oriented Programming (AOP)** and focused on multi aspects composition*: This 2nd family gathers studies PS1, PS2, PS3, PS4, PS5, PS6, PS15, PS19 and PS34. AOP recommends designing a system by separating it into different morsels. Each corresponds to a different aspect. This de-composition permits to deal with properties on each aspect before considering the system in its overall. This is an efficient way to decrease the analysis complexity of systems. However, this requires being able to integrate the morsels of a model with each other's. Thus, the studies aforementioned address this problem from an MDE point of view considering aspects as separate models and providing model composition solutions to gather them later. They often talk about pointcut model, base model and join points. Generally, the first step is the detection step. It uses the pointcut model and the base model to compute a set of join points. Each join point is characterized by a morphism from the pointcut to a corresponding element in the base model. In a second step, using these morphisms, the complete model is composed with each join point in the base model.

***Studies oriented Domain Specific Languages (DSL)** and focused on language composition*: This 3rd family gathers studies PS12, PS13, PS14, PS17, PS18, PS28 and PS29. These studies mainly explore how extending DSLs by composition their domains at an abstract level, using high-level domain concepts. Their goal is to reuse existing components and tools and to support different types of evolution. These approaches focus more on the projection of the composition on the syntactic level. They often use algebra to define the different variants of the composition, their interaction with the semantics of the models involved and their composition operators. They mostly rely on solutions of the first family to drive the projection of the composition on the syntactic level.

***Studies oriented Collaborative Programming (CP)** and motivated by version control*: This 4th family gathers studies PS16, PS24, PS26, and PS36. These studies address the problem of model merge in a collaborative development environment [83]. These approaches provide solutions to merge different versions of a model based on change definitions at a higher level of abstraction. While ensuring to maintain the model remains in a consistent state. The challenge of these approaches is to automate the merge mechanism dealing with issues of prioritization, consistency and conflict resolution. Besides, the composition is then restricted to merge versions of the same model.

A Study oriented Object-Role Modeling (ORM) [84]. Finally, we find a study out of these four families: PS32. It presents a different conceptualization of model composition. It proposes an approach, object-role driven, which establishes relationships between elements from different models, but does not produce composed models later. This alternative approach does not alter the source models. The composition is not defined as an operation but as set of constraints pointing correspondences between elements from different models.

5.2. RQ1. Which kinds of correspondences between models are used in the current state-of-the art in model composition in the context of MDE?

As we have mentioned before, authors of different proposals use different terms like “merging” or “weaving” which correspond to their specific interpretation of the model composition, their area of research (i.e. families of classification identified above) and their motivation. Consequently, depending of the semantic they give to the model composition operation, they define different kinds of correspondences between models to be composed.

However, it seems obvious that the most complete solution to define correspondences between models in order to compose them is to integrate the matching operation as a preliminary operation; by possibly automating it. Therefore, it has to incorporate the use of a correspondence model to represent links between the elements to be composed. Ideally, this model has to be a full-fledged model that conforms to a metamodel and that can be persisted. Afterwards, it has to give the possibility to weave the composition freely and customize the compound model.

Following on from this idea, in order to evaluate the proposals according to this research question, we have analyzed the results of the three AA questions AA1, AA2 and AA3.

- AA1 checks whether the proposal expresses means to carry out match operation;
- AA2 checks if the proposal integrates the use of the correspondence model in the composition process;
- AA3 checks if the proposal gives the possibility to weave the composition freely then to customize the compound model.

We have found that about 60% per cent of the PSs fulfill each of the questions. However, there are only three PSs, namely PS3, PS4 and PS31, which have obtained a full score for the three questions. In addition, eleven of the 36 PSs, namely PS1, PS2, PS5, PS10, PS15, PS18, PS23, PS27, PS30, PS33 and PS36, obtained at least a partial score per question. Accordingly, we can deduce that more than the third of the selected studies attempted to implement a complete solution of model composition.

With regard to results, we notice that the PSs, which recognize the matching operation, are mainly the proposals inspired by AOP and those oriented CP. This result can be explained by two facts. First, finding or defining points of correspondence between different aspects of a system is at the core of the AOP. Secondly, CP is interested by revision control and it seems obvious that merging two different versions of the same model relies first in establishing the matching between them.

Likewise, proposals inspired by AOP and those oriented MDD often define the composition operation as a weaving operation. However, only MDD oriented proposals are typically use a correspondence model to persist dependences between source models. This can be explained by the fact that MDD is interested by issues such as traceability and interoperability where this persistence makes is necessary.

Concerning DSL oriented approaches; they consider composition as a mean for reuse or extension. The composition is often between different domains that must be associated within the same DSL. As a result, matching is difficult to implement. Thus, they consider less the matching operation, but give more importance to the weaving of models, which allows designing a new DSL by weaving existing ones. Keeping a traceability of the elements that led to the composition is not of fundamental importance for these approaches. Therefore, the use and persistence of a correspondence model is not much considered by them.

5.3. RQ2. What level of automation is suggested for the creation and the management of links between models during the composition operation?

As MDE focuses on automating operations around models, it is clear that the ability to provide means for automating the creation and the management of links between models during the composition operation is an important asset for a model composition framework. Thus, we were interested to know how the different proposals deal with the automation of the model composition process.

In order to evaluate the proposals according to this research question, we have analyzed the results of the two AA questions AA4 and AA5.

- AA4 checks whether the proposal of the PS provides the means for minimizing the effort made to compose models elements;
- AA5 checks whether the proposal of the PS supports to go across elements of the composed model to their source elements in the original models.

We have found that about 65% per cent of the PSs fulfill AA4 and about 47% per cent of them fulfill AA5. However, there are only four PSs, namely PS22, PS24, PS28 and PS36, which have obtained a full score for the tow questions. In addition, twelve of the 36 PSs, namely PS2, PS5, PS7, PS10, PS15, PS18, PS26, PS27, PS30, PS31, PS32, and PS33, obtained at least a partial score per question. Accordingly, we can deduce that the half of the PSs was concerned with implementing an automation of some tasks related to the model composition process.

Concerning results, we noticed that no study fulfills completely AA4 except the four that fulfill the two questions AA4 and AA5. Of these four, two are CP oriented, one is MDD oriented and one is DSL oriented. Most of the other 12 studies that fulfill AA5 are oriented MDD or inspired by AOP.

These results are explained by the fact that it is very difficult to fully automate the matching search phase between models to be composed except in the case of CP where the goal is to merge two versions of the same model. Indeed, finding automatic matching in this case is feasible and makes sense. Contrarily, for MDD oriented or AOP inspired approaches, models to be composed are often heterogeneous. Even if the automation of matching search between models can be justified, the PSs mainly propose an assisted help with means to implement matching rules or heuristics. This explains why most of the approaches that have tried to answer AA4 have received a partial score (i.e. the proposal provides techniques to realize matching either semi-automatically).

Likewise, MDD oriented and AOP inspired approaches have a clear need to provide the means to navigate between composed models. As aforementioned, these approaches are interested in issues such as traceability and interoperability. For DSL oriented studies, this mean is interesting only if the language composition does not produce a new language but provides capabilities for referencing a DSL from another one. However, it concerns few approaches. For CP oriented studies, once the two models are merged the approaches are rarely interested in keeping traceability between the merged elements. This was surprising because keeping a history seems also important in this kind of issue. We estimate that it is a point of weakness for these proposals.

5.4. RQ3. What value can be drawn from the literature with regard to the use of model composition in the context of MDE?

As previously stated, from a theoretical point of view MDE furnishes a new landscape in which to better address larger and more complex systems contexts in software development. Hence, we were interested through our SLR to know whether PSs have provided any implementation, framework or technological support that has been put into practice and resulted in significant use cases.

In order to evaluate the proposals according to this research question, we have analyzed the results of the three AA questions AA6, AA7 and AA8.

- AA6 checks whether the proposal of the PS has been implemented;
- AA7 checks whether the proposal of the PS has been adopted by a wide software community or supported by a software editor;
- AA8 checks whether the proposal of the PS has been the subject of significant use cases.

Unfortunately, the traditional lag between theoretical proposals and practical implementation that often appears by many SE proposals is present in our case. The scores for AAs questions AA6, AA7 and AA8 presented in Table 7 allow us to confirm this statement.

We found that about 55% percent of PSs fulfill AA6. This shows that about a little over half (i.e. 20) of PSs deliver at least a partial implementation of their approach. However, this indicator is slightly misleading because it did not relate the practical deployment and usefulness of the implementation. Hence, there is the importance of questions AA7 and AA8. For them, the scores are significantly lower; around 27% per cent for AA7 and 35% per cent for AA8. There are only three PSs, namely PS1, PS7 and PS27, which obtained a full score for the question AA7 and only five PSs, namely PS7, PS13, PS15, PS27 and PS33, which obtained a full score for the question AA8. In addition, as can be seen, only PS7 and PS27 obtain a full score for the two questions. Besides, fourteen of the 36 PSs, namely PS5, PS12, PS14, PS15, PS16, PS17, PS21, PS22, PS23, PS24, PS26, PS29, PS33 and PS36, obtained at least a partial score for question AA7. Where, fifteen of the 36 PSs, namely PS1, PS2, PS3, PS10, PS12, PS16, PS17, PS21, PS23, PS24, PS26, PS29, PS29, PS30, PS35 and PS36, obtained at least a partial score for question AA8.

With regard to results we notice that the PSs which obtained a full or partial score are mainly the proposals oriented MDD or CP. Approaches inspired by AOP obtained low scores, as for those oriented DSL, they rarely provide an implementation and support tooling for an industrial scale. In addition, most of the PSs implementations have adopted the MOF[1] specification and have been developed using the Eclipse IDE and technologies furnished by the Eclipse Foundation projects. More precisely in the context of the Eclipse Modelling Project (EMP)[2] that provides a large set of tools and frameworks for standards that help with the development of technological support for proposals based on MDE. Besides, it is important to note that all the PSs that implement their approach have provided an open source implementation.

Therefore, two main conclusions can be drown from this research question RQ3. Firstly, it is difficult to provide an industrial scale technical support for model composition management in MDE proposals, especially for AOP use cases, and what is more, for DSL use cases. Secondly, existing proposals have commonly adopted EMF-based[3] technologies as standard de-facto for the implementation of their approaches.

5.5. RQ4. What are the limitations of state of the art in model composition in the context of MDE?

Analyzing the PSs proposals, we have noted the shortcomings and limitations of each one in regards of certain criteria. It should be noted that sometimes these limitations might be justified by the purpose of the approach. Anyhow, we have founded interesting to evaluate the limitations of our PSs on the following criteria: the number of models that can be composed, the nature of models (heterogeneous or homogeneous), the possibility of configuring the composition schema and finally the possibility of supporting the life cycle of composed models (future extension as backwards compatibility).

In order to evaluate the proposals according to this research question, we have analyzed the results of the four AA questions AA9, AA10, AA11 and AA12.

- AA9 checks whether the proposal of the PS allows composing an unlimited number of input models;
- AA10 checks whether the proposal of the PS allows composing heterogeneous models;
- AA11 checks whether the proposal of the PS enables the design of the composition metamodel;
- AA12 checks whether the proposal of the PS supports as well future extension as backwards compatibility of composed models.

Number of input models: Most of the proposals allow more than two models as inputs of the model composition. Only eight of them, namely PS4, PS17, PS22, PS24, PS25, PS27, PS31 and PS36, do not allow it. It is in the MDD oriented proposals that half of these are found, namely PS22, PS25, PS27 and PS31. Indeed, proposals that provide a composition solution by a weaving between the metamodels of the models to be composed are often limited to two input models. Likewise, this score seems logical for the proposals inspired by AOP (all PSs except PS4) and those oriented DSL (all PSs except PS17). For the former, it comes from the fact that these approaches propose the composition of multiple points of view of a system, each of which represented by a model. For the second, coming from the fact that the extension of a DSL often refers to elements coming from multiple grammars of other DSLs. Nevertheless, the result is surprising for CP oriented approaches. It would be trivial to believe that they provide a merge of two versions of a given model. However, two of the four CP oriented PSs, namely PS16 and PS26, gave the possibility to merge more than two models.

Heterogeneous models: More than two-thirds of PSs allow composing heterogeneous models. Exactly, twenty of them allow composing models conforming to different metamodels and four of them allow composing models conforming to different versions of the same metamodel. Only ten of the PSs, namely PS1, PS2, PS4, PS6, PS21, PS24, PS26, PS27, PS35 and PS36, do not allow composing heterogeneous models. It is in the proposals inspired by AOP and oriented CP that the proportion is significant. For the former a half of proposals did not allow heterogeneous composition. These proposals propose composition for multiple points of view of a system, which is often defined through one metamodel. For the second, no proposal allowed composing completely heterogeneous models. Effectively, it is difficult to perform, and hard to explain, a versioning of models conforming to completely different metamodels. Likewise, this score seems logical for the proposals oriented DSL where all PSs allowed composing heterogeneous models. Given that the composition between two languages, that have the same grammar, seems of no interest. For the MDD oriented proposals, 3 proposals, namely PS21, PS27 and PS35, did not support composing heterogeneous models and two proposals, namely PS8 and PS22 support only composing models conforming to different versions of the same metamodel. In their cases, we estimate that it is a point of weakness.

Design of the composition metamodel: Most of the proposals enable the design of the composition metamodel. Exactly, 26 of them gives means to design the schema of composition (i.e. the metamodel of the correspondence model) and five of them furnish a unique schema of composition but it is possible to extend it through external tools. Only five proposals, namely PS8, PS16, PS21, PS22 and PS24, provide a unique and rigid schema of composition. It is in the MDD and CP oriented proposals where we found all of them (i.e. 3 in MDD oriented proposals and 2 in CP oriented proposals). If this score is not surprising for CP oriented proposals because they have to provide a merge of two versions of a given model conforming to the same metamodel. Thus, there is no need to tune the metamodel of correspondences. Regarding the MDD oriented proposals; we estimate that it is clearly a point of weakness.

Future extension and backwards compatibility: The score for this limitation is mixed because a large part of the proposals (i.e. 21 proposals) partially supports the management of future evolutions or backwards

[1] Meta-Object Facility website: http://omg.org/mof/.
[2] Eclipse Modeling Project website: http://eclipse.org/modeling/.
[3] Eclipse Modeling Framework website: http://eclipse.org/modeling/emf/.

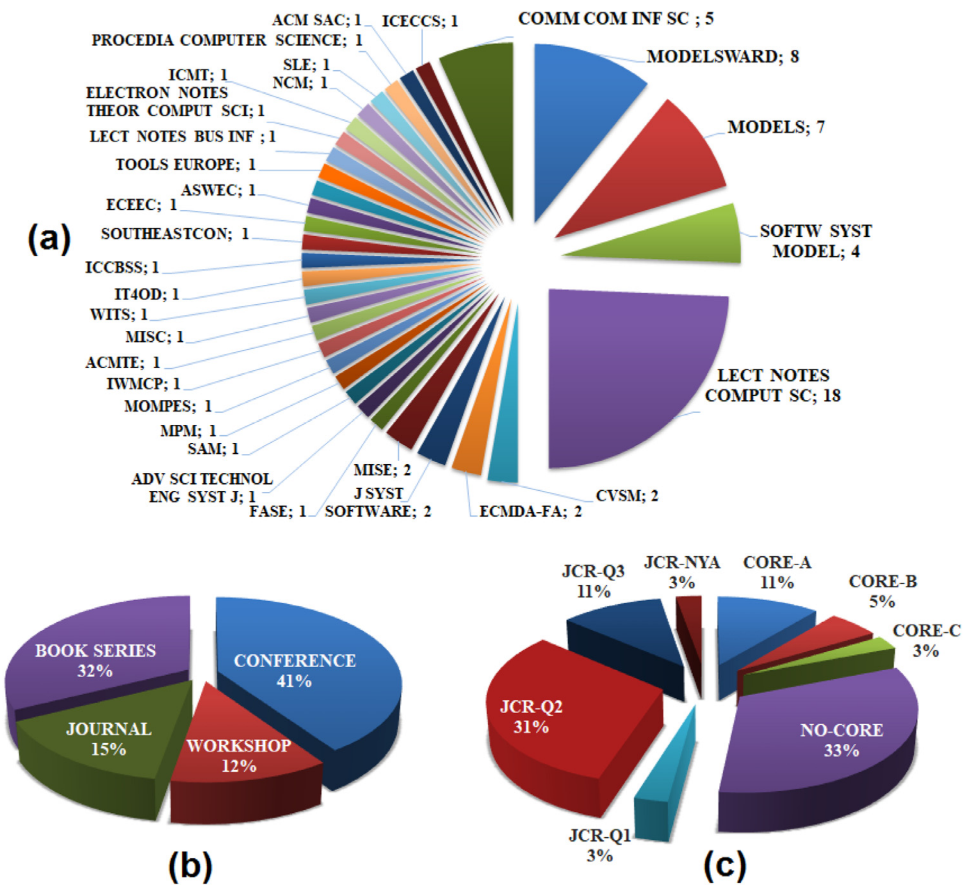

Fig. 10. The distribution of PSs and ISs publications on forums.

compatibility. This means that they provide a way to update the composition links, for example, and to maintain the composed model. Besides, nine proposals fully support this feature. Actually, all the CP oriented proposals provide this functionality, which is logical in view of their core motivation, i. e. the versioning of a model. Moreover, four MDD oriented proposals, namely PS9, PS10, PS21 and PS22 and a single proposal oriented DSL, namely PS12, support this feature. However, no proposal inspired by AOP provides this capability. We estimate that it is clearly a point of weakness. Because in a framework dedicated to multi view integration of a complex system, this capability would be of great use.

5.6. RQ5. Are there favored journals or conferences where model composition in MDE proposals are published, and what is the quality of these forums?

To answer this question, we decided to take into account the publications of the PSs as well as the publications of their ISs. In addition, if a PS is published in a journal following a previous communication at a conference, both publications are considered. Fig. 10 presents three pie charts. The first one (a) presents the distribution of all the publications across the different forums. The second one (b) presents the distribution of all the publications regarding the forums types. Finally, the third one (c) presents the distribution of all the publication by ranking them according to the importance and quality of the forums in which they were published; we relied on the CORE Conference Ranking for conference papers and the SJR Indicator for journal and book papers.

We observed in (a) that more than half of the publications, or exactly 57% per cent, are distributed over five forums. The LECT NOTES COMPUT SC book series alone represents a quarter of all publications. Then, four forums share the second quarter with two conferences, namely MODELS and MODELSWARD, and two journals, namely COMM COM INF SC and SOFTW SYST MODEL. The distribution of the rest of the publications is widely dispersed, with one or two publications per forum.

Besides, we note in (b) that about half of the publications, or exactly 53% per cent, were published in conferences and workshops and the other half, or exactly 47% per cent, were published in book series and journals. However, noting that a large majority of publications in book series follow a prior publication in a conference. It seems legitimate to submit a proposal on recent topics to a conference rather than directly to a journal. This allows to receive feedback and enabling to submit a more complete and more mature study to a journal. In addition, this can allows to possibly be invited by a journal. The low number of publications in journals can therefore be explained by the difficulty in achieving a sufficient level of maturity without going through conferences. Otherwise, a simple reason can be the fact that the model composition in MDE is still an emerging area. Indeed, out of the 13 publications published since 2016, only three ones were published as journal papers. Moreover, except one, they were preceded by a conference publication.

As aforementioned, publications are divided between a first half on conferences and workshops and a second half on journals and book series. Regarding (c), we noted that about 60% per cent of publications on conferences and workshops were not ranked in CORE, where about 20% per cent of them were ranked in CORE-A and 20% per cent were ranked in CORE-B or CORE-C. This result confirms that model composition in MDE is still an interesting area of research. This is reflected by the fact that this field is the subject of communications in conferences and workshops of all ranks. The largest ratio of unranked CORE conferences can be explained by the necessity to go through small conferences before exposing more accomplished studies in renowned ones. Regarding the journals and book series publications, except two publications that have no assigned quartile yet (JCR-NYA), the half of them were from the second quartile (JCR-Q2) and the second half were from the first quartile (JCR-Q1) or the third quartile (JCR-Q3). This result demonstrates the high scientific interest that proposals of model composition in MDE reached when they attempt a good level of maturity.

6. Limitations

As explained earlier, this study has been carried out by following Biolchini et al. [20], Brereton et al. [22], and Kitchenham and Charters [23] guidelines for conducting an SLR in SE. However, we have followed a process that deviated lightly from them. Here are the points of deviation:

- We limited our researches to four search engines of four libraries. As mentioned in Section 3.2, we considered characteristics such as ease of use, accessibility of documents, spectrum of indexed documents, ability of their search engines to take a complex textual query, accuracy of search and reproducibility of results. However, we have not execute researches in some rich search engines like Scopus and Google Scholar. For Scopus, it was difficult to create complex queries through its search engine even using the Scopus search API. We should have separated to split the query into multiple queries, each containing the term “modeldȁ and at least one of the synonyms of the terms “composition” (i.e., “weaving”, “merging”, etc.). Then we should have downloaded the results in several parts before merging them manually and removing the internal duplicates obtained by our process. Which could alter the systematic aspect of our approach. In addition, we should have used additional software to extract results (e.g. Harzing’s Publish or Perish[4]). Because the number of results returned was very important (over 50,000). However, we did not find the way to use them from the Scopus search engine interface and the Scopus engine did not furnish any mean to export results in a digital form to be able to process them. Besides, the Google Scholar search engine has the same problems when creating queries. However, it allows you to use software to extract results such as Harzing’s Publish or Perish. Only, the search engine limits the number of results per request to ca. 1.000. Therefore, we should have to download the results in several parts using for example the limits of the year of publication. However, above all, the Google Scholar search poses certain problems for the structured retrieval of the literature [85] (for example, the non-commutativity of logical disjunctions). .
- Our searches consisted mostly of searches executed in digital libraries on the web. We did not really execute manual searches as advised by the guidelines. Except studying the bibliographies of the selected primary studies in order to identify more relevant studies. After which we have not identified any other relevant study different from those already recorded.
- We did not solicit external experts to review our SLR. Despite of this fact, during the development of this study, we have often exchanged roles between authors to provide a kind of internal evaluation. In fact, some authors sometimes played the role of referees when the others led the research and vice versa. Even so, we cannot exclude that some mistakes have been made in the extracted data or in their analysis.
- We did not employ the PICOC (Population, Intervention, Comparison, Outcome, and Context) criteria to establish the RQs. Despite the fact that we did not employ the PICCO, the RQs can cover some of these criteria. For example, the context of our RQs is “MDE”, the population is “Model Composition” and the interventions are “MDD”, “AOP”, “DSL”, “CP” and “ORM”. In spite of this deviation that can imply subjective or unstructured RQs, we would like to note that the definition of our RQs was the result of an iterative process, which had repeatedly held to refine and improve them.

[4] Publish or Perish: https://harzing.com/resources/publish-or-perish.

7. Conclusions

Given the importance of composing software units in SE and the promise of MDE to offer new possibilities to improve this operation. We have conducted a systematic literature review on the field of model composition in an MDE context, based on the guidelines proposed by Biolchini et al. [20], Brereton et al. [22], and Kitchenham and Charters [23].

In summary, figures of this systematic literature review can be as follow: the execution of searches in each chosen digital library, using adequate query strings, reached 9270 outputs, from which just 56 were found to be relevant studies. After eliminating duplicates and interrelated studies, the 36 retained primary studies were assessed according to quality assessments and analysis assessments questions. This led to identify and analyse the state of the art in model composition in the context of MDE in regards to five research questions.

Conducting this SLR has allowed us to collect some data and establish some appreciations which have been exposed in the discussion section. The following is a summary of key findings and conclusions:

- Regarding quality assessments, the definition of the followed research methodology in the evaluated primary studies is an important handicap. Only four primary studies had clearly specified the research methodology. The other qualitative aspect to be improved is that of the specification of limitations of the research, which has been explicitly specified only by the quarter of the primary studies. The other evaluated quality aspects such as the definition of the research goal, the specification of the interest and the usefulness, or the comparison to related works, obtained nice scores.
- Primary studies can be classified in four major families: works oriented *Model Driven Development*, works inspired by *Aspect Oriented Programming*, works oriented *Domain Specific Language or Domain Specific Modeling Language* and works oriented *Collaborative Programming*. It worth to notice that we find a study out of these four families that proposes an *Object-Role Driven* approach.
- The match as a preliminary operation before the model composition is not a widely shared concept. Only about the third of primary studies attempted to implement it. We notice that the studies, which recognize the matching operation, are mainly the proposals inspired by *Aspect Oriented Programming* and those oriented *Collaborative Programming*. In addition, it is very difficult to automate completely the matching phase between models to be composed except in the case of *Collaborative Programming*.
- *Model Driven Development* and *Aspect Oriented Programming* approaches have a clear need to provide automatic support to navigate between composed elements keeping traceability between them. Regarding *Domain Specific Language or Domain Specific Modeling Language* oriented approaches, this mean is interesting only if the language composition does not produce a new language but provides capabilities for referencing a language from another one. By contrast, for *Collaborative Programming* oriented approaches, once the models are merged, the approaches are rarely interested in the navigability between composed elements and source elements. However, they provide a means of traceability that keeps the history between these elements.
- Few approaches provide industrial scaled frameworks for model composition management in MDE. A lack of maturity exits especially for approaches inspired by *Aspect Oriented Programming* and those oriented *Domain Specific Language or Domain Specific Modeling Language*.
- Primary studies proposals have commonly adopted EMF-based technologies as standard de-facto for the implementation of their approaches.
- Most of the primary studies allow more than two models as inputs of the model composition, allow composing heterogeneous models and enable the design of the composition schema. The principal limita-

tion of state of the art in model composition in the context of MDE is the support of the management of future evolutions or backwards compatibility.

This review has additionally helped to point out some of the main area of improvement that model composition approaches in MDE should tackle in the future. Some of these could be:

- As aforementioned, the definition of the followed research methodology in the evaluated primary studies was an important handicap. Future works needs to address this aspect better in order to produce rigorous studies and is of high quality. This will also facilitate their dissemination in the most impactful scientific forums.
- Most proposals have commonly adopted EMF-based technologies as standard de-facto for the implementation of their approaches. Thus, they are technology-dependent. It would therefore be appropriate to provide a technology-agnostic means of model composition in order to permit their application to any existing model language.
- Although most of the reviewed proposals have provided an implementation, several limitations have been observed. First, few approaches provided industrial scaled frameworks. Secondly, with regard to technical assistance and the graphical interface, they are mainly ad hoc and do not meet the criteria of modern ergonomic platforms. Finally, many of the approaches do not support automatic backward compatibility and future evolution. Modeling approaches need to take into account these issues to provide more mature software frameworks, which will allow for better deployment in the industry.

To conclude, we would like to underline that analysing the forums and dates of publications we believe that model composition in MDE is still an emerging and interesting area of research. It is time to address the different highlighted challenges that will enable us to make the most of model composition.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgment

We express our respected gratitude to the LabSIV Laboratory of Ibn Zohr University and to the Aeronautics, Space and Defense Division of Altran South West France for funding this research.

References

[1] E.W. Dijkstra, A Discipline of Programming, Prentice Hall PTR, 1976 ISBN: 978-0132158718.
[2] D.L. Parnas, On the criteria to be used in decomposing systems into modules, Commun. ACM. 15 (12) (1972) 1053–1058 https://doi.org/10.1145/361598.361623.
[3] K.M. Chandy, S. Taylor, A Primer for Program Composition Notation, California Institute of Technology, 1990 Technical Report https://core.ac.uk/download/pdf/4891542.pdf.
[4] M. Vargas-Vera, “Using prolog techniques to guide program composition,” PhD. Thesis, Edinburgh University, 1995. https://www.dai.ed.ac.uk/papers/documents/pt9508.html.
[5] A. Martín, L. Lamport, Composing specifications, ACM. T. Progr. Lang. Sys. 15 (1) (1993) 73–132 https://doi.org/10.1145/151646.151649.
[6] G. Kiczales, Aspect-oriented programming, 27th International Conference on Software Engineering, 2005 https://doi.org/10.1109/ICSE.2005.1553685.
[7] K. Czarnecki, U.W. Eisenecker, Generative Programming: Methods, Tools, and Applications, ACM Press/Addison-Wesley Publ. Co., 2000 ISBN: 978-0201309775.
[8] J. Bézivin, In search of a basic principle for model driven engineering, Novatica 5 (2) (2004) 21–24 https://hal.archives-ouvertes.fr/hal-00442702.
[9] D.C. Schmidt, Guest Editor’s Introduction: model-driven engineering, Computer. 39 (2) (2006) 25–31 https://doi.org/10.1109/MC.2006.58.
[10] M.Voelter Stahl, K. Czarnecki, Model-Driven Software Development: Technology, Engineering, Management, Wiley & Sons, Inc., 2006 ISBN: 978-0470025703.
[11] K. Czarnecki, S. Helsen, Feature-based survey of model transformation approaches, IBM. Syst. J. 45 (3) (2006) 621–645 https://doi.org/10.1147/sj.453.0621.
[12] P.A. Bernstein, A.Y Halevy, R.A. Pottinger, A vision for management of complex models, Sigmod. Rec. 29 (4) (2000) 55–63 https://doi.org/10.1145/369275.369289.
[13] R.A. Pottinger, P.A. Bernstein, Merging models based on given correspondences, 29th International Conference on Very Large Data Bases, 2003 https://doi.org/10.1016/B978-012722442-8/50081-1.
[14] J. Kienzle, J. Gray, D. Stein, Report of the 7th international workshop on aspect-oriented modeling, Lect. Notes. Comput. Sc. 3844 (2006) 91–99 https://doi.org/10.1007/11663430_10.
[15] J. Estublier, G. Vega, A.D. Ionita, Composing domain-specific languages for widescope software engineering applications, Lect. Notes. Comput. Sc. 3713 (2005) 69–83 https://doi.org/10.1007/11557432_6.
[16] F. Fleurey, B. Baudry, R. France, S. Ghosh, A generic approach for automatic model composition, Lect. Notes. Comput. Sc. 5002 (2007) 7–15 https://doi.org/10.1007/978-3-540-69073-3_2.
[17] J. Bézivin, R. F. Paige, U. Aßmann, B. Rumpe, D. Schmidt, "Manifesto-model engineering for complex systems" arXiv preprint, 1409.6591, 2014. https://arxiv.org/abs/1409.6591
[18] D.S. Kolovos, L.M. Rose, N. Matragkas, R.F. Paige, E. Guerra, J.S. Cuadrado, J. De Lara, I. R., Dániel Varró, M. Tisi, J. Cabot, A research roadmap towards achieving scalability in model driven engineering, 1st International Workshop on Scalability in Model Driven Engineering (BigMDE ’13) co-located with the 7th International Conference on Software Technologies - Applications and Foundations Conference (STAF’13), 2013 https://doi.org/10.1145/2487766.2487768.
[19] G. Mussbacher, D. Amyot, R. Breu, J.M. Bruel, B.H.C. Cheng, P. Collet, B. Combemale, R.B. France, R. Heldal, J. Hill, J. Kienzle, M. Schöttle, F. Steimann, D. Stikkolorum, J. Whittle, The relevance of model-driven engineering thirty years from now, Lect. Notes. Comput. Sc. 8767 (2014) 183–200 https://doi.org/10.1007/978-3-319-11653-2_12.
[20] J. Biolchini, P. G. Mian, A. C. C. Natali, G. H. Travassos, "Systematic review in software engineering," Technical Report ES,**679**(05), 45, System Engineering and Computer Science Department COPPE/UFRJ, 2005. https://www.cin.ufpe.br/~in1037/leitura/systematicReviewSE-COPPE.pdf
[21] B. Kitchenham, P. Brereton, D. Budgen, M. Turner, J. Bailey, S. Linkman, Systematic literature reviews in software engineering–a systematic literature review, Inform. Software. Tech. 51 (1) (2009) 7–15 https://doi.org/10.1016/j.infsof.2008.09.009.
[22] P. Brereton, B.A. Kitchenham, D. Budgen, M. Turner, M. Khalil, Lessons from applying the systematic literature review process within the software engineering domain, J. Syst. Software. 80 (4) (2007) 571–583 https://doi.org/10.1016/j.jss.2006.07.009.
[23] B. Kitchenham, S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Technical Report EBSE, Keele University and Durham University Joint Report, 2007. https://citeseerx.ist.psu.edu/viewdoc/doi=10.1.1.117.471.
[24] I. Santiago, A. Jiménez, J.M. Vara, V. De Castro, V.A. Bollati, E Marcos, Model-driven engineering as a new landscape for traceability management: a systematic literature review, Inform. Software. Tech. 54 (12) (2012) 1340–1356 https://doi.org/10.1016/j.infsof.2012.07.008.
[25] T. Cottenier, A. Van Den Berg, T. Elrad, Modeling aspect-oriented compositions, Lect. Notes. Comput. Sc. 3844 (2006) 100–109 https://doi.org/10.1007/11663430_11g.
[26] C. Batini, M. Lenzerini, S.B. Navathe, A comparative analysis of methodologies for database schema integration, Comput. Surv. 18 (4) (1986) 323–364 https://doi.org/10.1145/27633.27634.
[27] K. Czarnecki, S. Helsen, Feature-based survey of model transformation approaches, IBM. Syst. J. 45 (3) (2006) 621–645 https://doi.org/10.1147/sj.453.0621.
[28] Object Management Group terms and acronyms, Object Management Group, Inc., last accessed August 2019. https://omg.org/gettingstarted/terms_and_acronyms.htm.
[29] G. Mussbacher, O. Alam, M. Alhaj, S. Ali, N. Amálio, B. Barn, R. Bræk, T. Clark, B. Combemale, L.M. Cysneiros, U. Fatima, R. France, G. Georg, J. Horkoff, J. Kienzle, J.C. Leite, T.C. Lethbridge, M. Luckey, A. Moreira, F. Mutz, A.P.A. Oliveira, D.C. Petriu, M. Schöttle, L. Troup, V.M.B. Werneck, Assessing composition in modeling approaches, 2nd International Workshop on Comparing Modeling Approaches co-located with the 15th International Conference on Model Driven Engineering Languages and Systems (CMA@MoDELS 2012), 2012 https://doi.org/10.1145/2459031.2459032.
[30] J. Bézivin, S. Bouzitouna, M.D. Del Fabro, M.P. Gervais, F. Jouault, D. Kolovos, I. Kurtev, R.F. Paig, A canonical scheme for model composition, Lect. Notes. Comput. Sc. 4066 (2006) 346–360 https://doi.org/10.1007/11787044_26.
[31] Modelware IST project, science | business network, last accessed August 2019. https://sciencebusiness.net/news/73237/IST-sponsored-project-offers-productivity-tools-for-software-developers
[32] J.M. Jézéquel, Model driven design and aspect weaving, Softw. Syst. Model. 7 (2) (2008) 209–218 https://doi.org/10.1007/s10270-008-0080-5.
[33] L. Fuentes, P. Sánchez, Dynamic weaving of aspect-oriented executable UML models, Lect. Notes. Comput. Sc. 5560 (2009) 1–38 https://doi.org/10.1007/978-3-642-03764-1_1.
[34] P. Sánchez, L. Fuentes, D. Stein, S. Hanenberg, R. Unland, Aspect-oriented model weaving beyond model composition and model transformation, Lect. Notes. Comput. Sc. 5301 (2008) 766–781 https://doi.org/10.1007/978-3-540-87875-9_53.
[35] M.E. Kramer, J. Klein, J.R. Steel, B. Morin, J. Kienzle, O. Barais, J.M. Jézéquel, Achieving practical genericity in model weaving through extensibility, Lect. Notes. Comput. Sc. 7909 (2013) 108–124 https://doi.org/10.1007/978-3-642-38883-5_12.
[36] O. Barais, J. Klein, B. Baudry, A. Jackson, S. Clarke, Composing multi-view aspect models, 7th International Conference on Composition-Based Software Systems (ICCBSS 2008), 2008 https://doi.org/10.1109/ICCBSS.2008.12.
[37] J.Y. Marchand, B. Combemale, B. Baudry, "A categorical model of model merging and weaving, 4th International Workshop on Modeling in Software Engineering (MiSE ’12), 2012 https://hal.inria.fr/hal-00714373/document.

[38] D.S. Kolovos, R.F. Paige, F.A.C. Polack, Merging models with the epsilon merging language (EML), Lect. Notes. Comput. Sc. 4199 (2006) 215–229 https://doi.org/10.1007/11880240_16.

[39] M. Boubakir, A. Chaoui, A pairwise approach for model merging, Lect. Notes Netwo. Syst. 1 (2016) 327–340 https://doi.org/10.1007/978-3-319-33410-3_23.

[40] H. Berg, B. Møller-Pedersen, metamodel and model composition by integration of operational semantics, Comm. Com. Inf. Sc. 580 (2015) 172–189 https://doi.org/10.1007/978-3-319-27869-8_10.

[41] H. Berg, B. Møller-Pedersen, Towards non-intrusive composition of executable models, 3rd International Conference on Model-Driven Engineering and Software Development (MODELSWARD), 2015 https://doi.org/10.5220/0005242401110121.

[42] H. Berg, B. Møller-Pedersen, Type-safe symmetric composition of metamodels using templates, Lect. Notes. Comput. Sc. 7744 (2012) 160–178 https://doi.org/10.1007/978-3-642-36757-1_10.

[43] I.C. Yu, H. Berg, A framework for metamodel composition and adaptation with conformance-preserving model migration, Comm. Com. Inf. Sc. 580 (2015) 133–154 https://doi.org/10.1007/978-3-319-27869-8_8.

[44] A. Anwar, A. Benelallam, M. Nassar, B. Coulette, A graphical specification of model composition with triple graph grammars, Lect. Notes. Comput. Sc. 7706 (2013) 1–18 https://doi.org/10.1007/978-3-642-38209-3_1.

[45] A. Anwar, T. Dkaki, S. Ebersold, B. Coulette, M. Nassar, A formal approach to model composition applied to VUML, 16th IEEE International Conference on Engineering of Complex Computer Systems, 2011 https://doi.org/10.1109/ICECCS.2011.26.

[46] B. Westfechtel, Merging of EMF models, Softw. Syst. Model. 13 (2) (2014) 757–788 https://doi.org/10.1007/s10270-012-0279-3.

[47] F. Schwägerl, S. Uhrig, B. Westfechtel, Model-based tool support for consistent three-way merging of EMF models, International Workshop on ACadeMics Tooling with Eclipse (ACME '13), 2013 https://doi.org/10.1145/2491279.2491281.

[48] F. Schwägerl, S. Uhrig, B. Westfechtel, A graph-based algorithm for three-way merging of ordered collections in EMF models, Sci. Comput. Program. 113 (P1) (2015) 51–81 http://doi.org/10.1016/j.scico.2015.02.008.

[49] B. Westfechtel, A formal approach to three-way merging of EMF models, 1st International Workshop on Model Comparison in Practice (IWMCP '10), 2010 http://doi.org/10.1145/1826147.1826155.

[50] J. Estublier, A.D. Ionita, Extending UML for model composition, 2005 Australian Software Engineering Conference, 2005 Brisbane Queensland Australia http://doi.org/10.1109/ASWEC.2005.25 .

[51] A. Haber, M. Look, P.M.S. Nazari, A.N. Perez, B. Rumpe, S. Völkel, A. Wortmann, Composition of heterogeneous modeling languages, Comm. Com. Inf. Sc. 580 (2015) 45–66 https://doi.org/10.1007/978-3-319-27869-8_3.

[52] A. Haber, M. Look, A.N. Perez, P.M.S. Nazari, B. Rumpe, S. Völkel, A. Wortmann, Integration of heterogeneous modeling languages via extensible and composable language components, 3rd International Conference on Model-Driven Engineering and Software Development (MODELSWARD), 2015 http://doi.org/10.5220/0005225000190031.

[53] B. Meyers, A. Cicchetti, E. Guerra, J. de Lara, Composing textual modelling languages in practice", 6th International Workshop on Multi-Paradigm Modeling, 2012 https://doi.org/10.1145/2508443.2508449.

[54] E. Guerra, J. de Lara, F. Orejas, Inter-modelling with patterns, Softw. Syst. Model. 12 (1) (2013) 145–174 https://doi.org/10.1007/s10270-011-0192-1.

[55] J. Whittle, P. Jayaraman, A. Elkhodary, A. Moreira, J. Araúj, MATA: a unified approach for composing UML aspect models based on graph transformation, Lect. Notes. Comput. Sc. 5560 (2009) 191–237 https://doi.org/10.1007/978-3-642-03764-1_6.

[56] U. Mansoor, M. Kessentini, P. Langer, M. Wimmer, S. Bechikh, K. Deb, MOMM: multi-objective model merging, J. Syst. Software. 103 (2015) 423–439 https://doi.org/10.1016/j.jss.2014.11.043.

[57] C. Herrmann, H. Krahn, B. Rumpe, M. Schindler, S. Völkel, An algebraic view on the semantics of model composition, Lect. Notes. Comput. Sc. 4530 (2007) 99–113 https://doi.org/10.1007/978-3-540-72901-3_8.

[58] A. Yie, R. Casallas, D. Deridder, D. Wagelaar, Deriving correspondence relationships to guide a multi-view heterogeneous composition, Lect. Notes. Comput. Sc. 6002 (2010) 225–239 https://doi.org/10.1007/978-3-642-12261-3_2.

[59] A. Hovsepyan, S. Van Baelen, Y. Berbers, W. Joosen, Specifying and composing concerns expressed in domain-specific modeling languages, Lect. Notes. Bus. Inf. 33 (2009) 116–135 https://doi.org/10.1007/978-3-642-02571-6_8.

[60] I. Weisemöller, A. Schürr, Formal definition of MOF 2.0 metamodel components and composition, Lect. Notes. Comput. Sc. 5301 (2008) 386–400 https://doi.org/10.1007/978-3-540-87875-9_28.

[61] D. Strüber, G. Taentzer, S. Jurack, T. Schäfer, Towards a distributed modeling process based on composite models, Lect. Notes. Comput. Sc. 7793 (2013) 6–20 https://doi.org/10.1007/978-3-642-37057-1_2.

[62] S. Jurack, G. Taentzer, Towards composite model transformations using distributed graph transformation concepts, Lect. Notes. Comput. Sc. 5795 (2009) 226–240 https://doi.org/10.1007/978-3-642-04425-0_1.

[63] C. Bartelt, Consistence preserving model merge in collaborative development processes, 2008 International Workshop on Comparison and Versioning of Software Models, 2008 https://doi.org/10.1145/1370152.137015.

[64] M. Sabetzadeh, S. Nejati, S. Easterbrook, M. Chechik, A relationship-driven framework for model merging, International Workshop on Modeling in Software Engineering (MISE'07: ICSE Workshop 2007), 2007 https://doi.org/10.1109/MISE.2007.4.

[65] M. Schmidt, S. Wenzel, T. Kehrer, U. Kelter, History-based merging of models, 2009 ICSE Workshop on Comparison and Versioning of Software Models (CVSM '09), 2009 https://doi.org/10.1109/CVSM.2009.5071716.

[66] D. Zhang, S. Li, X. Liu, An approach for model composition and verification, 1th IEEE Computer Society International Joint Conference on INC, IMS and IDC, 2009 https://doi.org/10.1109/NCM.2009.271.

[67] H.K. Dam, A. Egyed, M. Winikoff, A. Reder, R.E. Lopez-Herrejon, Consistent merging of model versions, J. Syst. Software 112 (C) (2016) 137–155 https://doi.org/10.1016/j.jss.2015.06.044.

[68] A. Boronat, J. Á. Carsí, I. Ramos, P. Letelier, Formal model merging applied to class diagram integration, Electron. Notes Theor. Comput. Sci. 166 (2007) 5–26 https://doi.org/10.1016/j.entcs.2006.06.013.

[69] A. Abouzahra, A. Sabraoui, Afdel A, A practical approach for extending DSMLs by composing their metamodels, Adv. Sci. Technol. Eng. Syst. J. 3 (6) (2018) 358–371 https://doi.org/10.25046/aj030644.

[70] A. Abouzahra, A. Sabraoui, K. Afdel, A metamodel composition driven approach to design new domain specific modeling languages, 1st European Conference on Electrical Engineering and Computer Science, 2017 https://doi.org/10.1109/EECS.2017.30.

[71] R. Bill, P. Neubauer, M. Wimmer, Virtual textual model composition for supporting versioning and aspect-orientation, 10th ACM SIGPLAN International Conference on Software Language Engineering (SLE 2017), 2017 https://doi.org/10.1145/3136014.3136037.

[72] F. Rabbi, Y. Lamo, L.M. Kristensen, A model driven engineering approach for heterogeneous model composition, Comm. Com. Inf. Sc. 880 (2017) 198–221 https://doi.org/10.1007/978-3-319-94764-8_9.

[73] F. Rabbi, L.M. Kristensen, Y. Lamo, Optimizing distributed resource allocation using epistemic game theory: a model-driven engineering approach, 5th International Conference on Model-Driven Engineering and Software Development (MODELSWARD), 2017 https://doi.org/10.5220/0006121400410052.

[74] N. El Marzouki, O. Nikiforova, Y. Lakhrissi, M. El Mohajir, Toward a generic metamodel for model composition using transformation, Procedia Computer Science 104 (2017) 564–571 https://doi.org/10.1016/j.procs.2017.01.173.

[75] N. Elmarzouki, Y. Lakhrissi, M. Elmohajir, A study of behavioral and structural composition methods and techniques, International Conference on Information Technology for Organizations Development (IT4OD), 2016 https://doi.org/10.1109/IT4OD.2016.7479254.

[76] N. El Marzouki, Y. Lakhrissi, O. Nikiforova, M. El Mohajir, The application of an automatic model composition prototype on the-two hemisphere model driven approach, 2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS), 2017 https://doi.org/10.1109/WITS.2017.7934673.

[77] L. Taylor, S. Sharma, Model composition via object-role modeling, SoutheastCon 2018, 2018 https://doi.org/10.1109/SECON.2018.8478867.

[78] M.D. Del Fabro, P. Valduriez, Semi-automatic model integration using matching transformations and weaving models, 22nd Annual ACM symposium on Applied computing (SAC'07), 2007 https://doi.org/10.1145/1244002.1244215.

[79] M.D. Del Fabro, P. Valduriez, Towards the efficient development of model transformations using model weaving and matching transformations, Softw. Syst. Model. 8 (3) (2009) 305–324 https://doi.org/10.1007/s10270-008-0094-z.

[80] S. Roychoudhury, J. Gray, F. Jouault, A model-driven framework for aspect weaver construction, Lect. Notes. Comput. Sc. 6580 (2011) 1–45 https://doi.org/10.1007/978-3-642-22031-9_1.

[81] J. Bowles, M. Alwanain, B. Bordbar, Y. Chen, Matching and merging scenarios automatically with Alloy, Comm. Com. Inf. Sc. 506 (2015) 100–116 https://doi.org/10.1007/978-3-319-25156-1_7.

[82] C. Debreceni, I. Ráth, D. Varró, X. De Carlos, X. Mendialdua, S. Trujillo, Automated model merge by design space exploration", Lect. Notes. Comput. Sc. 9633 (2016) 104–121 https://doi.org/10.1007/978-3-662-49665-7_7.

[83] P. Rittgen, Collaborative modeling: roles, activities and team organization, Int. J. Inf. Syst. Model. Des. 1 (3) (2010) 1–19 https://doi.org/1prédocument018/jismd.2010070101.

[84] T. Halpin, Object-Role Modeling Fundamentals: A practical guide to data modeling with ORM, Technics Publications, 2015. ISBN: 978-1634620741

[85] M. Boeker, W. Vach, E. Motschall, Google Scholar as replacement for systematic literature searches: good relative recall and precision are not enough, BMC. Med. Res. Methodol. 13 (1) (2013) 131 https://doi.org/10.1186/1471-2288-13-13.