Conflict Management Techniques for Model Merging: A Systematic Mapping Review

Mohammadreza Sharbaf, Bahman Zamani, Gerson Sunyé

Noname manuscript No.
(will be inserted by the editor)

Conflict Management Techniques for Model Merging: A Systematic Mapping Review

Mohammadreza Sharbaf · Bahman Zamani · Gerson Sunyé

Received: date / Accepted: date

Abstract Model merging conflicts occur when different stakeholders aim to integrate their contradicting changes that are applied concurrently to update software models. We conduct an extensive systematic mapping study on conflict management techniques and relevant collaboration attributes to the versioning and merging models from 2001 to the middle of 2021. This study follows the standard guidelines within the software engineering domain. We analyzed a total of 105 articles extracted from an initial pool of more than 1800 articles to infer a taxonomy for conflict management techniques. We use this taxonomy to classify existing approaches to understand characteristics, shortcomings, and challenges on conflict management techniques in merging models. It also provides a solid foundation for future work in this area. We show that syntactic conflicts are the most studied type and that the top three popular conflict detection techniques are constraint violation, change overlapping, and pattern matching. We observe the lack of a comprehensive state-of-the-art comparison between academic or industrial tools, as well as the need for real-world case studies. Finally, we show that recent trends have focused on online collaboration, where teams of stakeholders work on large-scale models.

Keywords Conflict Management · Model Merging Conflict · Collaborative Modeling · Model Driven Engineering · Taxonomy · Systematic Mapping

Mohammadreza Sharbaf
MDSE Research Group, Department of Software Engineering, University of Isfahan, Isfahan, Iran
LS2N, University of Nantes, Nantes, France
E-mail: m.sharbaf@eng.ui.ac.ir

Bahman Zamani
MDSE Research Group, Department of Software Engineering, University of Isfahan, Isfahan, Iran
E-mail: zamani@eng.ui.ac.ir

Gerson Sunyé
LS2N, University of Nantes, Nantes, France
E-mail: gerson.sunye@univ-nantes.fr

1 Introduction

Conflict is inevitable, when several users work in parallel on the same artifact. However, to err is human, and people need more time to analyze problems when working at high levels of abstraction [1]. Consequently, people must work together to complete large projects in a reasonable time and have other people to help them catch their mistakes. Recently, several pieces of research are conducted to propose methods to enable large teams of developers to work in a collaborative manner [2]. Collaborative software engineering (CoSE) is a discipline introduced in the early 1990s to enhance collaboration, communication, and coordination among software development team members [3, 5]. Team collaboration is not only limited to technical collaborators, such as developers and designers, but it also includes external and non-technical stakeholders, such as domain experts, non-technical managers, customers, and end users [6]. CoSE can be applied to manage the evolution process of various artifacts in the lifecycle of software system development. When focusing on software design, models are the primary artifact. Models present an abstraction of specific aspects of the systems and help to early assess based on the domain-specific concepts [13].

Model-driven software engineering (MDSE) [7, 8] provides appropriate techniques and tools to cope with the complexity of software system development by shifting from code-centric to model-centric paradigm. In MDSE, models are considered as first-class entities which are used for descriptive and prescriptive purposes [9]. Such software models are usually created by more than one user and are resulted from team collaboration in software modeling. Multi-user modeling relates to the domain of collaborative MDSE [13], which focuses on techniques and approaches in which several distributed stakeholders collaborate to analyze, produce, and manipulate models of a software system. To coordinate teamwork among different collaborators in a local or remote shared workspace, either synchronously or asynchronously, versioning systems have proven to be indispensable for managing the modifications and merging updated versions of a model [11]. However, some concurrent updates may be incompatible and contribute to conflict during the merge process [17]. Consequently, the collaborative modeling frameworks and environments should provide support for concurrency in modeling time and manage conflicts in the merge process.

Conflict management in model merging deals with techniques, activities, and tools for enhancing consistency in the result of the merge process. To support conflict management in model merging, several approaches have been proposed so far. Most of those approaches provide conflict detection techniques to discover conflictual situations that may occur due to concurrent changes. Some approaches focus on reconciling conflicts with techniques independent of the context [87] and work based on a three-way model merging strategy [13]. Other approaches try to prevent conflicts by using mechanisms to inform the user about conflicting situations for concurrent modifications [15]. Despite these ef-

forts, a precise taxonomy or classification for conflict management techniques has not yet been provided.

To address the aforementioned issue, this study focuses on the approaches that are proposed to manage conflict or inconsistency in the integration of models. Despite the lack of a well-defined terminology, models integration can be distinguished into activities such as merge of model versions, the composition of multi-view models, weaving of partial models, and synchronization of different views that projects a common model [19]. In this study, our definition of conflict management approach in model merging is (i) a technique that tries to prevent conflicts and build a consistent and integrated model by warning collaborators from discordant changes or (ii) a mechanism that helps users describe a conflictual situation to detect and resolve inconsistencies and conflicts during the merge process. Several researchers [47,48,49,112] have addressed conflict management activities where each approach focused on one or more aspects. This variety of approaches may indicate that each conflict management activity can be carried out in different ways. Although quick overviews of conflict management approaches have been discussed in earlier papers, no extensive study has been reported a systematic comparison of conflict management approaches to model merging. This paper aims to elaborate a classification framework to help understand existing approaches related to model merging conflict management by identifying their current characteristics, shortcomings, and challenges.

To summarize, the main contributions of this study are as follows:

- Provide definitions of the conflict management dimensions in model merging: conflict specification, conflict preservation, conflict detection, conflict resolution, and conflict awareness;
- Present a taxonomy for conflict management in model merging, starting with the provided definitions to identify all possible conflict management techniques;
- Create a reusable classification framework for understanding, classifying, and comparing the characteristics of present and future work on model merging conflict management using the elicited attributes;
- Identify the challenges, shortcomings, and publication trends of the existing conflict management approaches in model merging.

The rest of this paper is organized as follows. Section 2 briefly provides background information on collaborative software engineering, collaborative modeling, and model merging conflicts. Section 3 provides goals and poses research questions as well as explains the research methodology for performing systematic mapping study. Section 4 represents the extracted conflict management taxonomy. The results of systematic mapping are reported based on the research questions in Section 5. Section 6 provides an overall discussion about open research challenges. Section 7 presents the related work and is followed by threats to validity in Section 8. Finally, Section 9 concludes the paper and highlights areas for future work.

2 Background

In this section, we introduce the foundation of conflict management in model merging and discuss the collaborative modeling characteristics that might affect managing conflicts in this domain. More specifically, we first introduce the concept of collaborative software engineering in Section 2.1. Then, we focus on the collaborative modeling in Section 2.2 and describe its characteristics helping researchers distinguish current approaches on the topic. Finally, we present the main reasons for conflict in the model merging process in Section 2.3.

2.1 Collaborative software engineering

Software projects naturally require collaboration among team members to produce a large and complex software system [4]. Collaborative Software Engineering (CoSE) [5] refers to the techniques and tools used to support *coordination*, *communication*, and *collaboration* between teams of stakeholders with different expertise [13]. The coordination infrastructure deals with methods and tools for creating and versioning artifacts, as well as managing their persistence during the project life cycle. The communication relies on messages, emails, videos, audios, and annotations to allow collaborators to be aware of each other's activities. Moreover, the collaboration support provides a set of means, such as shared workspaces, versioning systems, comparison techniques, merging mechanisms, and process management. In this context, Version Control Systems (VCSs) [12] provide the foundation for collaborative software development. They facilitate storing the history and sharing the versions of one artifact, such as source code [11].

VCSs support distributed software development by imparting the stakeholders' access to particular artifacts independent of their location [1]. To make the concurrent development of artifacts possible, VCSs must permit the modification of one artifact simultaneously and merge the various alternative modifications in one new version of the artifact. Merging parallel modifications in code-centric software engineering is a well-developed task when the artifacts in a VCS are text files, which are one-dimensional and only need line-based processing [11]. But, a line-based version control poses a major challenge if applied for software models defined in a graphical space, which are at least two-dimensional and consist of a complex internal representation [17]. When multiple stakeholders with different technical information collaborate on the software models, the syntax and rich semantics of modeling languages are important to understand modifications. While using the line-based approaches, the interconnected and graph-like nature of models are absolutely neglected [20]. For example, the line-based VCSs report modifications by mistake if identical model elements are serialized in a different order. Furthermore, such systems cannot prevent and, even more problematic, resolve conflicts to produce a consistent merged model version [11]. Focusing on collaboration

modeling tools, specifically versioning systems dedicated to managing model artifacts, is necessary to address these shortcomings.

2.2 Collaborative modeling

Model-Driven Software Engineering (MDSE) [9] promotes the migration from code-centric to model-based development, where models are used as first-class entities throughout the whole software engineering life cycle [8]. Models define a system at a high abstraction level for a particular purpose using a simplified representation of the reality [7]. Model-driven approaches adopt the principles of separation of concerns to represent systems as collections of interconnected models expressed in proper domain-specific modeling languages [9]. Similar to the code-centric approach, the model-based development of complex software-intensive systems requires stakeholders with different backgrounds and skills to collaborate on various aspects. This collaboration results in the software models that define the specification, architecture, and design of the system. In this context, the joint creation of models is considered as collaborative modeling, which depends on a set of collaborative means such as model versioning systems and model merging mechanisms for allowing stakeholders to coordinate themselves as a team [13, 14].

To ensure collaborative modeling, model versioning is a crucial activity for managing the model evolution process [21]. Model versioning systems represent modeling artifacts and deltas using the states or change operations [78]. In the state-based versioning, model versions are expressed using a set of entities and relations, while change-based versioning represents model versions as a set of change operations that are applied to produce models. Model versioning systems must support the merge process in order to obtain a consistent, integrated version for concurrent evolved software models [20]. The model merging process is composed of the main phases comparison, conflict reconciliation, and merge [17, 19]. Incorporating the language-specific knowledge in the merge process may considerably improve the conflict management activities.

In collaborative modeling, change propagation occurs in an offline or online scenario [123, 158]. Offline collaboration is based on asynchronous interactions. In this scenario, users check out models from a version control system (VCS) and commit local changes to the repository. In the online scenario, users work synchronously and may simultaneously edit a model. Changes are immediately propagated to all users. While changes are propagated in both offline and online scenarios, appropriate mechanisms for conflict management are required in the merge process [15]. This includes the support for *conflict detection* [11] in which potential conflicts are discovered, *conflict awareness* that warns users of potential conflicts, and *conflict resolution* by which detected conflicts are fixed.

In any conflict management technique, the features provided by the collaboration environments can make a difference. Franzago et al. [13] distinguish several attributes of collaborative modeling approaches with respect to the

model management facilities. Moreover, Masson et al. [15] detailed the important features of collaborative modeling environments. In the following, we introduce the categories of *collaborative modeling characteristics* that affect on how to manage the conflict in model merging according to the attributes and features represented by Franzago et al. [13] and Masson et al. [15]. For each characteristic, we describe possible options provided by various approaches over time. We can use the mentioned characteristics for analyzing the conflict management techniques and building a classification framework in the context.

1. *Model type* [10] refers to the characterization of modeling languages that are supported by conflict management techniques for model merging. Therefore, following the proposal of Hojaji et al. [16], we considered the generality level of target modeling language that is stated by a conflict management approach. This way, we can identify the following attributes:

   Any refers to approaches that are independent of the modeling language and can be used to manage conflict in the merge of models conforming to any modeling language.

   UML models refer to approaches that have been stated to be specifically applicable to manage the merging conflict for UML (and UML subset) models.

   EMF-based refers to approaches specifically applicable to manage the merging conflict for domain-specific modeling languages implemented by Ecore metamodels using the Eclipse Modeling Framework [10].

   Workflow models refer to approaches specifically applicable to manage the merging conflicts for modeling languages that define a business process or flow of the work, such as BPMN and DFD.

   Other refers to approaches that can be used to manage the merging conflict for a restricted set of modeling languages (e. g., ER and Feature modeling languages). Approaches in the category Other cannot support modeling languages that are included in UML, EMF-based, or Workflow categories. While approaches in the category Any can support conflict management for any modeling language, the approaches in the category Other are limited to a specific set of modeling languages.

   Note that conflict management approaches can be language-independent or language-specific. The model type for the language-independent approach is *Any*. But language-specific approaches might support one or more categories among *UML models*, *Emf-based*, *Workflow models*, and *Other*.

2. *Repository architecture* [15] refers to the architecture used to locate model storage for the collaborative environment. *Centralized* and *Decentralized* are two fundamental types of architecture that respectively use a central authority or a distributed authority for the models' primary storage location. Centralized and decentralized architectures are two mutually exclusive possibilities.

3. *Versioning strategy* [11] refers to the mechanism used to access and modify models. *Pessimistic* and *Optimistic* are two versioning paradigms. In the pessimistic paradigm, only one user is allowed to modify a model, which leads to conflict prevention. In contrast, the optimistic paradigm allows users to modify a model in parallel, requiring conflict detection and resolution. An approach may support both pessimistic and optimistic versioning paradigms.

4. *Collaboration mechanism* [15] refers to the scenarios of sharing changes between collaborators. *Online* and *Offline* are two different scenarios of collaboration. Online collaboration presents a synchronous mechanism, where users are informed of the changes made by others, immediately. In contrast, offline collaboration uses an asynchronous mechanism, in which users can apply modifications locally and push the changes later. A conflict management technique may support both collaboration mechanisms.

5. *Comparison technique* [11] refers to techniques used to detect the differences between two versions, or the changes that are applied on a new version. We can identify the following comparison techniques, which can be used in parallel by a conflict management approach.
   State-based comparison detects differences by simply comparing the states of the model version elements based on matching techniques.
   Operation-based comparison works by tracking and saving the change operations, which encodes the original version's differences.
   Hybrid comparison combines state-based and operation-based techniques to synthesize the change operations, based on the matching techniques and identifying similar concepts in different versions.

6. *Merging technique* [11] refers to the mechanism for integrating different versions of a model. In the following, we introduce three variants of the merge technique.
   Raw merging simply applies a sequence of all the users' changes over the original version to create an integrated version.
   Two-way merging compares all new versions and integrates them into a single version without having access to the original version.
   Three-way merging attempts to compare all new changes based on the original version to apply a precise combination of them into a merged version.
   Note that each conflict management technique is mostly introduced based on a specific merge mechanism but may also support others simultaneously.

7. *Ordered feature support* [86] refers to approaches that can manage conflict for merging ordered features, which are relevant only for multi-valued elements, e. g., attributes and references, that have different values under different conditions.

2.3 Model merging conflict

In the model merging process, conflicts may arise due to a set of *contradicting changes* or different modifications with the *same intentions* [23]. The first is raised when concurrent incompatible modifications are applied to the same model element. For instance, in updating a UML class diagram, two modelers rename the same class with different intentions. The second results when modelers, due to the same intention, modify different model elements but with the same meaning. For example, in updating a UML activity diagram to define a parallel flow, a modeler adds an expansion region, while another modeler uses split and join bars. In such situations, either the modifications cannot be integrated to produce a unique model, or the integration would result in an inconsistent merged version of the model. The main reason for conflicts is the existence of modifications, which do not commute [78]. According to this, different types of model merging conflicts can be defined by considering the syntax and the semantics of models [17]:

Syntactic conflicts are those which take the modeling language syntax into account. As an example, the *dangling reference* is a syntactic conflict in a UML class diagram that occurs when one modeler adds an association between two classes, whereas another modeler concurrently deletes one of those classes [11]. However, syntactic conflicts may not necessarily produce a language syntax violation. For instance, contradicting updates to the name of a same model element constitutes a syntactic conflict. This type of conflict may be detected by checking the structure of models and comparing the similarity of elements.

Semantic conflicts go beyond the syntactic conflicts and need to consider the system behavior and the modeling language semantics. Indeed, the integration of concurrent changes may result in a syntactically correct merged version, yet semantically invalid [23]. For example, the *equivalent associations* [112] is a semantic conflict that arises when two modelers use different ways to model the parent relationship for the class Person in a UML class diagram. The first modeler adds one association with role name *parents* to express the fact that a Person has two *parents*. In parallel, the second modeler adds two associations to describe the fact that a Person has one *mother* and one *father*. In this situation, the merged model would contain three associations that lead to a semantic conflict, considering the ontological equivalence. Semantic conflicts are divided into three categories: Static semantics, behavioral semantics, and semantic equivalence [11].

In this context, *static semantics* conflicts refer to issues and side effects detected at compile-time, such as violation of hierarchy constraints and incompatible types, which remain hidden in the merged version without considering the model semantics. An example of static semantics conflict is the *polyforest cycle* [112], where adding edges in different versions of a directed acyclic graph may lead to a cycle in the merged version. In most modeling

languages, e. g., UML, the static semantic conflicts violate the well-formedness rules and could be misidentified as syntactic conflicts. *Behavioral semantics* conflicts denote different or unexpected behavior in the merged model, relying on runtime semantic. These conflicts affect the execution behavior of modeled systems based on the *data* and *control flow*. An unwanted *control flow loop* [112] in the UML activity diagram is an instance of behavioral semantic conflict. *Semantic equivalence* conflicts denote contradicting conditions in which the modeler expresses the same meaning in different ways using equivalent concepts or equivalent constructs. In those conditions, syntactically, both modifications can be applied because of their different appearance, while only one of them should be applied to perform a valid merged model. The *equivalent associations* conflict described above is an example of semantical equivalence conflict.

3 Research methodology

In this section, we describe the research methodology that we have adopted to carry out this study. Systematic mapping has a well-defined methodology that is originally used in medical research [33]. We followed a research methodology in accordance with the guidelines provided by Brereton et al. [31] and Petersen et al. [32,33] for performing systematic mapping studies. It starts from an independent objective analysis that organizes the research flow to obtain meaningful results. Fig. 1 depicts the overview of our research protocol, which is divided into three main phases:

1. *Planning phase:* establish the need for a review, identify the goals and the main research questions, and define the inclusion and exclusion criteria (see Sect. 3.1).
2. *Conducting phase:* perform the review based on the planning phase and extract relevant data from the selected primary studies to create conflict management taxonomy and systematic map (see Sect. 3.2).
3. *Reporting phase:* report and discuss the results of the review and perform an analysis of possible threats to validity (see Sect. 5 to Sect. 8).

The review was conducted in an iterative fashion. To mitigate potential threats to validity, the results of article selection and data extraction activities are reviewed by one of the authors who was not involved in that phase of the study process. We analyzed the most relevant articles from the first publication in the domain in 2001 until the advent of this article at the end of June 2021. Note that we found some articles published before 2000 that proposed techniques to check consistency between different artifacts. But those approaches did not focus on the integration of models and were removed from the potential articles based on the exclusion criteria.

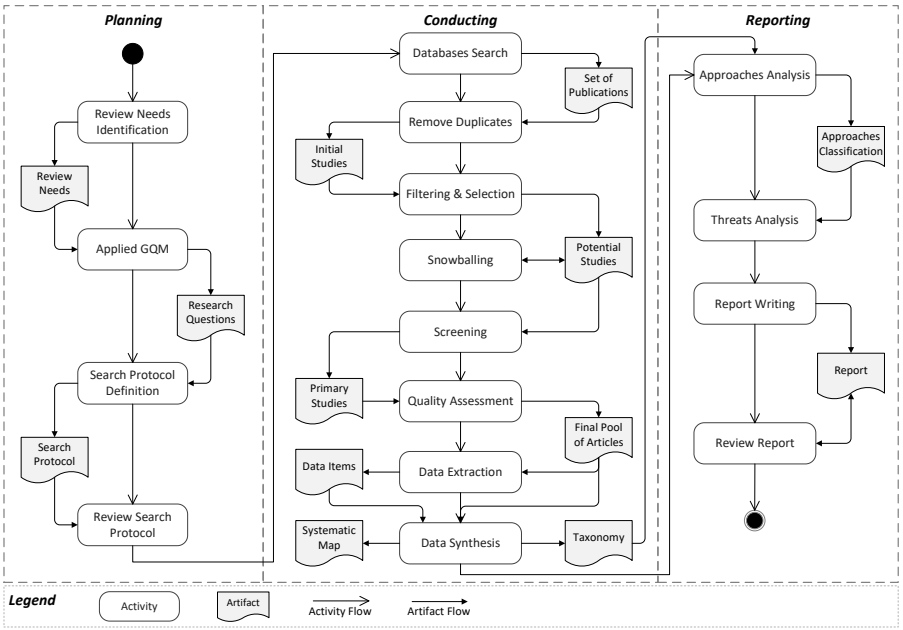


Fig. 1 Research protocol followed in this study

3.1 Planning the review

Our systematic study followed a strict research protocol to improve its quality, satisfaction, and influence. Planning phase defines the research protocol for our study. In this section, we first outline our motivations for conducting this study. Then we follow the Goal/Question/Metric (GQM) paradigm to define goals and research questions. After that, we formulate the search string and define our search strategy. Finally, we list the inclusion and exclusion criteria. Research protocol might be repeatedly refined based on the search results, such as the synonyms of keywords we have not thought of before. Thereby, we conducted an iterative process when we examined the search string as well as the inclusion and exclusion criteria.

We provided a complete replication package to enable the easy reproduction of our systematic review. The prepared package that has been made publicly available[1] includes a spreadsheet containing a list of filtered articles, the classification of primary studies, and extracted data.

3.1.1 Motivation

There exist surveys and systematic reviews that focus on different forms of model management [9]. Altmanninger et al. [11] provide a survey on model versioning approaches. They also discuss different types of conflict in model

[1] `https://github.com/MSharbaf/CMSysMap`

merging, however, focusing more on the architecture of versioning systems. Mens [17] provides a comprehensive survey and analysis of available merge approaches. A survey [25] and two systematic studies [26,27] discuss model comparison algorithm and state-of-the-art. Brunelière et al. [28] provide a detailed overview of important characteristics to support view-based modeling. However, none of those studies focuses on conflict management techniques in the model merging process.

Although the concept of model merging conflicts has gained significant attention in academia and industry, it is still unclear which are the most used techniques to manage model merging conflicts. So far, several approaches propose to manage model merging conflicts, and this research topic has been still active for more than a decade. However, no evidence helps to understand different aspects of the existing approaches in this area. Thus, our study aims to identify and extract key attributes of existing approaches to present a classification scheme that could help researchers to compare present and future research on conflict management in merging models. To summarize, our main motivations behind conducting this systematic review are as follows:

- Conflict management is an important task in the model merging process and is crucial for the collaborative development of software systems.
- There is no survey that focuses on the classification of conflict management techniques.
- There is no taxonomy to specify various dimensions and possible solutions to support conflict management in model merging.
- The analysis of the aforementioned topics can help researchers find characteristics, trends, limitations, and gaps of the current conflict management approach in model merging.

3.1.2 Goals

Formulating the goal of our study is a pivotal task to have a clear definition of the research questions. We use the GQM paradigm [36] to identify the goals of this study, specifying meaningful research questions and carefully identifying potential metrics that are later used as attributes. The aim is to have a systematic data extraction process to collect metrics from our data and understand how to use them to create a systematic map. We define the research goals of this study as follows:

G1. Identify the nature and maturity of research in the model merging conflict management area
G2. Understand the influence of collaborative modeling characteristics on the various conflict management techniques
G3. Identify the current state of the art and practice in the model merging conflict management area
G4. Classify the major challenges, limitations and future directions for upcoming research in the model merging conflict management area

3.1.3 Research questions

In this section, we summarize the research questions of the study based on the aforementioned research goals to specify the review scope of the mapping study. Note that the construction of a systematic map needs a careful association of the facets with the high-level attributes of the articles. We consider *conflict management facet*, *contribution facet*, and *research facet* to better understand the systematic map of the conflict management techniques. We describe these facets as follows:

- The *conflict management facet* refers to different dimensions that structured the topic of conflict management on a higher level of abstraction to show partial relevance between conflict management approaches and the extracted attributes.
- The *contribution facet* reflects the type of contribution that is driven from the selected articles, which for example, could be a process, an algorithm, a method, or a tool.
- The *research facet* refers to the research type of the approach used in the articles to express the knowledge type of research in general and independent from a specific focus area.

Furthermore, we underlined the metrics that we gathered for the systematic mapping as part of the research questions.

G1. Identify the nature and the maturity of research in the model merging conflict management area

Q1.1. Which **conflict management facets** are being employed for model merging?
Rationale. We are interested in understanding the different dimensions of conflict management employed for model merging. Dimension refers to a set of activities that support a specific goal in the field of conflict management, e. g., conflict detection or conflict specification.

Q1.2. Which is the **contribution facet** of the articles?
Rationale. We are interested in knowing the contribution facets the researchers tend to focus on for each conflict management activity. We use five contribution facets that were derived from the keywords by Petersen et al. [33]. Hence, we consider the keywords that are used by the authors of articles to distinguish the type of contribution for each article. The contribution facets are defined as follows:
  - Tool: Article proposing a new tool or improving an existing one and describing its evaluation.
  - Method or Approach: Article proposing a new approach or improving an existing one.
  - Model or Framework: Article introducing a new framework or a new model for managing conflicts.
  - Process or Algorithm: Article proposing a new algorithm or describing a conflict management process.

- Metric or Benchmark: Article introducing a benchmark or proposing comparative metrics for conflict management techniques.

Q1.3. Which is the **research facet** of the articles?
Rationale. We are interested in knowing the maturity of the research for each conflict management facet in terms of the solution or evaluation provided in the articles. We use the classification of research facets proposed by Wieringa et al. [35]. To distinguish the research facet that reflects the specific focus of an article, we investigate the novelty of work, the level of implementation, and the quality of evaluation. The research facets are summarized as follows:

- Validation Research: Article investigating the properties of a solution and technique that have not yet been implemented in practice. Validation research includes experiments in the lab and other validation procedures such as simulation, mathematical analysis, and proof of properties.
- Solution Proposal: Article proposing a new solution, arguing for its relevance, and describing its applicability with the help of examples, arguments, and prototyping, without a full-blown validation.
- Evaluation Research: Article introducing an implementation of a conflict management technique and evaluating the proposed technique using empirical methods. Evaluation research investigates the novelty of the knowledge claim made by the research results and the soundness of the research method used.

Q1.4. Which is the **tool support maturity level** for available conflict management techniques in the area?
Rationale. To measure the maturity level of conflict management techniques, we use the four-level scale proposed by Cuadros López et al. [34] as follows:

- Level 1 *(not implemented)*. The approach is not implemented in a tool.
- Level 2 *(partially implemented)*. The approach is implemented in a prototype tool but not all features are supported.
- Level 3 *(fully implemented)*. The approach is completely implemented in a tool. The tool has been used for several applications to validate the approach.
- Level 4 *(empirical evaluation)*. The approach is completely implemented in a tool, and the tool has been evaluated empirically. Note that the empirical evolution level has thereby been closely validated by end-users, while the fully implemented level has only been tested by developers.

G2. Understanding the influence of collaborative modeling characteristics on the various conflict management techniques

Q2.1. Which **kind of merging conflict** is mostly supported in the area?
Rationale. The model merging conflict is a situation in which different versions of a model cannot be integrated into a unique version, which is

syntactically and semantically valid. Merging conflicts can be divided into *Syntactical* and *Semantical* conflicts. Here, we want to know which type of conflict is supported by existing approaches.

Q2.2. Which **collaborative modeling characteristics** are crucial to the area?
Rationale. Our interest is to know which attributes are employed for collaborative modeling characteristics in conflict management techniques and whether a particular attribute impacts on conflict management activities.

G3. Identify the current state of the art and practice in the model merging conflict management area

Q3.1. Which **type of conflict specification techniques** are being employed for model merging?
Rationale. We are interested to know what percentage of approaches propose a technique to specify conflicts and whether a particular specification technique is used for conflict description.

Q3.2. Which **conflict prevention approaches** are being used by researches in the area?
Rationale. Our interest is to understand which techniques are commonly used to avoid conflict situations and preserve consistency during the model merging process.

Q3.3. Which are the most used **approaches for conflict detection** by the academic and industrial researchers?
Rationale. We are interested to know which techniques are increasingly employed by researchers to discover conflict situations in the model merging process. Whether each conflict detection technique is used for a particular kind of conflict or spplies to all types of merging conflict.

Q3.4. Which kind of **conflict resolution mechanism** is widely used?
Rationale. We are interested in analyzing the resolution mechanisms of the investigated work to know how many approaches can (semi-) automatically resolve conflicts and how many of them prefer to delegate the resolution to the users. It is also interesting that approaches mostly resolved conflict during the merge process or postponed the resolution phase.

Q3.5. Which **conflict awareness methods** are being used in the area?
Rationale. Our interest is to understand how conflict management approaches have been warning users of conflict situations.

G4. Classify the major challenges, limitations and future directions for upcoming research in the model merging conflict management area

Q4.1. Which **limitations** have been reported?
Rationale. We are interested to know that all investigated articles have identified their weaknesses and what limitations they have in common.

Q4.2. Which are the **trends** in the area?
Rationale. Our interest is to understand that new technologies have led researchers to focus on which aspects of model merging conflict management.

Q4.3. Which **future research directions** are being suggested?
Rationale. We are interested to know about the recommended research directions by researchers for future work.

3.1.4 Search strategy

To retrieve the set of relevant studies, we followed a search strategy, which consists of five consecutive steps. Fig. 2 outlines our search and selection process. This process depicted a detailed view of steps *Databases Search* to *Quality Assessment* from Fig. 1. For planning the search strategy, we first selected databases based on their overall coverage, then identified major terms concerning our study, and defined relevant search strings for automatic database searches. In the following, we expand upon the search strategy definition.

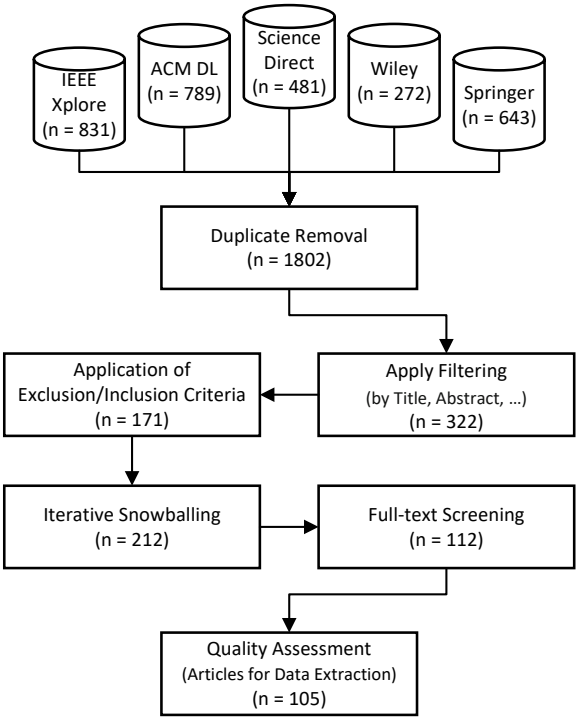


Fig. 2 Overview of search and selection process

3.1.4.1 Database selection

We started the search process by selecting the electronic repositories we used for the search: IEEE Xplore[2], ACM Digital Library[3], Science Direct[4], Wiley Online Library[5], and Springer Link[6], which are well known digital repositories that make up the primary source of articles for potentially relevant studies on software engineering [37]. At the earlier steps of the articles selection process, we do not use a search engine (e. g., Google Scholar) since their results may overlap with those from the selected repositories. Also, these engines do not have a robust advanced search that may result in many irrelevant articles. However, we use these engines to find potential studies in the snowballing process.

3.1.4.2 Search string formation

The guidelines provided by Kitchenham and Charters [38], PICO (*Population*, *Intervention*, *Comparison* and *Outcomes*) suggests to identify keywords and formulate appropriate search strings from research questions. We only used *Population* and *Intervention* elements of the PICO framework, and *Comparison* and *Outcomes* are not investigated to limit the broad scope of the study. In our context, *Population* refers to the application area e. g. *Model Merging*, and *Intervention* is a software methodology, tool, technology, or procedure that addresses a specific issue, e. g., *Conflict Specification*. The terms we used to form the search query include several keywords that were identified in an iterative fashion from research questions and the PICO criteria. After a series of test executions and refinements, we grouped keywords into three sets that are described in Table 1. The overall search strings used in all repositories to select relevant articles can be combined as Equation 1. In this equation, we use a disjunction operator between keywords in the same set, i. e., every search string must contain at least one keyword from each term set.

$$Search\ String = (A\ and\ (B\ and\ C)) \tag{1}$$

3.1.5 Inclusion and exclusion criteria

We defined a set of exclusion and inclusion criteria in order to further refine the quality of potentially relevant articles. To make these procedures precise and rigorous, Kuhrmann et al. [39] recommend aligning the inclusion and exclusion criteria with the research questions. To achieve this, we followed the suggestion of Franzago et al. [13] in the definition of these criteria to keep only articles

[2] `http://ieeexplore.ieee.org`
[3] `http://dl.acm.org`
[4] `www.sciencedirect.com`
[5] `https://onlinelibrary.wiley.com`
[6] `https://link.springer.com`

Table 1 Search String Keywords

Term Set	Keywords
A	*Model merging*, *Model composition*, *Model weaving* *Multi-view synchronization*, *Model versioning*, *Collaborative modeling*
B	*Conflict*, *Inconsistency*, *Interference*
C	*Manag**, *Prevent**, *Avoid**, *Specif**, *Detect**,*Discover**, *Fix**, *Resol**, *Reconcil**, *Represent**, *Awar**, *Warn**

** Zero or more characters in a word*

that focus on the scope of the study and avoid non-scientific works. We chose the inclusion (IC) and exclusion (EC) criteria of our study as follows:

IC1. Publications in which models are the primary artifacts.
IC2. Publications that present a technique or contribution for supporting the conflict in model merging.
IC3. Publications in peer-reviewed journals, conferences, or workshops.
EC1. Publications that are not in English.
EC2. Publications not focusing on MDE (e. g., conflict management in code-centric approaches).
EC3. Books, technical reports, dissertations, tutorial papers, or non-peer-reviewed publications.
EC4. Secondary study publications (e. g., summaries, systematic reviews, or surveys).
EC5. Publications appeared after the June 2021.
EC6. Publications that are not available in full-text for download.

3.2 Conducting the review

The second phase of our systematic review is conducting the selection process based on the search protocol that we defined in the planning phase. In the following, we first detail the search and selection process of primary studies for our review. Then we present the data extraction procedure and report the extracted systematic map from the primary studies.

3.2.1 Study selection process

Our study selection process was conducted in an iterative process to acquire relevant primary studies from the identified databases. This process comprises an automatic database search and an iterative snowballing-based search followed by screening, quality assessment, and selection of primary studies. In the following, we detail the selection process of our study.

3.2.1.1 Database search

To start the article selection process, we performed the defined search queries on each repository to acquire a set of relevant articles from the data sources that have been selected in the search protocol. Note that each digital repository has a particular syntax for a search string, which was configured to result only in research papers, such as journal articles, conference papers, and workshop papers. The number of articles obtained from each repository is shown in Fig. 3. Then we merged all the obtained results and removed duplicates that resulted in a total of 1802 articles as the initial pool of studies. However, due to the nature of indexing systems, some of the returned articles were quite irrelevant to the scope of our study. Hence, we reviewed the remaining articles and manually removed all irrelevant publications considering their title, abstract, introduction (if the article's goal was unclear from the abstract), and conclusion to reach 322 articles. After that, we manually applied all inclusion and exclusion criteria to the remaining articles in order to further refine their quality. This activity yielded 171 articles that satisfied all inclusion criteria and did not meet any exclusion criterion.

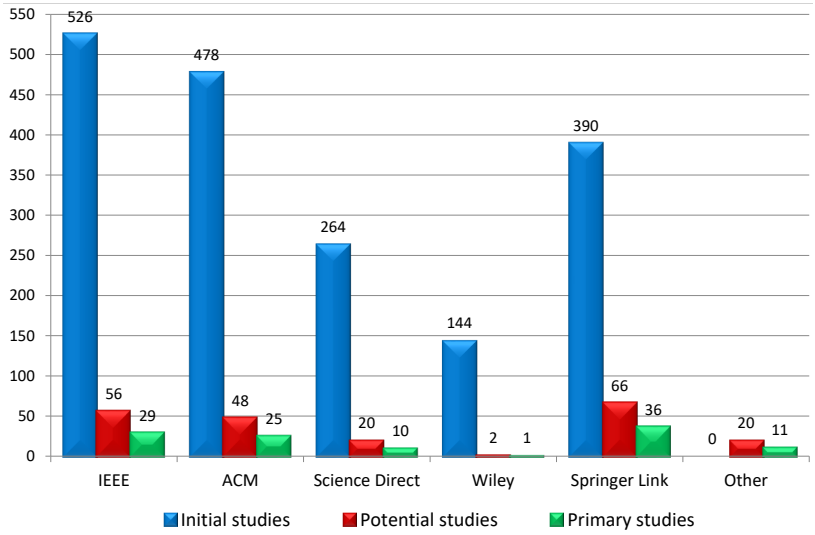


Fig. 3 Studies retrieved through online libraries

3.2.1.2 Snowballing-based search

To finish the search process, we complemented the results obtained from the last step with a backward and forward snowballing search [40] in order to ensure covering the broad scope of the study. The backward snowballing started by checking the references of 171 remaining articles. Forward snowballing performed further searches based on those articles citing the existing articles of

the pool in Google Scholar. The snowballing process is then repeated in three iterations with the newly included articles until we can no longer include new articles. This process led to 51 additional articles, which are mainly articles that have not been published in the well-known repositories. After that, we reviewed the full-text of the resulting articles and realized that only 41 articles covered well the inclusion/exclusion criteria as well as the scope of our study. With 171 articles obtained from the last steps and 41 articles added to the snowballing search, the potential pool of studies contained 212 articles.

3.2.1.3 Screening of publications

After identifying the potential pool of articles, all 212 articles from the search process should be reviewed in more detail. We performed a comprehensive reading task all over the full-text of the papers to select primary studies. The first author performed this activity, including the review of the abstract, all sections, and appendices (if any) of each paper. The second author reviewed the selection, which resulted in a filtered set of 112 relevant articles as primary studies. Fig. 3 shows the distribution of all articles analyzed during this study. The *initial studies* represent the total number of studies that were retrieved from each digital repository using the defined search strings, with a total of 1,802 articles. The *potential studies* show the number of studies remaining after the filtering, applying inclusion and exclusion criteria, and following an iterative snowballing that resulted in 212 articles. Among those 212 potential articles, 41 articles were found in the snowballing search, including six articles from IEEE, nine articles from ACM, two articles from Science direct, four articles from Springer Link, and 20 articles from other online publishers. Other refers to the conferences and journals (e. g., the Journal of Object Technology) that were not published in our selected repositories. Furthermore, the *primary studies* specify the number of studies remaining in our mapping study after the full-text screening, with a total of 112 articles. Moreover, the distribution of publication type for the primary studies is depicted in Fig. 4 that shows the most common publication type is *conference*.

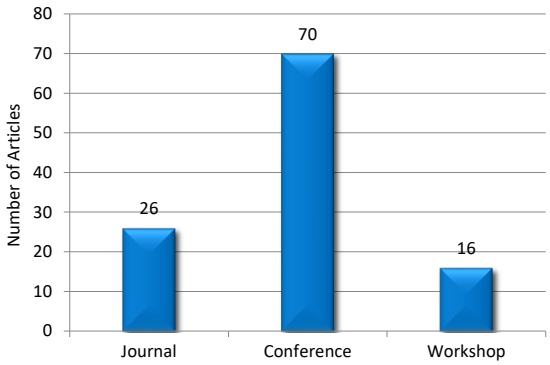


Fig. 4 Primary studies per publication type

Table 2 Quality Assessment Checklist

ID	Topic	Question
QA1	Objective	Is the purpose of the study clearly stated?
QA2	Motivation	Is the interest and the usefulness of the work clearly presented?
QA3	Approach	Are the techniques and concepts of the approach clearly defined?
QA4	Validity and Reliability	Did the study include a discussion on the validity and reliability of their results?
QA5	Empirical Evaluation	Did the study include a tool implementation or an empirical evaluation of the approach?
QA6	Related Work	Is the work measured and compared with similar works?
QA7	Future Work	Did the study point out potential further research?

3.2.1.4 Quality assessment

The attributes identification phase starts with the *Quality Assessment* step. In a systematic mapping review, assessing the quality of studies ensures that sufficient information is available to actually extract the information [32]. To determine the state of quality for the obtained primary studies, we followed the guidelines proposed by Kitchenham and Brereton [41]. We developed a seven-question checklist as presented in Table 2, based on the quality assessment suggestions given in work by Kitchenham and Charters [38]. We use a three level scale ('*Yes*', '*Partially*', and '*No*') for all questions, where their score values are *1*, *0.5*, and *0*, respectively.

For each primary study, the sum of scores for all questions can vary from 0 to 7. To improve the quality of the final pool of the study used for attribute extraction and map construction, we removed any primary study with a total score of less than 50 % (i. e., 3.5 of 7). The quality assessment was conducted by the first author, who answered all seven questions for each primary study. After that, the second author reviewed all articles excluded during the quality assessment process, which finally led to 7 primary studies being excluded.

Fig. 5 shows the percentage of three-level scale ('*Yes*', '*Partially*', and '*No*') for all primary studies over each quality assessment question. It shows that the *Objective* and *Approach* are well stated for all studies. Additionally, most of the studies (80 % to 88 %) score '*Yes*' or '*Partially*' in providing sufficient information for *Motivation*, *Empirical Evaluation*, *Related Work*, and *Future Work*. However, the *Validity and Reliability* question has a '*No*' score of 42 %, which is due to studies that have proposed only a new solution for this area with the help of examples. We also calculated the distribution of scores for each quality assessment question (Fig. 6), which shows the percentage of scores obtained by all primary studies for each question. The scores illustrate that QA1 and QA2 received the two highest percentages (20% and 19%), while QA4 has the least (8%). It indicates that in our quality assessment, the precise

explanation of the objective and the proposed approach in the articles had a greater impact than the representation of evaluation results.

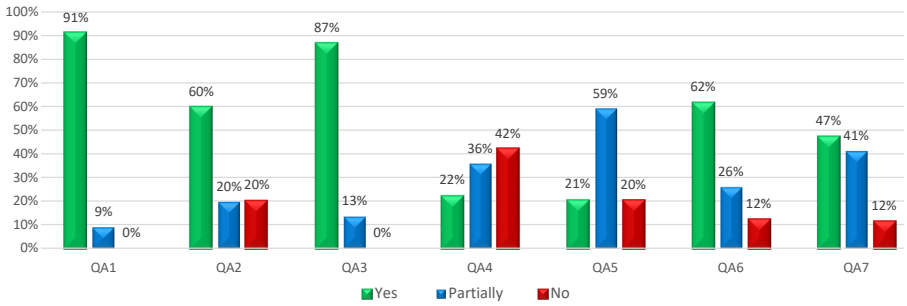


Fig. 5 Results of quality assessment of primary studies

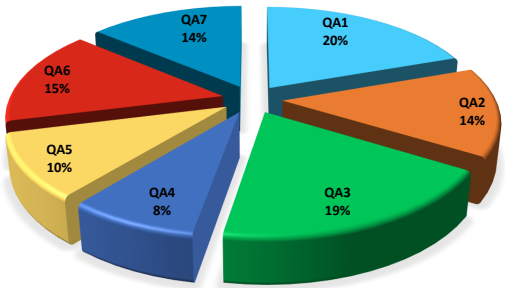


Fig. 6 Distribution of total score for quality assessment questions

To show how the total score of each primary study is dispersed from the average of the total score, we calculated the standard deviation for all primary studies. The average was 4.91, and the standard deviation returned 1.02. But from 112 primary studies that we evaluated based on the quality assessment checklist, seven articles did not achieve acceptance scores, 3.5 of 7, which led to the final pool of articles, was left with a tally of 105. We also calculated the average and standard deviation without considering the excluded articles that resulted in 5.07 for average and 0.83 for standard deviation. The results indicate that the average has gone larger for the final pool of articles, and the total scores of the remaining articles have been closer to the average. It shows that the quality of the final pool of articles has improved after performing the quality assessment step.

Fig. 7 shows the distribution of publication year for the final pool of articles in this study. It indicates the evolution of conflict management techniques for model merging. As depicted in the figure, the first specific work for managing conflict in the merge of model versions was published in 2001, and the number of publications reached its peak in 2010 (10 articles). We observe a sharp

decrease from 2010 to 2012. However, the total number of articles published from 2013 to 2020 relatively increased, and the average number of published articles is 7.25 articles per year. Moreover, we recorded four articles by mid-2021, while the main conferences in this domain (e. g., MODELS) will be held in the second half of the year. Finally, the review of articles published in the last five years shows that the trend is slowly going towards a significant change over new technologies. This change promises publication of new research results in the conflict management domain.

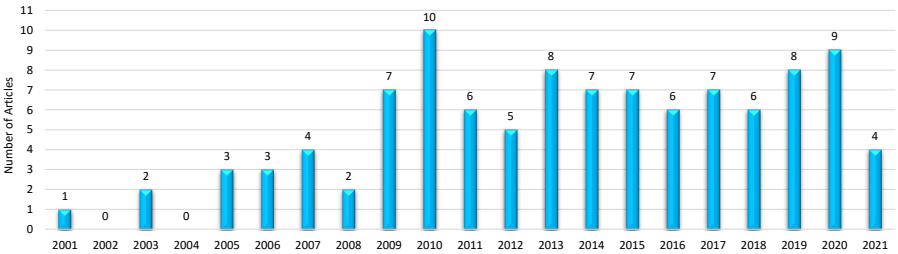


Fig. 7 Final pool of articles per years

3.2.2 Data extraction and synthesis

By finalizing the pool of articles, we followed the conducting phase with the *Data Extraction* activity to identify and extract attributes of conflict management techniques that are required to define a systematic map and taxonomy for the classification of conflict management approaches. To achieve this, the first author went through each article's full-text and derived the main keywords and concepts from all final pool articles based on the *Keywording* [33] method. After that, the second author reviewed the extraction by tracing back the initial attributes to each article's statements and checking their correctness to create the set of initial attributes (data items) as the output of this step.

Moreover, we performed an iterative refinement and detailed analysis of the extracted data items to create a robust and effective classification framework. We applied a card sorting technique [42] on keywords and concepts that have been extracted in the previous step to categorize them. More precisely, the obtained values for each conflict management attribute were considered as cards and authors sorted them into categories. To this end, the first and second authors established the initial set of categories. Then, the third author, who was not involved in the *attribute extraction* step, reviewed and refined the initial set of categories iteratively and combined similar items to classify the context clearly. After that, we categorized attributes based on the knowledge areas provided by Franzago et al. [13] and Masson et al. [15]. Note that the mentioned references offer knowledge about all collaborative modeling areas. We only employed them as a defining factor to prepare a high-level abstraction of attributes.

We also performed a sensitivity analysis by considering a random sample of five articles from the final pool in which all authors discussed and resolved any ambiguity or disagreement. Finally, we applied the data synthesis activity to collect and summarize the extracted data. We followed the recommendations and guidelines presented by Cruzes and Dybå [43] to understand and classify current conflict management techniques with the goal of answering the research questions mentioned above. In the following sections, we present the obtained results as well as the systematic map and taxonomy of conflict management techniques that were the output of this step.

4 Conflict management techniques in a nutshell

In this section, we describe the characteristics of conflict management techniques with respect to the current state-of-the-art of the collaborative modeling field. We first outline the taxonomy that has been extracted by following the research methodology in Section 3. Then, we represent the obtained systematic map to provide a bird view on the existing approaches.

4.1 Conflict management taxonomy

Fig. 8 shows a taxonomy for conflict management techniques. This taxonomy includes all the identified concepts and attributes concerning the conflict management aspects for model merging, which can be used to classify existing and future approaches in this domain. In the taxonomy, a feature can be either "Mandatory" or "Optional". This choice is based on the obtained domain knowledge, on authors' intuition, and on the frequency of attributes in the results of our systematic review.

According to the conflict management taxonomy, a conflict management approach must include all mandatory aspects and specify their alternative sub-features or any sub-features included for optional aspects. The top-level mandatory aspects in our taxonomy are "Conflict Specification", "Conflict Detection", and "Conflict Resolution". Missing any of these features, a conflict management approach is unreliable since it cannot guarantee the compatibility of the merged model. The conflict specification is mandatory to start each conflict management activity since it is essential to know what situation is considered as a conflict. Particularly, conflict detection and conflict resolution are the mechanisms that can be applied to ensure that the merge result is consistent. Moreover, "Conflict Awareness" and "Conflict Prevention" are optional aspects that help notify users and minimize the number of conflicts in the merging process. Both aspects are not crucial to ensure that the merging process results in a conflict-free model. Note that the lock-based prevention techniques are inflexible, and other prevention techniques cannot be perfect to avoid all conflicts. Therefore, we assume that conflict prevention cannot substitute conflict detection and resolution.

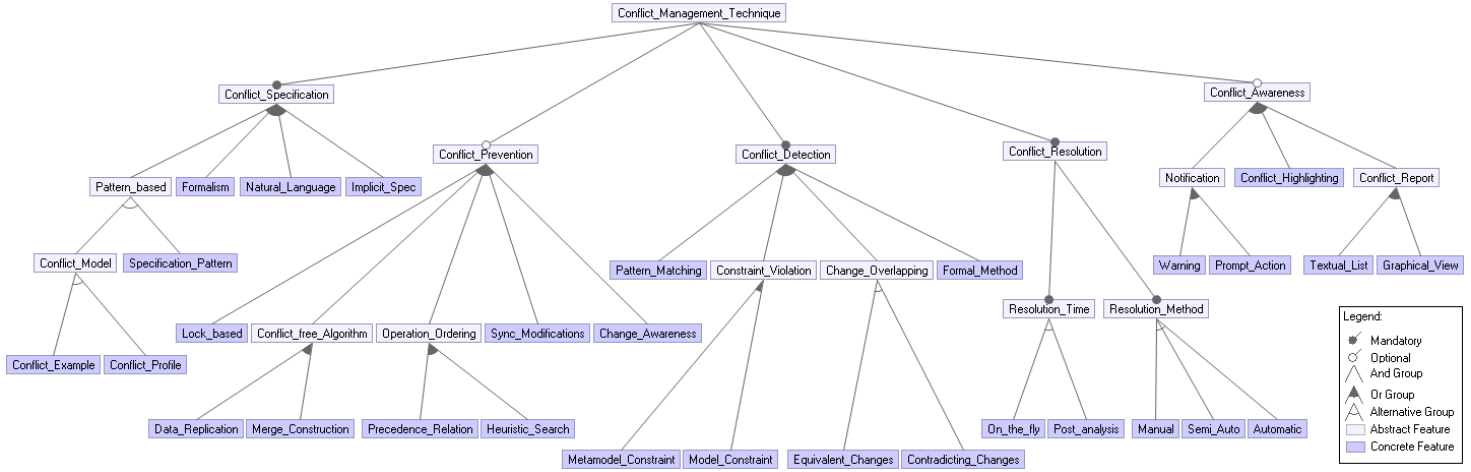


Fig. 8 Taxonomy for model merging conflict management techniques

In the presented taxonomy (Fig. 8), those sub-features that only one of them must be chosen, are specified as "Alternative Group," and others that different sub-features can be used concurrently are established as "Or Group". Furthermore, the "And Group" defines different dimensions of a top-level feature using mandatory or optional sub-features. Moreover, an "Abstract" feature is used to build groups of features with a specific connection, while "Concrete" features can be instantiated to reflect actual techniques in the domain. In the following, we describe the specified attribute sets for each aspect of the conflict management taxonomy.

4.1.1 Conflict specification

This concept focuses on the techniques used for describing the conflict situation and specifying the condition in the model merging process in which a conflict may occur. We define the following attributes to categorize conflict specification techniques identified in the investigated articles.

- *Pattern-based* refers to techniques that propose a template-based method to describe a conflict situation. Conflict patterns can be expressed through different approaches such as conflict profile, conflict model, or even example-based approach. For instance, A14 [67] provides a UML profile for modeling a conflicting case by defining examples of the conflict parties, while A26 [89] proposes a model-based pattern language to specify conflict patterns by visualizing undesired scenarios.
- *Formalism* refers to techniques that provide a formal language or grammar to specify merging conflicts. For example, A17 [70] uses Constraint Logic Programming [151] to provide a formal definition of conflicting situations, while A41 [112] presents a conflict definition formalism to specify conflicts using the GEBNF [152] meta-notation.
- *Natural language* refers to techniques that use explicit sentences in natural language to describe a conflict manifest. For example, A21 [75] explicitly uses natural language to document conflict scenarios. They present a web-based collaborative conflict lexicon in which a conflict is documented by providing a name, a description, the conflicting modifications, and the scenario leading to the conflict. A01 [47] has also explicitly defines conflicts by describing problematic relations among a set of conflicting operations.
- *Implicit Spec* refers to approaches where the conflicts are self-defined using implicit relationships on conflict detection and prevention techniques. For instance, A02 [48] implicitly describes conflicts by specifying a validation tree, while A28 [93] uses the comments that are wrapped around the conflict detection rules.

4.1.2 Conflict prevention

This concept focuses on the techniques used to avoid the creation of conflict situations or to prevent the occurrence of conflict in the merging of model

versions. We define the following attributes to categorize conflict prevention techniques identified in the investigated articles.

- *Lock-based* refers to techniques that allow fine-grained locks to restrict modifications of a model by users, when necessary. For example, A07 [57] defines a property-based locking in which users must lock certain properties of the model to modify them.
- *Conflict-free algorithm* refers to techniques that use specific algorithms to ensure that the result of the merge process will be without any conflict. Two sample techniques employed for this attribute are using conflict recognition rules in the merge construction and profiting from conflict-free replicated data types (CRDTs) [153] to ensure that the shared model is identical. For instance, A11 [64] proposes consistency-preserving edit steps to perform the merge process for new modifications using a transformation chain, while A39 [110] realizes the consistency of models by CRDT commands represented for each type of model element modification.
- *Operation ordering* refers to techniques that check a precedence relation between change operations to ensure that applying two operations are not conflictual in the current change sequence. As an example, A36 [105] uses a hash function to define dependencies between operations. Then for each conflicting pair of operations, they investigate the precedent relation to determine which operation must be executed first. Moreover, A22 [77] uses the NSGA-II algorithm for maximizing the number of detected critical pairs in a sequence of operations, which leads to minimizing the number of conflicts.
- *Sync modifications* refers to techniques that will synchronize and equate different modifications applied to the same model elements by different users before starting the merge process. As an illustration, A66 [145] uses the local checking idea to ensure global consistency using a stepwise local models repairing that synchronizes the overlaps on the corresponding elements.
- *Change awareness* refers to techniques that ensure every collaborator is aware of what the other collaborators are currently working on by propagating and highlighting changes. This way helps to avoid or minimize conflictual situations. For example, A52 [123] presents a highlight propagation algorithm to specify the model elements that are impacted by the modifications of other users.

4.1.3 Conflict detection

This concept focuses on the techniques used to discover the consistency violation or to detect the conflict situation arisen in the model merging process. We define the following attributes to categorize conflict detection techniques identified in the investigated articles.

- *Pattern matching* refers to techniques that search for the presence of a set of model elements as a conflict pattern in the merge process. We also

consider graph matching as a special form of pattern matching in which a conflict is the mismatch of graphs. For example, A15 [68] checks the input models for conflictual conditions using the user-defined patterns written in EVL [155]. A46 [117] serializes model specifications into RDF graphs, then detects conflicts by checking the RDF subgraphs that violate the homomorphisms graph, while A69 [149] defines a dependency relation for model elements that enables the detection of conflicts using incremental graph pattern matching techniques.
- *Constraint violation* refers to techniques that detect conflict by checking the validity and conformance of merged models, based on well-formedness rules for models and metamodels. For example, A04 [53] presents the proactive conflict detection by continuously analyzing the change operations based on the consistency rules defined by the metamodel or system constraints. A24 [84] merges two modifications then checks the result against pre-defined graph constraints, while A61 [138] systematically reuses consistency checking rules that are written in OCL [154] for a specific context.
- *Change overlapping* refers to techniques that discover conflicts by checking for contradicting changes that modify a model element in different ways or unknown equivalent changes that may be applied twice in the merged model. For instance, A08 [59] uses a conflict matrix to specify conditions in which change operations that modify corresponding model elements are conflicting. Furthermore, A27 [90] utilizes a topological sorting strategy to detect concurrent operations that violate their mutual intentions, while A47 [118] presents a critical pair analysis in which a pair of operations that contains a glue element is checked to see whether or not they can be combined into a single operation.
- *Formal methods* refers to techniques that discover conflicts based on mathematics and formal logic. For this, a model checker tool may be used to analyze the satisfaction of the logical formula in models. For instance, A09 [62] proposes an approach based on description logics to automate conflict detection using logical inference techniques. A13 [66] uses Alloy formal definitions to identify conflict based on first-order relational logic, while A68 [148] uses model checking using temporal properties expressed in CTL to detect conflicts.

4.1.4 Conflict resolution

This concept focuses on resolving the conflict during the merging process or fixing inconsistency in the merged model. We categorized conflict resolution techniques identified in the investigated articles using two attribute sets that refer to the method and time of conflict resolution. In the following, we separately describe the attributes for each category. To classify the conflict resolution method, we define the following attributes:

- *Manual* refers to the method that completely delegate the resolution phase to the users, once a conflict is detected. For example, A01 [46] shows the

conflicting operations to the user to manually reject or resolve them, while A61 [138] aborts the modifications and expects the user resolve them analogous to other versions.
- *Semi-automatic* refers to techniques requiring interaction with the user for further information in the resolution phase while trying to solve conflicts automatically. For instance, A07 [58] computes possible resolution candidates for each conflict and user must select the most suitable one, while A27 [91] follows a conflict resolution process based on the negotiation and voting among collaborators to identify their preferred modifications.
- *Automatic* refers to techniques that fully automate the resolution phase by performing operational transformation or applying a general rule to serialize change operations. For instance, A32 [99] uses the operational transformation algorithm for each conflict to produce new operations to be applied consistently on all versions. A37 [108] applies pre-defined conflict resolution patterns, while A69 [149] runs the previously introduced TGG rules based on three specified conflict resolution strategies.

To classify the conflict resolution time, we define the following attributes:

- *On-the-fly* refers to the techniques that resolve conflicts during the merge process. As an example, A30 [96] displays four resolution strategies for each conflict, and the merge process waits for the user to select one of them.
- *Post-analysis* refers to the techniques that postpone the resolution phase to the end of the merge process. As an illustration, A35 [104] creates an initial merged model by ignoring the conflicting operations. Then, after the merge process, the user should decide and resolve each conflict.

4.1.5 Conflict awareness

This concept focuses on the techniques used to warn the user of the conflict situation or conflict occurrence in the model merging process. We define the following attributes to categorize conflict awareness techniques identified in the investigated articles.

- *Notification* refers to the techniques that use an alarm (message or sound) to let the user know that a conflict has happened or to prevent a conflict by attracting the user's attention to new changes. For example, A04 [53] notifies team members by a PCD notification when a conflict occurs, while A55 [127] shows warning messages within the modeling tool. A06 [56] also displays a conflict box in the graphical modeling editor for each conflicting change to receive prompt action from the user.
- *Change highlighting* refers to the techniques that identify the involved model elements in a conflict using color highlighting or flags. For instance, A10 [63] marks all involved model elements in all other views when a user selects a conflict item. A22 [80] presents a UML profile to visualize merge conflicts in the concrete syntax of the UML model using coloring techniques and conflict flags, while A35 [104] highlights conflicting changes in a different color instead of non-conflicting ones.

– *Conflict report* refers to the techniques that present a separate textual or graphical panel to visualize or represent the detected conflicts in a user-friendly format. For example, A09 [62] shows a textual inconsistency warning based on the reasoner report, while A25 [85] displays a list of detected conflicts in the outlines. Moreover, A29 [95] presents a conflict view to display conflicts in a new window with more details, including the description and link to the conflicting element.

4.2 Systematic map

Having a taxonomy and classification scheme, we can enter the information of all articles into the scheme to explore patterns and show the frequencies of publications in a certain category. A systematic map is a tool used to achieve this goal. It helps to analyze the intersection of the facets with the high-level attributes that represent a set of articles. At this step, we considered the concept of *contribution facet* and *research facet* based on the motivated guideline by Petersen et al. [33]. The *conflict management facet* was obtained from our classification scheme to understand the contribution and research value of the final pool of articles in our study. Fig. 9 shows the resulting systematic map, which represents the intersection of contribution and research facets with the conflict management facet in different quadrants of a bubble plot. The size of a bubble shows the number of articles in the corresponding pair. Note that one article may contribute to more than one pair.

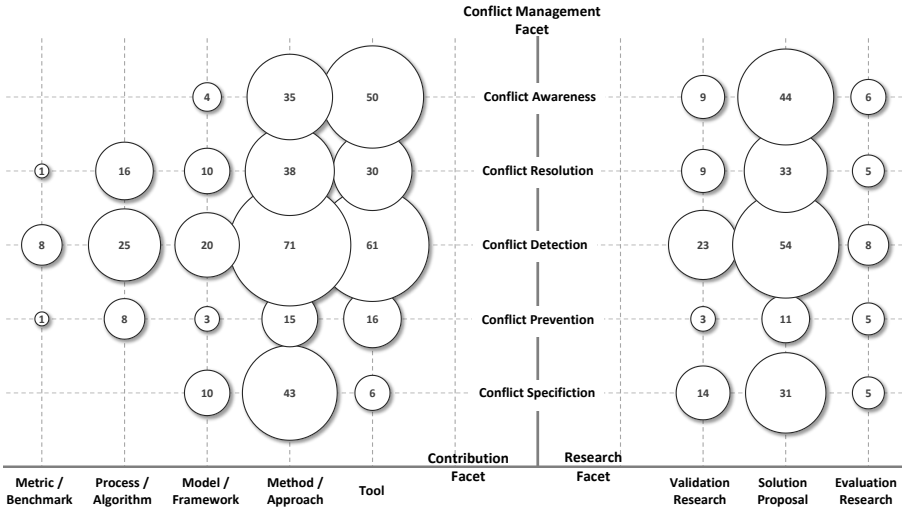


Fig. 9 Systematic Map - Association of contribution/research facets with the conflict management facet

5 Results

In this section, we report the results of our systematic review. It starts with the representation of resulted classification for approaches. Then, we evaluate the extracted map by addressing the research questions related to goals one to three. Finally, we present the reported limitations and future directions to answer the fourth goal's research questions.

5.1 Classification of approaches

In the last step of the conducting phase in our research protocol, we established the collected attributes and proposed a taxonomy and classification scheme for organizing the conflict management approaches in a structured manner. When a random sample of studies was assessed and their classification scheme was determined, the first author read the complete full-text of the final pool of articles to extract their data according to the classification. Since there are multiple articles in our final pool that present the same conflict management approach with different levels of details, we decided to summarize such related articles as a single approach and classify the approach instead of articles. At this step, the attributes assigned to each approach are obtained using two main rules: (i) using the union of data extracted from all articles that represent the approach in cumulative attributes such as the *"Conflict Detection"* technique, and (ii) keeping the latest data for singular attributes such as the *"Conflict Resolution"* method.

The summarization of the 105 final pool articles results in 69 approaches. The first author performed the classification and assigned the attributes to the identified approaches, and other authors reviewed the results to make sure that the classification was done correctly. The classification results are presented in Tables 4 to 6 of Appendix A, which show the frequency of techniques that are proposed by the investigated conflict management approaches. The classification results also reported the supported conflict types and the frequency of collaborative modeling characteristics assigned to each approach in Tables 7 to 9 of Appendix A.

According to the extracted taxonomy (Figure 8), a *valid* conflict management approach must support all mandatory aspects, i. e., conflict specification, conflict detection, and conflict resolution. Table 3 represents the distribution of supported conflict management aspects for the investigated approaches. Our results indicate that 57 of 69 (89.6 %) approaches provide valid conflict management approaches, in which only five approaches support all of the mandatory and optional aspects of conflict management. Furthermore, most of the approaches (42, or 60.9 %) cover all aspects except conflict prevention. We also found that the remaining 12 approaches have invalid configuration based on the proposed taxonomy. These approaches did not provide valid conflict management support and only focused on optional aspects. However, they can complement other conflict management approaches to enhance optional

Table 3 Distribution of supported conflict management aspects in the investigated studies

Supported Aspects	#APRs	Approaches	Valid CM
Specification, *Prevention*, *Detection*, *Resolution*, *Awareness*	5	A15, A24, A32, A39, A55	Yes
Specification, *Prevention*, *Detection*, *Resolution*	1	A07	Yes
Specification, *Detection*, *Resolution*, *Awareness*	42	A01, A02, A03, A04, A05, A06, A08, A09, A10, A12, A16, A18, A22, A23, A25, A26, A27, A28, A29, A30, A33, A34, A35, A38, A42, A43, A44, A45, A46, A50, A51, A53, A54, A56, A57, A60, A61, A63, A64, A65, A67, A68	Yes
Specification, *Detection*, *Resolution*	9	A13, A17, A31, A37, A47, A49, A59, A62, A69	Yes
Specification, *Prevention*	6	A11, A20, A36, A52, A58, A66	No
Prevention	3	A19, A40, A48	No
Specification	3	A14, A21, A41	No

aspects such as conflict specification or conflict prevention. Although those 12 approaches cannot be described as a conflict management approach, we still considered them in the reported results because they provided valuable techniques in this area.

5.2 Mapping evaluation

We analyzed the extracted attributes to yield a map containing different aspects of model merging conflict management. This section evaluates the generated map by addressing research questions Q1.*, Q2.*, and Q3.*.

Q1.1. *Which conflict management facets are being employed for model merging?* The question aims to understand which type of activities are being employed to manage conflicts in model merging. To address this question, we have developed a taxonomy based on our findings as well as the body of knowledge in model merging conflict management, e. g., [13, 15], and [24]. We captured key techniques and activities and characterized them into the appropriate categories. The extracted taxonomy is shown in Fig. 8. The first layer of the taxonomy is composed of the following five categories:

– *Conflict specification* refers to the activities that describe a conflict situation in the merge process.

- *Conflict prevention* refers to the activities that avoid or prevent the conflict occurrence before or during the merge process.
- *Conflict detection* refers to the activities in which potential conflicts during the merge process are discovered.
- *Conflict resolution* refers to the activities by which the detected conflicts are fixed during or after the merge process.
- *Conflict awareness* refers to the activities that warn users of potential conflicts before the merge process or display the detected conflicts during and after the merge process.

Fig. 10 depicts the number of articles by conflict management facets with regard to the main focus of proposed techniques in the final pool articles. As shown in Fig. 10, 47.6 % (50 of 105) of the articles are dedicated to conflict specification, while 18.1 % (19 of 105) of the articles address conflict prevention. Most of the final pool articles, 80.9 % (85 of 105), focus on conflict detection. Finally, 44.8 % (47 of 105) of the articles address conflict resolution, and 56.2 % (59 of 105) belong to conflict awareness.

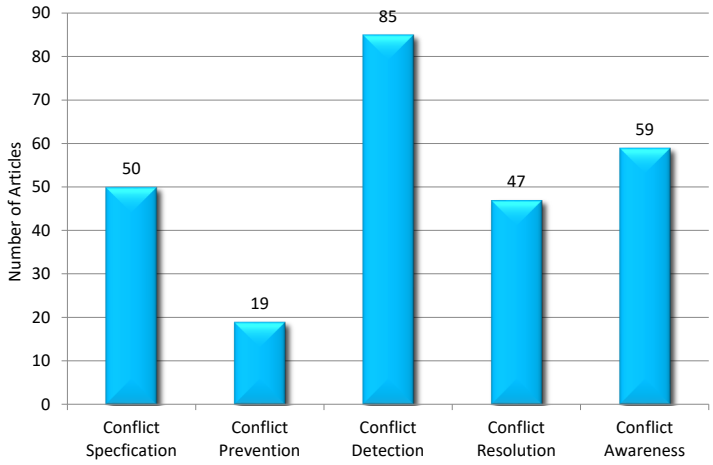


Fig. 10 Data for Q1.1: Article by conflict management facets

Moreover, Fig. 11 shows the number of approaches focused on each conflict management facet. We can see that conflict detection with 82.6 % (57 of 69) of the approaches is the most considered conflict management facet. Even though conflict prevention with the appearance in 21.7 % (15 of 69) approaches is not widely supported yet, the (semi-)automatic conflict resolution techniques provided by 46.4 % (32 of 69) of the analyzed approaches. According to the extracted data, 59.4 % (47 of 69) of the approaches use an explicit conflict specification method. Furthermore, 68.1 % (47 of 69) of the approaches are endowed with conflict awareness mechanisms. Indeed, further existing conflict management approaches are typically focused on finding con-

flicts and informing users instead of avoiding or automatically resolving them. In Q3.*, we focus on the obtained data for each category.

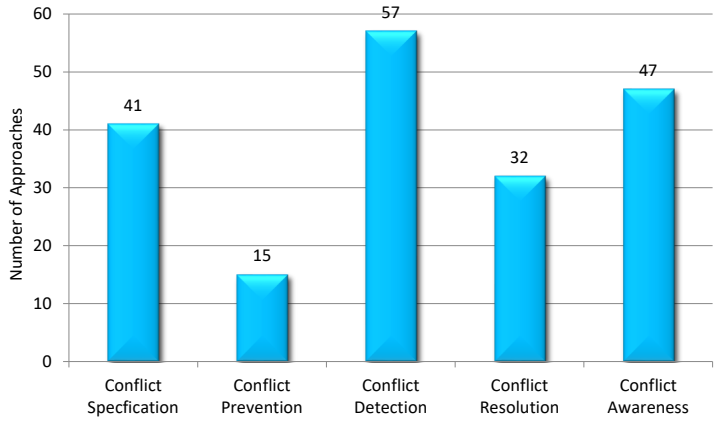


Fig. 11 Data for Q1.1: Approaches by conflict management facets

Q1.2. *Which is the contribution facet of the articles?* This question relates to the contribution side of our systematic map, which broadly categorizes the novel propositions of the articles based on the contribution facet, including conflict management tool, method/approach, model/framework, or metric/benchmark. This provides an overview of the current state-of-the-art and practice in the area of model merging conflict management for researchers and industrial practitioners. Fig. 12 shows the contribution facet distribution for all 105 articles of our final pool. Most of the articles (96 articles, 91.4 %) contribute to developing a method/approach, and 78 articles (74.3 %) provide tool support for their proposal. Furthermore, only 31 articles (29.5 %) suggest a model/framework for conflict management, and 37 articles (35.2 %) follow a process/algorithm to manage conflicts. Few articles (9 articles, 8.6 %) have proposed metrics or benchmarks for evaluating conflict management techniques.

Fig. 13 shows the annual distribution of the contribution facet for the proposed techniques and tools in each final pool article. This shows that every year in the period from 2005 to 2021, articles not only propose at least one novel method/approach for conflict management but also they support their idea with tool implementation. The number of articles increased during the period from 2009 to 2010, which has been accompanied by the implementation of significant projects in conflict management, such as AMOR project[7]. In contrast, from 2011 to 2021, many novel research efforts have been presented in various categories, highlighting the interest of researchers in conflict management.

Q1.3. *Which is the research facet of the articles?* This question relates to the research side of our systematic map, which broadly categorizes the

[7] `http://www.modelversioning.org/`

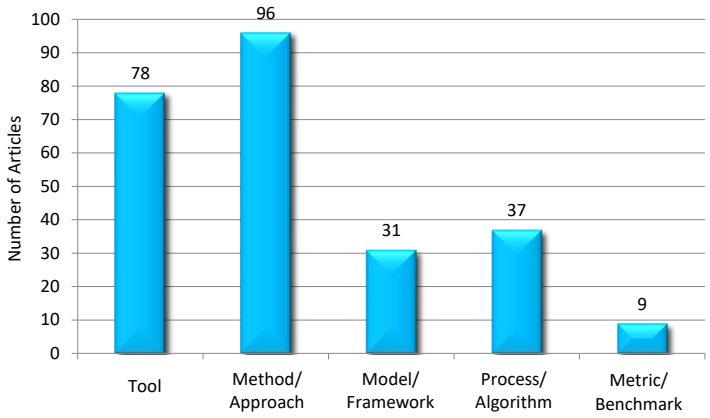


Fig. 12 Data for Q1.2: Article by contribution facet

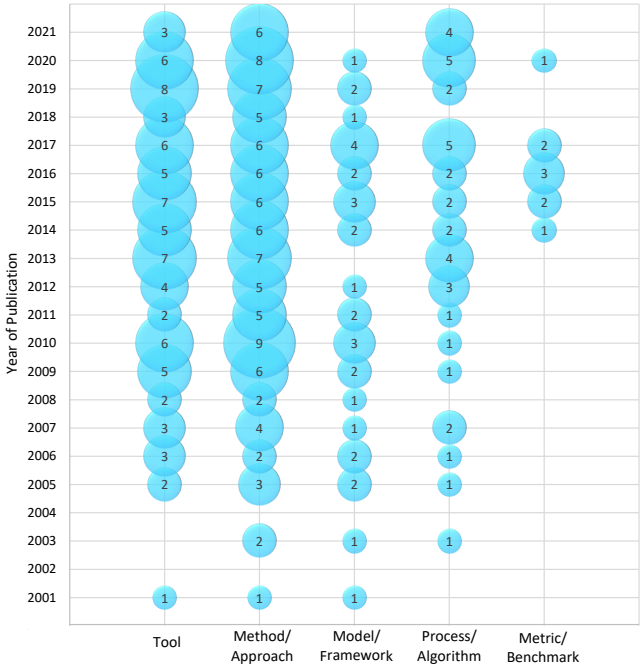


Fig. 13 Data for Q1.2: Number of articles for contribution facet by year

nature of articles based on their purpose of conducting the research, including validation research, solution proposal, or evaluation research. This provides an overview of the nature of research presented in the area of model merging conflict management. Note that each article is placed in one category. Fig. 14 shows the research facet distribution for all 105 articles of our final pool. Most of the articles (67 articles, 63.8 %) have proposed solutions with the help of examples or prototyping, whereas 26 articles (24.8 %) are validation

research, and only 12 articles (11.4 %) have used an empirical evaluation for their proposals.

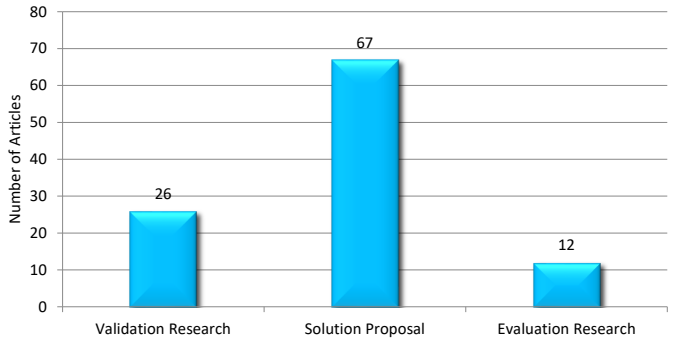


Fig. 14 Data for Q1.3: Article by research facet

Fig. 15 shows the annual distribution of the research facet for all 105 articles of our final pool. This shows that every year during the period 2005 to 2021, most articles proposed solutions. The largest number of validation research articles were published in 2010 and 2011. In contrast, the number of articles that present evaluation research has increased after 2013, showing the rising interest in proposing novel techniques with empirical experiments.

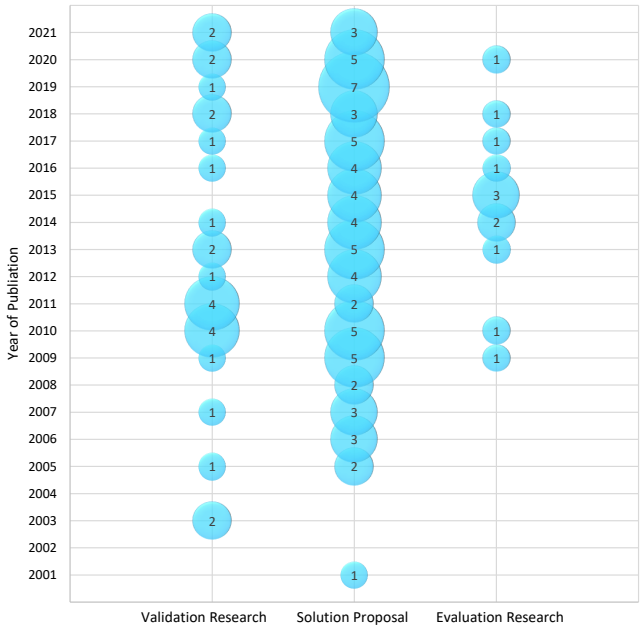


Fig. 15 Data for Q1.3: Number of articles for research facet by year

Q1.4. *Which is the tool support maturity level for available conflict management techniques in the area?* To answer this question, we investigate the maturity level of tool support for the proposed approach in each final pool article based on the guideline motivated by Cuadros López et al. [34]. Fig. 16 provides an overview of the maturity level for 105 articles. This shows that for most of the articles (73 of 105, 69.5 %), conflict management techniques are partially implemented. Only nine (8.6 %) of the articles demonstrate the complete implementation of the approach without showing their usefulness in the industry, and even less (3 of 105, 2.9 %) have addressed an empirical evaluation to investigate the applicability of the proposed techniques in the industry. In contrast, the level of maturity for 20 (19 %) of the articles is "not implemented". Overall, the result shows that most of the works only provide a prototype to validate their proposal and to further mature the conflict management field. Conducting validation in industry and more empirical experiments are needed.

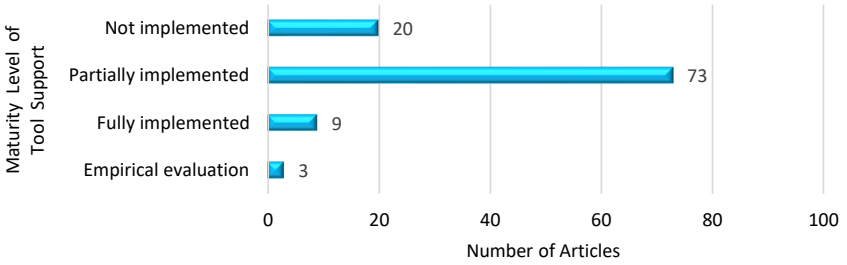


Fig. 16 Data for Q1.4: Maturity level of CM techniques

Q2.1. *Which kind of merging conflict is mostly supported in the area?* To address this question, we decided to investigate 69 approaches instead of 105 final pool articles. This provides accurate information on solutions that are offered to support merging conflicts. Fig. 17 presents the percentage of the approaches that can cover each kind of merging conflict. Our results show that in the existing approaches, 55.1 % (38 of 69) support only syntactic conflicts, and 7.2 % (5 of 69) can only manage several conflicts that are arisen for semantical reasons. Also, 37.7 % (26 of 69) of the approaches are applied for both syntactic and semantic conflicts.

Fig. 18 shows the distribution of approaches that support different types of semantic conflict. It illustrates that from the approaches supporting semantic conflicts, 36.2 % (25 of 69) of the approaches are applied for static semantic conflicts, followed by 14.5 % (10 of 69) approaches proposing semantic equivalence conflicts, and only four (5.8 %) approaches focusing on behavioral semantic conflicts, which indicates the difficulty of investigating this type of conflict.

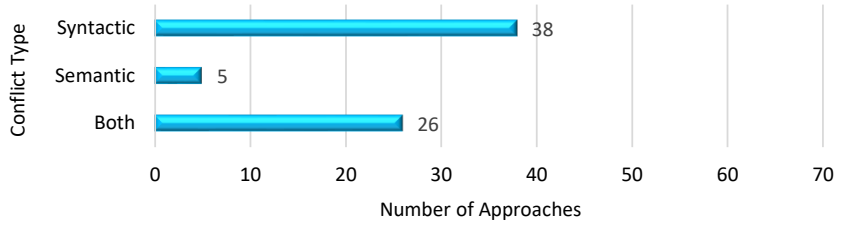


Fig. 17 Data for Q2.1: Number of conflict types by approaches

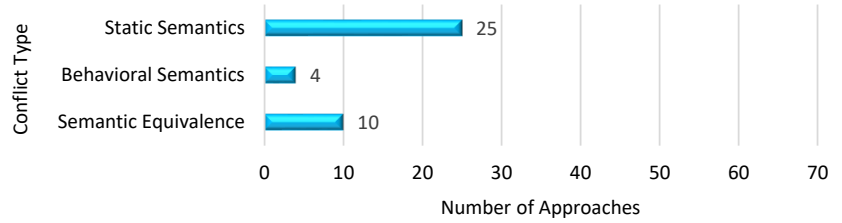


Fig. 18 Data for Q2.1: Number of semantic conflict types by approaches

Q2.2. *Which collaborative modeling characteristics are crucial to the area?* To answer this question, we investigated conflict management approaches to identify the attributes of the main characteristics of collaborative modeling that are significantly important for designing the conflict management techniques, including: *model type*, *repository architecture*, *versioning strategy*, *collaboration mechanism*, *comparison technique*, *merging technique*, and *ordered feature support*. Fig. 19 presents the percentage of collaborative modeling characteristics for 69 approaches.

The results show that UML and EMF-based models are two widely supported model types in conflict management techniques that respectively is used in 28 (40.6 %) and 20 (29.0 %) of the approaches. They are followed by 15 (21.7 %) of the approaches that are applied for any independent model type, and five (7.2 %) of the approaches that focus on workflow models. Furthermore, 13 (18.8 %) of the approaches are assigned to other model types, such as architectural models that are targeted in A05 [54] and XML schema in A29 [95]. Note that most existing approaches have not presented information about the facets of structural or behavioral in the supported modeling languages. However, during the analysis of the obtained data, we noticed that the UML class diagram is the most considered structural modeling language. Moreover, a few conflict management approaches are designed to support behavioral modeling languages such as BPMN (e. g., A06 [56], A08 [60], and A30 [97]), UML state machine (e. g., A24 [83], A60 [137], and A68 [148]), and UML sequence diagram (e. g., A22 [80] and A45 [116]).

In our analysis, we also found that most of the existing conflict management techniques use a centralized repository architecture (60, i. e.87.0 % of the approaches) instead of distributed storage (nine, i. e.13.0 % of the approaches).

Our results show that 64 (92.8 %) of the approaches proposed techniques based only on the optimistic versioning paradigm, and four (5.8 %) of them worked based only on the pessimistic paradigm. Finally, only one approach (1.4 %), A07 [57,58], proposes conflict management techniques based on both model versioning strategies.

As shown in Fig. 19, the existing approaches mostly contributed to conflict management techniques based only on one collaboration mechanism. For 54 (78.3 %) of the approaches, offline collaboration is supported, and for ten (14.5 %) of the approaches, online collaboration is used. Interestingly, five (7.2 %) approaches support conflict management techniques for both offline and online collaborations. Out of these five approaches, three approaches focus on the conflict specification (A14 [67], A21 [75], and A41 [112]), one approach is dedicated to conflict prevention (A19 [72]), and one approach supports conflict detection and resolution (A07 [57]). Moreover, we found that 33 (47.8 %) of the approaches only track changes with operation-based techniques, 30 (43.5 %) of the approaches use only state-based comparison techniques. It is also worth mentioning that two (2.9 %) approaches can detect changes with both operation-based and state-based techniques. Furthermore, four (5.8 %) approaches use a hybrid comparison technique to identify changes by following state-based and operation-based manners.

The merging techniques in our results show that the three-way merging is becoming increasingly important for conflict management, as 33 (47.8 %) of the approaches are applied to this type of merging. They are followed by 29 (42.0 %) of the approaches that have used two-way merging, and only seven (10.1 %) of the approaches have proposed conflict management techniques based on the raw merging, while interestingly all of them are operation-based approaches. Finally, concerning the support of ordered feature attribute, we found that only 14 (20.3 %) of the approaches have supported conflict management techniques for multi-valued elements.

Fig. 20 shows the distribution of conflict management techniques for the investigated attributes of collaborative modeling. In this figure, the size of a bubble represents the number of approaches that support a conflict management technique based on the corresponding collaborative modeling attribute in the pair. For instance, the conflict detection techniques have been mostly proposed for collaborative modeling environments with *centeralized* repository, *optimistic* versioning strategy, or *offline* collaboration mechanism. Furthermore, UML is the most considered modeling language in conflict detection techniques: 23 of 69 approaches support conflict detection for UML models.

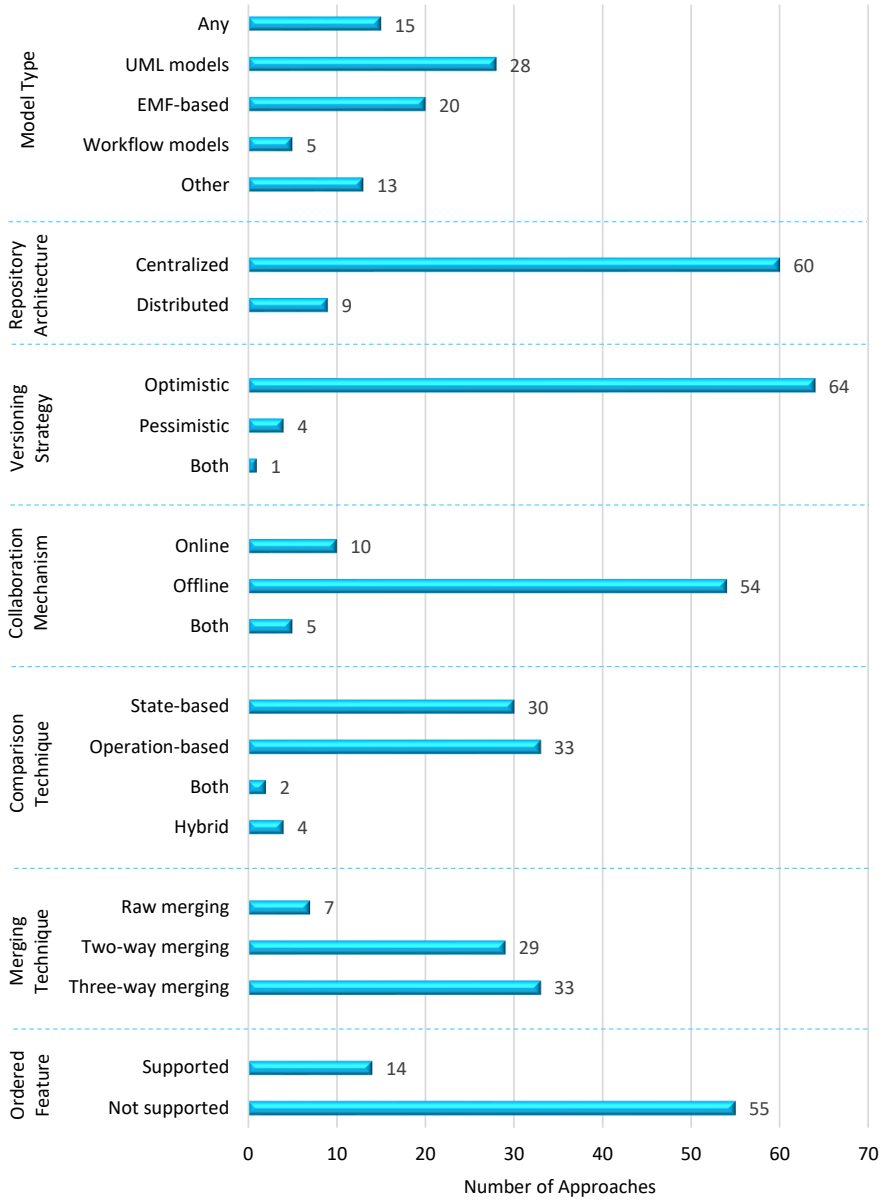


Fig. 19 Data for Q2.2: Classification of collaborative modeling characteristics in conflict management techniques

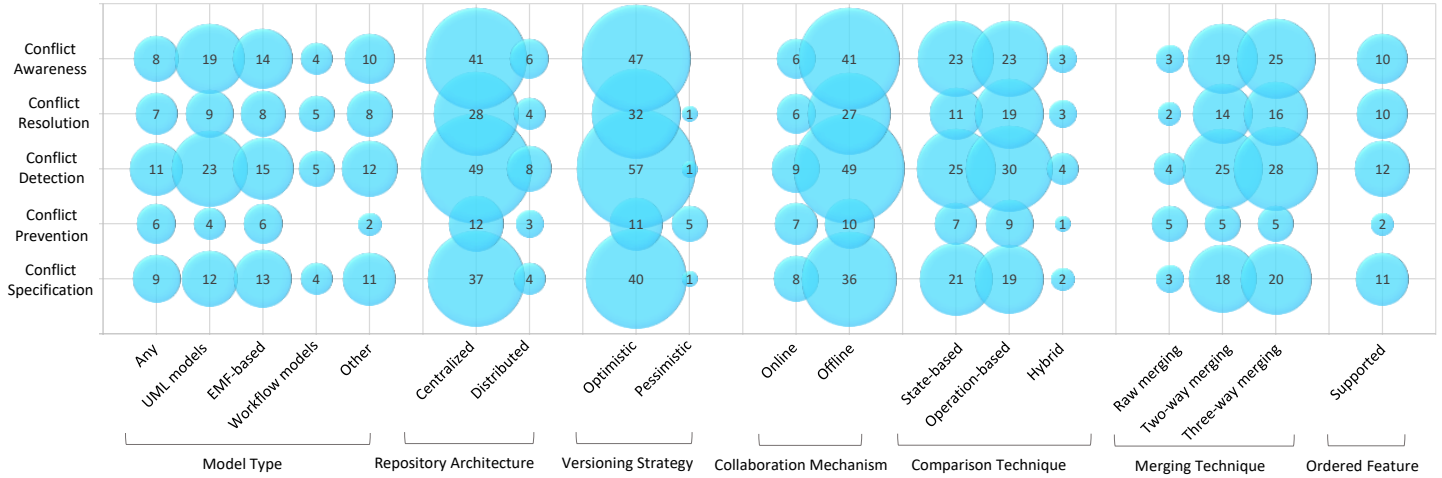


Fig. 20 Data for Q2.2: Number of approaches for Collaborative modeling characteristics by conflict management facet

Q3.1. *Which type of conflict specification techniques is employed for model merging?* To answer this question, we refer to the conflict specification category of extracted taxonomy shown in Fig. 8. As described in Section 4.1.1, we organized the techniques for the specification of merging conflict into four subbranches including *Formalism*, *Pattern-based*, *Natural Language*, and *Implicit Spec*. Pattern-based techniques may describe conflicts as *Conflict models* or *Specification patterns*, which use model-based and template-based methods, respectively. Fig. 21 shows that conflicts are implicitly specified in 25 of the 69 approaches investigated in our study. According to the extracted results, 17 of the approaches use natural language to describe merging conflicts, and 18 approaches employ specification techniques based on a given formalism. Moreover, a few investigated approaches (10 approaches) have used pattern-based techniques to specify conflict. Note that four approaches (A30, A41, A50, and A61) have in parallel employed two conflict specification techniques.

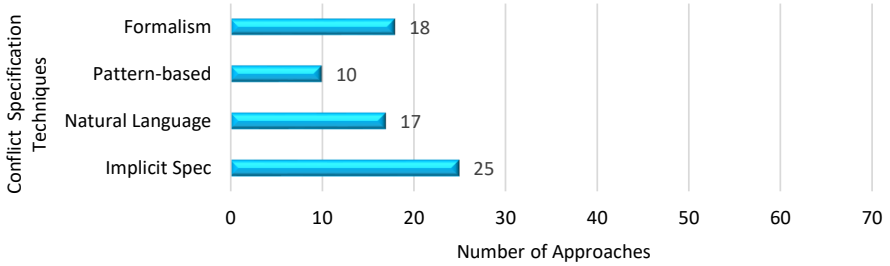


Fig. 21 Data for Q3.1: Conflict specification techniques

Q3.2. *Which conflict prevention approaches are being used by researchers in the area?* The obtained results from the classification scheme in Section 4.1.2 show that conflict prevention techniques can be divided into five subbranches, including *Conflict-Free Algorithm*, *Operation Ordering*, *Lock-based*, *Sync Modifications*, and *Change Awareness* techniques. As shown in the taxonomy of conflict management techniques, Fig. 8, conflict-free algorithms comprise two methods, namely, *Merge Construction* and *Data Replication*. The conflict-free merge algorithm avoids conflict by the specific rules in the merge process, and conflict-free replication avoids conflict by replicating data types for the shared models. Moreover, *Precedence Relation* and *Heuristic Search* are two primary methods for preventing merging conflicts based on the operation ordering technique. As shown in Fig. 22, five of 69 approaches use pessimistic *Lock-based* methods to prevent conflicts. Moreover, four of the approaches propose a *Conflict-free Algorithm* to support conflict prevention in the merge process. Three of the approaches present *Operation Ordering* based on a change serialization. Finally, each *Sync Modifications* and *Change Awareness* technique is applied in only two approaches to avoid conflicts by determining a change synchronization, or highlighting the changes in a real-time propagation.

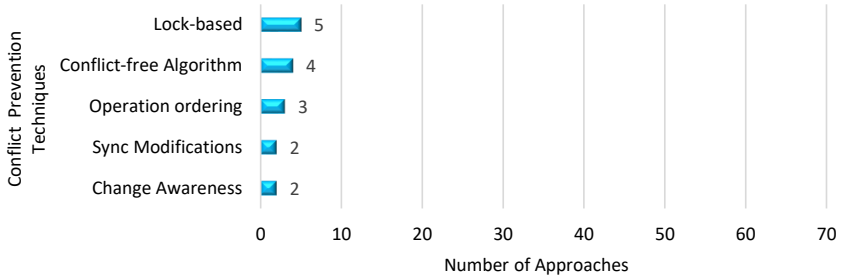


Fig. 22 Data for Q3.2: Conflict prevention techniques

Q3.3. *Which are the most used approaches for conflict detection by academic and industrial researchers?* Our findings in Section 4.1.3 show that *Constraint Violation*, *Pattern Matching*, *Change Overlapping*, and *Formal Methods* are the common categories of conflict detection techniques. The conflict detection subbranch in the extracted taxonomy, Fig. 8, shows that every approach may simultaneously benefit more than one technique to detect conflicts. Moreover, the constraint violation techniques can check *Model* and *Metamodel* constraints, whereas the change overlapping techniques comprise *Contradicting Changes* and *Equivalent Changes*. As Fig. 23 shows, constraint violation is the most used technique to detect a conflict, which is adopted by 23 of the existing approaches. Change overlapping is the second most popular conflict detection technique applied by 20 of the approaches. They are followed by pattern matching and formal methods techniques that are used in 15 and 13 of the approaches, respectively. More surprisingly and apart from Fig. 23, 12 (17.4 %) of existing conflict detection approaches are implemented and evaluated by the industry, which shows that more research is needed to further mature the conflict detection area.

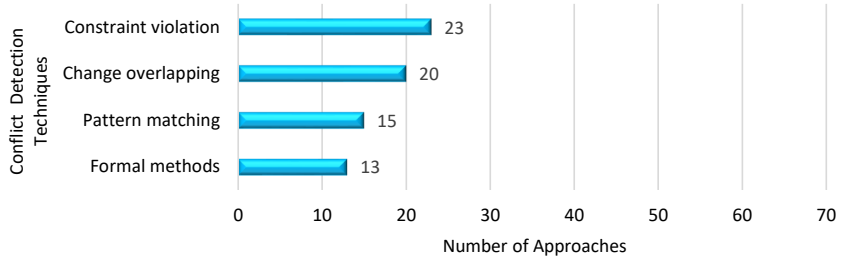


Fig. 23 Data for Q3.3: Conflict detection techniques

Fig. 24 shows the distribution of conflict detection techniques based on the conflict type. According to the result, each conflict detection technique is mostly used to discover syntactic conflicts, and among them, constraint violation, used in 20 approaches, is the most common technique. Interestingly,

semantic conflicts are mostly detected by constraint violation technique, while change overlapping and pattern matching are commonly used as next technique. More specifically, behavioral semantic conflicts are mostly detected by constraint violation techniques, and semantic equivalent conflicts are mostly discovered with change overlapping and pattern matching techniques. However, the number of approaches that use each conflict detection technique is almost the same for each conflict type.

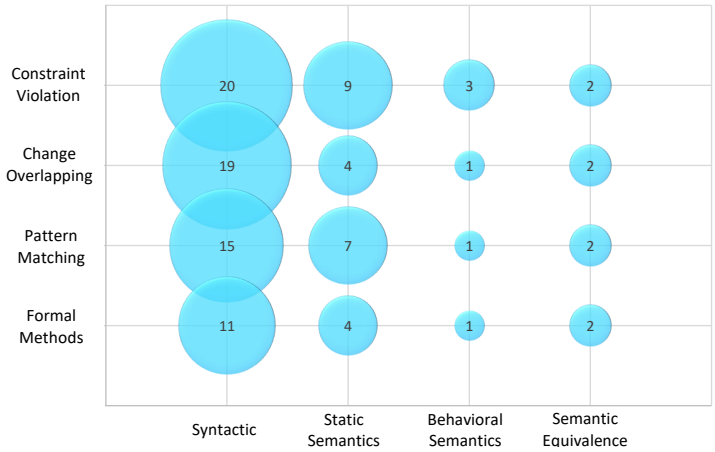


Fig. 24 Number of approaches for Conflict Detection techniques by conflict types

Q3.4. *Which kind of conflict resolution mechanism is widely used?* Our study results show that the *Resolution Method* and the *Resolution Time* are key attributes to categorize conflict resolution mechanisms. The resolution method indicates user involvement in the resolution phase, which is divided into *Manual*, *Semi-automatic*, and *Automatic*. Fig. 25 shows that 14 of the approaches provide an automatic mechanism that solves the conflict without involving the user in the resolution phase. Moreover, 18 of the approaches propose a semi-automatic mechanism that involves users only to deal with discordant changes, and 25 of the approaches completely leave the reconciliation of conflict to the users.

Similarly, the resolution time specifies when the resolution phase can be executed in the merge process. As already discussed in Section 4.1.4, we identified *On-the-fly* and *Post-analysis* as two attributes for this category. Fig. 26 shows that more than half of the conflict resolution approaches (35) propose on-the-fly resolution techniques, whereas 22 approaches rely on post-analysis techniques.

Note that we can only determine resolution method and resolution time for approaches that have proposed conflict resolution or at least conflict detection techniques. Thereby, Fig. 25 and Fig. 26 show that conflict resolution mech-

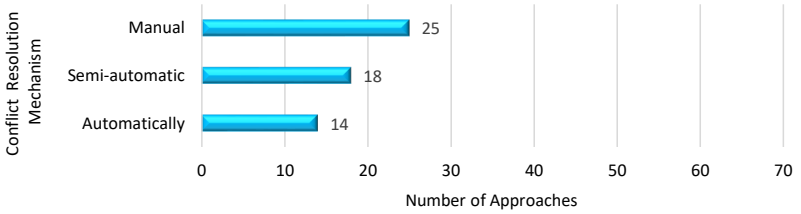


Fig. 25 Data for Q3.4: Conflict resolution method

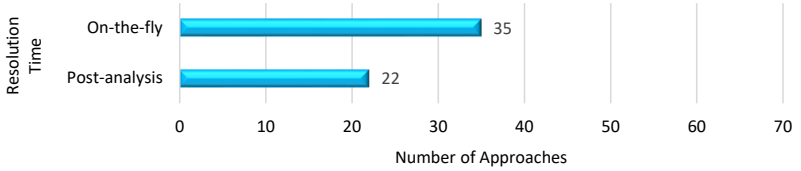


Fig. 26 Data for Q3.4: Conflict resolution time

anisms and resolution time are not applicable for 15.9 % of the approaches. Moreover, the subbranch of conflict resolution in the taxonomy of conflict management techniques (see Fig. 8) indicates that for both resolution method and resolution time categories, only one of the mentioned values should be selected for each resolution approach.

Q3.5. *Which conflict awareness methods are being used in the area?* Our classification results in Section 4.1.5 show that *Notification*, *Conflict Highlighting*, and *Conflict Report* are common categories for conflict awareness techniques. The taxonomy for conflict management techniques (Fig. 8) shows that the notification category comprises *warning* and *prompt action* methods. Warning refers to a message that alerts the user of a conflict condition and prompts action, which encourages the user to handle a conflictual condition quickly. Also, we identified *textual list* and *graphical view* as two methods for conflict awareness as conflict report. Textual list provides a text-based list of detected conflicts, and graphical view visualizes conflicts in a graphical editor with specific notations. Fig. 27 shows the composition of three identified categories for conflict awareness. We found that 18 of the investigated approaches provide a conflict report, while 13 approaches perform a highlighting method and 11 approaches warn the users with a notification method.

5.3 Map limitations and future directions

Systematic review articles usually state the limitations of approaches and recommend directions for continuing research in the area. In the following, we address the RQ4.* research questions set, which is concerned with classifying the reported limitations and future directions from the articles of our final pool.

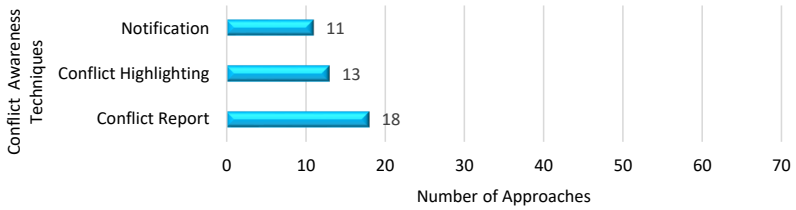


Fig. 27 Data for Q3.5: Conflict awareness techniques

Q4.1. *Which limitations have been reported?* To answer this question, we double-checked all final pool articles to gather the reported limitations. Unfortunately, only 29.5 % (31 of 105) of the articles discuss the shortcomings and the limitations of their work. Fig. 28 categorizes every article that may have reported one or more limitations. The figure shows that lack of proper *tool support* is the most commonly reported limitation, while several approaches are limited to a specific modeling language. Moreover, lack of practical analysis and weaknesses in various technicalities have also been mentioned in many research efforts. In the following, we describe each category of limitation.

- ***Tool support:*** The proposed technique has not been implemented, or the presented tools have some limitations. For example, A03 [49] proposes an operation-based conflict detection and resolution based on issue-based collaboration. However, there is not much support for building such a system. Moreover, the main limitation of A57 [134] is lack of the required translators or adapters to integrate the proposed comprehensive systems with heterogeneous modeling tools.

- ***Domain-specific:*** The techniques presented only guarantee the general level of EMF models and do not address additional domain-specific constraints. For instance, A42 [113] can only ensure the generic consistency control such as well-formedness conditions for EMF models regardless of the underlying Ecore model. As other examples, A55 [126] only considers constraints and rules for consistency between a single element type, or A65 [143] is limited to the homogeneous model since they cannot define the merging process at a notational level.

- ***Evaluation:*** Lack of real-world models and actual environment to conduct sufficient validation that needs more experiments and empirical evaluation. For example, A20 [73] reports a need for further empirical studies to deeply evaluate the performance of the proposed approach, or A28 [92] has specified that the expressiveness of the approach has to be analyzed for different modeling languages. Moreover, A04 [52] reports lack of real-world models and design tasks from an actual project.

- ***Visualization:*** Limitation on the graphical aspect and the need to visualize the conflict representation. For instance, A51 [122] reports lack

of proper support to visualize conflicts in the graphical modeling interfaces. Moreover, A22 [79] specifies the need to represent conflicts, which has paramount importance for usability reasons.

- ***Scalability:*** Lack of support for large models and limitation in the number of team members to manage collaborative modeling. For example, A01 [46] mentions that the use of a central controller had limited the scalability of the approach to managing large models and huge modifications. Furthermore, A04 [52] reports that their approach is limited to support collaboration for teams with a few participants.

- ***Functionality:*** Missing technical functionality and features in tools and technologies used, such as:
  - *Lack of support for all modeling language features reported by A22 [79]*;
  - *Limitation on the ability of model checker to detect all conflictual conditions reported by A32 [99]*;
  - *Lack of support for some change operators reported by A30 [97]*;
  - *Limitation on the confidence of compatibility between locks reported by A07 [57]*;
  - *Lack of support for cascading deletion in conflict resolution reported by A52 [123].*

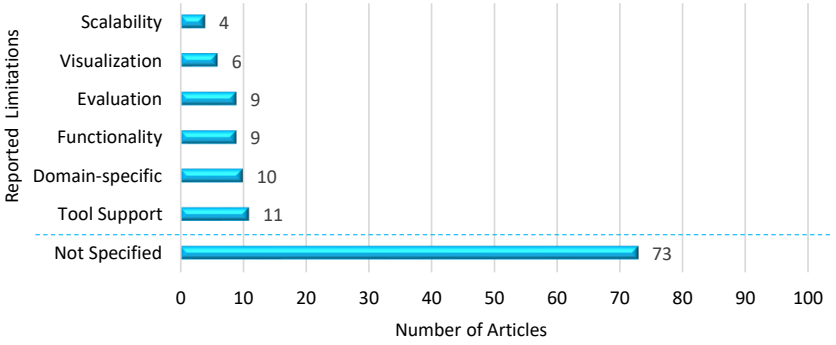


Fig. 28 Data for Q4.1: Reported limitations

Q4.2. *Which are the trends in the area?* To answer this question, we consider the results of our review for approaches that have been proposed in the last 5.5 years (i. e., between 2016 to mid-2021), which includes 28 of 69 approaches. According to the result, trends in conflict management approaches suggest the preparation of the following features:

- ***Scalability:*** Nine out of ten approaches (90.0 %) supporting scalability in the number of team members and size of model elements were published between 2016 and 2021.

- ***Distributed repository:*** Seven out of nine approaches (77.8 %) with distributed repositories have been proposed after 2015.
- ***Online collaboration:*** 11 out of 15 approaches (73.3 %) that support conflict management in online collaborative modeling have been proposed after 2015.
- ***N-way merging:*** Only 18 approaches can manage conflicts in the merge of N models and 13 of them (72.2 %) were presented after 2016.
- ***Automatic conflict resolution:*** Since 2017, six approaches to automatically resolve the conflict have been proposed. While overall, we only found 13 approaches for automatic conflict resolution.
- ***Web-based client:*** Five out of six approaches (83.3 %) with a web-based client have been proposed after 2017.
- ***Conflict prevention:*** 11 out of 16 approaches (68.8 %) to prevent conflict before the merge process have been proposed after 2016.

Q4.3. *Which future research directions are being suggested?* To identify research directions provided by researchers in the articles, we collected all the information and discussion for future work. Out of the 105 articles, 95 articles provided guidance for future enhancements. Fig. 29 shows the future directions that were extracted from our final pool articles. We can observe that improving *evaluation* is reported by 33.3 % of the articles as the most common future directions. It is followed by extending implementation and providing proper *tool support* (32.4 %), guidelines for automating *conflict resolution* (23.8 %), *visualizing* conflict management activities (13.3 %), and handling *semantic conflicts* (13.3 %) as popular future directions. We categorized the reported future directions as follows:

- ***Evaluation:*** Prove the correctness of the proposed approach and perform more experiments and evaluations to assess its effectiveness and usability.
- ***Tool support:*** Address implementation of algorithms and proposals or extend and improve the current implementation to the need for tool support.
- ***Conflict resolution:*** Automate the conflict resolution phase or provide effective recommendations for the users.
- ***Visualization:*** Add a new ability to report conflict using a concrete syntax and provide a graphical and user-friendly interface for supporting conflict management.
- ***Semantic conflict:*** Plan to implement the check for behavioral equivalence and take into account the modeling language semantics to detect conflicts.
- ***Syntactical conflict:*** Extend approach to automate the generation of conflict detection rules and improve model consistency by identifying more syntactic conflict.
- ***Conflict awareness:*** Improve or add new capability to raise awareness of users for potential conflicting changes and detected conflicts.
- ***Support other models:*** Add support for other types of models, or expand the approach to work directly on the meta-metamodeling language.

- ***Composite operation:*** Analyze the candidate operations to improve and speed up the composite operation detection and support more complex operations.
- *Add new techniques or features for other aims such as;*
  - *Improving* ***merge techniques***
  - *Supporting P2P and* ***distributed communication***
  - *Adding conflict* ***recomputation*** *after resolution*
  - *Supporting* ***online collaboration***
  - *Enhancing* ***conflict specification***
  - *Checking the* ***architectural conformance***
  - *Supporting* ***incremental consistency checking***
  - *Enhancing* ***categorization of conflicts***

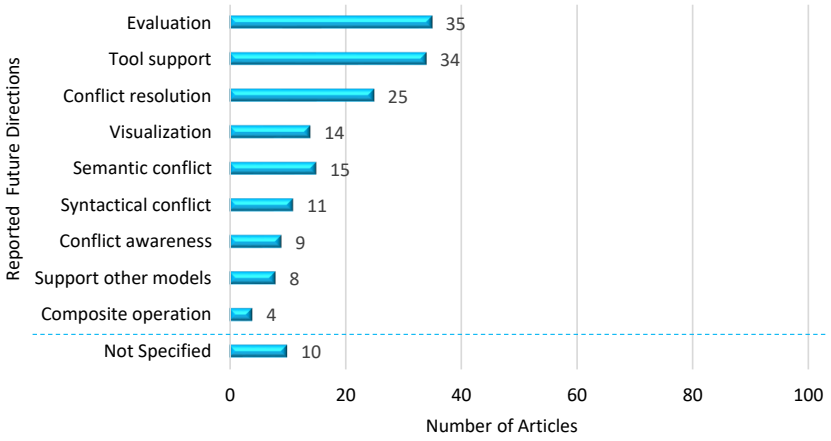


Fig. 29 Data for Q4.3: Reported future directions

6 Open issues and future trends

This section structures our discussion to represent open issues and directions for future work on conflict management as the most relevant challenge for collaborative modeling [13]. Our systematic review results shows that conflict management in model merging has attracted significant research attention over recent years. However, the review also reveals several challenges. In the following, we discuss a set of open issues and research challenges observed in the existing conflict management approaches and speculate on research opportunities and future trends.

Automatic Resolution of Conflicts. One open issue is based on the extent of the existing conflict resolution techniques and tools to automatically

reconcile any conflictual situations. Our result shows that only 20.3 % (14 of 69) of the existing approaches provide an automatic mechanism to resolve conflicts. Moreover, the current approaches (e. g., A37 [108], A49 [120], and A69 [149]) are limited to automatically performing the resolution phase for certain conflictual situations. Furthermore, some existing approaches (e. g., A06 [56], A23 [81], and A55 [127]) only involve the user during the resolution phase to decide on the final choice in the suggestions. The user may ignore or modify the changes applied by other users. In this situation, the lack of knowledge about intentions each user had in mind when modeling can miss the support for passed requirements or lead to a chain of new conflicts [48, 78]. Thus, the modeling intentions of users should be taken into account to reconcile conflicts properly. Accordingly, the conflict resolution approach must provide more intelligent support to automate the resolution phase for any conflictual situation [150].

Semantic Conflict Management. During the analysis of the extracted data, we noticed that some of the approaches are designed to support specific examples of semantic conflicts. Specifically, the proposed techniques are not flexible enough to handle different semantic conflicts or interpret the semantics of various modeling languages. This shortcoming has led research on handling semantic conflict to remain an open challenge in the merge of models. Merging conflicts go beyond modeling language syntax, and without considering the system behavior and the modeling language semantics, many remain undetected. Our results on the type of supported conflict show that the approach proposed in A28 [93,94] is the only approach that focuses on the semantics of a modeling language to detect all three types (cf. Section 2.3) of semantic conflicts. However, it is an inception work that depends on the semantic view definition possibilities. A semantic view definition is a representation of models to introduce certain semantic aspects explicitly [92]. There are also approaches, (e. g., [55,68,87]), that mostly focus on static semantic conflicts and are only available to handle specific semantic conflicts for a specific modeling language. As a result, we speculate that new language-independent techniques and tools that use reasoning capabilities to provide industrial support for all types of semantic conflicts might seem a fruitful avenue for future research.

Visualization for Conflict Management Activities. Our results indicate that an important shortcoming of existing approaches is the visualization techniques needed for most conflict management activities. The visualization of conflicts in the concrete syntax of various models is an important challenge that is not supported for any modeling language. Moreover, only two existing approaches (i. e., A10 [63] and A23 [81]) provide graphical supports for the conflict resolution process. Nevertheless, that capability is important to provide a proper overview of the model elements that appeared in a conflict situation. Our interpretation of this phenomenon is that graphical supports in the conflict resolution phase will be a future trend in conflict management. Another interesting future research direction might be visually guiding users

to describe conflict specifications. Moreover, the graphical representation of changes could be added in the concrete syntax editor to raise the awareness of collaborators to avoid conflicts in the modeling phase. However, there are only a few approaches that focus on conflict visualization (e. g., A22 [80]) or provide a user-friendly graphical editor (e. g., A16 [69] and A43 [114]) for handling conflicts in the model merging process. As a result, the evolution of graphical and visual solutions for conflict management activities remains a main open challenge in this area.

Scalable Conflict Management. To cover different domain aspects of a system, teamwork in software modeling and evolution needs a high number of users that collaboratively work on the model. Moreover, working on complex and industrial cases needs to quickly process large numbers of concurrent edits, which mainly lead to a large-scale model. In this context, collaborative modeling faces issues such as handling conflicts in a reasonable response time and establishing the consistency of the interconnected structure of large models, where concurrent modifications may lead to side effects. Therefore, the availability of scalable techniques and tools for conflict management activities is a prerequisite to ensure the consistency of models in model-based collaboration. However, our study results reveal that only 14.5 % of the existing conflict management approaches (e. g., A32 [99], A39 [110], A61 [138]) support large-scale collaborative modeling, in which half of the approaches proposed a conflict preservation mechanism. Consequently, a further open issue relates to providing scalable solutions to support conflict detection and resolution in the model merging process.

Support of Ordered Features. One open research challenge relates to the support of ordered features in conflict management activities. Ordering property is an attribute for collections and multivalued features in which absolute indices are assigned to each value, where indices are increased sequentially by inserting a new value. In the merge of two versions, two different indices for the features with the same value arise conflict since the feature values cannot be represented with both indices in the merged version. Although most of the modeling languages, e. g., EMF models, allow specifying ordered features, only 20.3 % (14 of 69) of the approaches (e. g., A25 [87], A37 [106], A42 [113]) in our study support conflicts resulting from ordered features. We also noticed that only three approaches (i. e., A01 [47], A05 [55], and A42 [113]) are fully implemented, indicating the need for more empirical works.

Experimental Validity and Empirical Evaluations. The essence of conflict management techniques is based on their ability to integrate with the merge process to ensure the merged model consistency. To achieve this, proper experiments with real-world models from a variety of domains must be performed to fully assess their effectiveness and usability. However, the results of our review reveal that industrial and real-world models, as well as experimental benchmark case studies, were available in just 13.0 % (9 of 69) of

investigated approaches (e. g., A01 [47], A07 [58], and A20 [74]). As a result, one challenge relates to experimental validity and empirical evaluations of conflict management techniques to assess their level of support for model merging.

Local and Global Consistency Checking in Multimodeling. Modeling a complex system requires the deployment of interdependent models conforming to different metamodels that normally result in a *multimodel*. Our study reveals that most of the analyzed approaches focused on a single model view. Although, considering internal relationships among views are important for conflict management and consistency checking between system models [156]. One open challenge about multimodeling is the merge of valid local models, which can violate global constraints declared in the integrated metamodel. Global consistency checking needs to build big and unfeasible merges of metamodels and models, as well as costly model matching to specify the overlaps. Thereby, an important direction for future research is to generalize local checking of constraints imposed on multimodels for the global constraints that regulate the interaction of the multimodel components in a size-controllable way. But the results of our review show that only two approaches (i. e., A66 [146] and A57 [134]) proposed an initial solution, which has not been implemented yet and lacks practical evidence.

Benefiting from Artificial Intelligence Techniques. Due to the recent advance of artificial intelligence, an interesting future trend might be based on the combination of machine learning and deep learning techniques with conflict management activities [150]. We speculate that those techniques can be used to learn implicit user preferences to improve conflict detection and resolution phases. To this end, a challenge relates to the lack of adequate datasets to perform the training phase. However, this challenge does not appear in some artificial intelligence techniques, e. g., reinforcement learning. Natural language processing is another technique that can be helpful to detect semantic equivalent elements and decrease conflict situations in the early phases of collaborative modeling. For instance, Pérez-Soler et al. [157] proposed a collaborative modeling interface using a chatbot that exploits natural language processing to ease decision-making. Interestingly, one of the investigated articles [74] presents an approach (i. e., A20) that avoids conflicting situations using a multi-objective model merging based on NSGA-II. This example suggests that artificial intelligence can bring beyond the conflict management activities in model merging.

Other Opportunities. The results of our study indicate some other gaps in the conflict management techniques for merging the models. Moreover, an in-depth analysis shows that some of the collaboration attributes are more rarely applied by existing approaches that might turn into research opportunities. The results obtained for research questions Q3.1 and Q3.2 clearly show that **conflict specification** and **conflict prevention** are two activities with the least number of proposed approaches with respect to others. The lack of a

proper comprehensive approach in these areas providing support for any type of conflict presents a research direction for the future. Besides, the results of our review show that **conflict management techniques for N-way model merging** are still in their infancy, and starting new research in this area seems interesting. We also think that a conflict management approach should concern **global undo/redo** in a realtime collaborative modeling. Moreover, we speculate that modern organizational practices enforced real-time, distributed, and multi-site networks around the world. Thereby, more researches are required to provide proper conflict management support for **online collaborative modeling over distributed repositories using web-based editors**.

7 Related work

Several systematic mapping studies have been conducted in the field of collaborative modeling. However, none of these studies targets techniques for conflict management in model merging. To the best of our knowledge, there is no systematic mapping study on this area. Only some previous studies on collaborative modeling investigate conflict management as a key feature in collaboration environments. In this section, we review studies that can be related to our research, while they have different objectives.

Zhang et al. [24] study several existing model-based collaborative editing environments to explore options applicable for efficient collaboration in domain-specific modeling environments. They discuss demands and concerns for collaborative modeling in software engineering, giving insight into the development of taxonomies for the concurrency modeling and conflict management. They introduce *concurrency model* and *merging compatible operations* as two main characteristics in the context of collaborative model editing. Moreover, they taxonomized conflict management approaches used in different model-based collaborative editing systems based on the *pessimistic* and *optimistic* collaboration types. They define conflict *avoidance*, *prevention*, and *identification* for pessimistic collaborative modeling, as well as conflict *resolution* approaches for optimistic collaboration. While our taxonomy considers all types of conflicts, they only investigate the approaches that support syntactic conflict management. They do not provide information regarding activities performed in order to manage semantic conflicts. In addition, unlike our work, they do not focus on the detail of techniques that are used for each conflict management approach.

Franzago et al. [13] present a systematic review on collaborative model-driven software engineering. They define a framework for understanding, classifying, and comparing present and future work on collaborative MDSE based on the extracted key information of existing approaches. They represent *model management*, *collaboration*, and *communication* as three complementary dimensions of collaborative MDSE, where conflict management is identified as a necessary part of the *collaboration* dimension. Their study focus on collaborative modeling approaches and only report the ability of approaches to support

conflict detection and resolution. In contrast, we focus on the various aspects of conflict management in the model merging domain to provide an objective taxonomy of conflict management techniques.

David et al. [158] perform a systematic update study over the collaborative modeling papers between 2016 to 2020 to extend the original classification framework proposed by Franzago et al. [13]. This update provides a feature analysis and reports the latest challenges and trends on collaborative MDSE. They found that the typical means of collaborative modeling include versioning systems with conflict management mechanisms, such as conflict *resolution*, conflict *awareness*, and *preventive* conflict management. Compared to the original classification framework, they represent conflict awareness as a new feature for the collaboration dimension. They also indicated that the preventive conflict management approaches had increased slightly in the last five years. However, the goal of their study was to identify the important research directions of the collaborative MDSE field, and in contrast to our work, they do not provide a holistic view of conflict management techniques in the model merging domain.

Masson et al. [15] present a feature model [29] for collaborative modeling environments, where conflict management is identified as a top-level feature. They divide the conflict management feature into mandatory conflict resolution mechanisms and optional conflict awareness approaches. The main difference between their study and ours is that they explore collaborative modeling tools and approaches, while we focus on model merging conflict management approaches. More specifically, we prepare a classification scheme for these approaches to identify their techniques and further explore their relationship with collaborative modeling features.

Finally, Edded et al. [30] propose a systematic mapping study to classify existing collaborative configuration approaches in software product line engineering. Their study addresses conflicting situation management during the collaborative configuration process, where the conflict is defined according to the dependencies between features. They define conflict management strategies in the collaborative configuration approaches that are only used for the purpose of resolving conflicts. Therefore, they have focused on different resolution methods that have been proposed for the collaborative configuration of product lines. However, the main scopes and objectives of our study are totally different, where the conflict is defined in the merge process and based on the syntax and semantics of the models.

To summarize, while these studies can be useful in introducing the concepts of collaborative modeling and understanding their relationship with the conflict management context, they do not provide a comprehensive, in-depth overview of techniques and the state-of-the-art approaches to support conflict management in the model merging process. In this paper, we extended the definition of the conflict management concept introduced by Franzago et al. [13] and reused the conflict management dimensions (i. e., conflict detection, conflict resolution, conflict prevention, and conflict awareness) identified by Masson et al. [15] and David et al. [158]. However, in contrast to existing

studies, we performed a feature analysis on conflict management approaches, finding conflict specification as a new dimension. Moreover, we extended the original techniques with the capability to consider possible sub-features for each dimension. We also reported the latest challenges and trends in conflict management and described envisioned future research.

8 Threats to validity

In this systematic mapping review, we proposed a comparative classification scheme to characterize conflict management techniques in the model merging process. To achieve this, we followed a set of updated guidelines for conducting systematic mapping studies [31,32,33] and carefully designed our study based on a research protocol. Threats to the validity of this systematic review mainly relate to the selection of final pool articles and the process of extracting the appropriate data for understanding and classifying studies. In the following, we discuss the potential threats to the validity of this study and elaborate on how we mitigated them.

The first threat relates to external validity, which refers to the missing of relevant articles that are representative of the research on conflict management in model merging. Another serious threat to the validity of our study results is the bias of researchers in searching, selecting, and classifying articles. We followed Peterson et al. [32] guidelines to ensure that relevant studies were selected to mitigate these potential threats. We formulated a list of search strings from various terms in the domain and adapted them for each digital repository and search engine. Moreover, we have performed both backward and forward snowballing for selected articles to identify articles that may be relevant to our study. To identify terms, we considered the Population and Intervention criteria of PICO guidelines provided by Kitchenham and Charters [38]. Therefore, another potential threat could have been ignoring Comparison and Outcomes features. To ensure the reliability of our identified terms, we used an iterative fashion and a series of test execution and refinement to obtain the final list of keywords.

To apply the inclusion and exclusion criteria, we considered the title, abstract, introduction, and conclusion of articles. In this context, a threat related to external validity may consist of having a set of primary articles not representative of the research on conflict management techniques. This issue can happen to an abstract-based analysis in which introductory sentences are frequently used in abstracts without articles really addressing it. We mitigated this risk by prototyping our method with a full-text review of five random articles and did not find any misclassification. Moreover, in the next step of a systematic mapping review, the full text of all papers came to in-depth analyses. Another potential threat refers to the conflict management tools and industrial researches, which may have been omitted in our study. This potential threat might be mitigated by conducting a future systematic study dedicated

to analysis conflict management tools. Thereby, we avoided to provide any conclusion about conflict management tools and industrial researches.

Concerning the validity of articles in the final pool, we performed a quality assessment process to ensure final pool articles are available to provide sufficient information. This led us to disregard lower-quality studies that were likely decreased the soundness of our study and could bias the final results. Moreover, to ensure the validity of extracted data with respect to the research questions, we performed our search on multiple well-known electronic repositories to avoid potential biases due to publishers' policies and business concerns. Also, to construct the search strings, we considered research questions and refined them by analyzing a set of random studies that reasonably confident about the extracted information.

Concerning the process of extracting accurate and objective data from final articles, we first reviewed the full-text of all articles instead of reviewing only some parts of articles. Also, we precisely described extracted attributes based on the specific context of model merging conflict. Furthermore, each final article was independently classified by two reviewers, and any ambiguity or disagreement was resolved after discussion with the third reviewer. Finally, we applied both vertical and horizontal analysis to prevent any inconsistency among the extracted data according to the conflict management challenges.

9 Conclusion

This systematic review is the most comprehensive mapping of articles in the area of conflict management in model merging. It includes 105 articles, from the years 2001 to mid-2021, which are completely studied to extract influential attributes and construct a precise classification scheme for understanding, classifying, and comparing current and future work in this area.

Our findings indicate that syntactic conflicts are the most common type of conflict that is supported, and most approaches proposed new techniques to detect conflicts or resolve them. A few articles express opinions about the state-of-the-art in conflict management for merging models. Lock-based and conflict-free algorithms are two main techniques for conflict prevention, and most conflict specification approaches work based on patterns. Constraint violation, change overlapping, and pattern matching are three popular categories of conflict detection techniques, and most conflict resolution techniques require user involvement. Also, conflict awareness techniques are not yet fully mature because only A22 has visualized the conflict in the concrete syntax of modeling languages. Other attempts have only been made to report a textual list, notify a message or highlight elements. While the collaboration between academia and industry is increased, there is no study to compare state-of-the-art between academic and industrial tools and techniques for conflict management. Such analysis will help better shape the conflict management tools and techniques for tomorrow.

Next to and based on the extracted systematic map, we also identified several limitations and future directions related to conflict management in model merging. Limitations are primarily about lack of real-world case studies to show the usefulness of the approaches and future directions mostly plan to develop adequate tool support and conduct more experiments. Moreover, trends recently focus on automatic conflict resolution techniques and solutions that have to provide proper conflict management support to deal with online collaboration where multi-team of collaborators work on large-scale models.

Appendix A Classification results

The classification results are presented in Table 4 to Table 9. While we use all features of the taxonomy for the classification, we omit some sub-features in the tables due to lack of space. All the extracted features for each paper have been made publicly available within the replication package at *https://github.com/MSharbaf/CMSysMap*.

References

1. J. Whitehead, "Collaboration in software engineering: A roadmap," in *Proceedings of the Future of Software Engineering (FOSE'07)*, 2007, pp. 214–225.
2. D. Kolovos, L. Rose, N. Matragkas, R. F. Paige, E. Guerra, J.S. Cuadrado, J. De Lara, et al, "A research roadmap towards achieving scalability in model driven engineering," in *Proceedings of the Workshop on Scalability in Model Driven Engineering*, 2013, pp. 1–10.
3. J. Whitehead, I. Mistrík, J. Grundy, and A.V.D. Hoek, "Collaborative software engineering: concepts and techniques," in *Collaborative Software Engineering*, 2010, pp. 1–30.
4. A. Goldberg, "Collaborative software engineering," *Journal of Object Technology*, vol. 1, no. 1, pp. 1–19, 2002.
5. I. Mistrík, J. Grundy, A.V.D. Hoek, and J. Whitehead, "Collaborative software engineering: challenges and prospects," *Collaborative software engineering*, 2010, pp. 389–403.
6. K. Vredenburg, J.Y. Mao, P.W. Smith, and T. Carey, "A survey of user-centered design practice," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2002, pp. 471–478.
7. J.Bézivin, "On the unification power of models," *Software & Systems Modeling*, vol. 4, no. 2, pp. 171–188, 2005.
8. J. Hutchinson, M. Rouncefield, and J. Whittle, "Model-driven engineering practices in industry," in *Proceedings of the 33rd International Conference on Software Engineering*, 2011, pp. 633–642.
9. M. Brambilla, J. Cabot, and M. Wimmer, *Model-driven software engineering in practice, Second Edition*. CA, USA: Morgan & Claypool Publishers, 2017.
10. D. Steinberg, F. Budinsky, E. Merks, and M. Paternostro, *EMF: eclipse modeling framework, Second Revised Edition*. Pearson Education, 2008.
11. K. Altmanninger, M. Seidl, and M. Wimmer, "A survey on model versioning approaches," *International Journal of Web Information Systems*, vol. 5, no. 3, pp. 271–304, 2009.
12. N.N. Zolkifli, A. Ngah, and A. Deraman, "Version control system: A review," *Procedia Computer Science*, vol. 135, pp. 408–415, 2018.
13. M. Franzago, D. Di Ruscio, I. Malavolta, and H. Muccini, "Collaborative model-driven software engineering: a classification framework and a research map," *IEEE Transactions on Software Engineering*, vol. 44, no. 12, pp. 1146–1175, 2017.
14. D. Di Ruscio, M. Franzago, I. Malavolta, and H. Muccini, "Envisioning the future of collaborative model-driven software engineering," in *Proceedings of the 39th International Conference on Software Engineering Companion (ICSE-C)*, 2017, pp. 219–221.

Table 4 Classification of model merging conflict management approaches (A01 to A23)

Approach	Conflict Specification				Conflict Prevention					Conflict Detection				Conflict Resolution					Conflict Awareness		
														Method			Time				
	Formalism	Pattern-based	Natural language	Implicit Spec	Lock-based	Operation ordering	Conflict-free Algo.	Sync Modifications	Change Awareness	Constraint violation	Change overlapping	Pattern matching	Formal methods	Manual	Semi-automatic	Automatic	On-the-fly	Post-analysis	Notification	Conflict Highlighting	Conflict Report
A01 [45,46,47]			*							*	*			*			*				*
A02 [48]				*								*			*		*				*
A03 [49,50]			*								*	*			*			*	*		
A04 [51,52,53]				*						*				*			*		*		
A05 [54,55]				*						*					*			*	*		
A06 [56]				*								*			*		*		*		
A07 [57,58]				*	*					*						*	*				
A08 [59,60,61]			*								*					*	*			*	
A09 [62]				*									*	*				*			*
A10 [63]			*							*				*				*		*	*
A11 [64]				*			*														
A12 [65]				*							*				*		*				*
A13 [66]				*									*			*	*				
A14 [67]		*																			
A15 [68]		*						*				*			*			*	*		
A16 [69]	*											*			*			*		*	*
A17 [70]	*									*			*			*	*				
A18 [71]				*							*			*			*				*
A19 [72]					*																
A20 [73,74]			*			*															
A21 [75]			*																		
A22 [76,77,78,79,80]		*								*	*		*		*			*		*	*
A23 [81]			*									*			*			*		*	*

Table 5 Classification of model merging conflict management approaches (A24 to A46)

Approach	Conflict Specification				Conflict Prevention					Conflict Detection				Conflict Resolution					Conflict Awareness		
														Method			Time				
	Formalism	Pattern-based	Natural language	Implicit Spec	Lock-based	Operation ordering	Conflict-free Algo.	Sync Modifications	Change Awareness	Constraint violation	Change overlapping	Pattern matching	Formal methods	Manual	Semi-automatic	Automatic	On-the-fly	Post-analysis	Notification	Conflict Highlighting	Conflict Report
A24 [82,83,84]		*					*			*					*			*		*	
A25 [85,86,87,88]	*									*	*	*			*		*			*	*
A26 [89]		*										*		*			*				*
A27 [90,91]	*										*				*			*	*		
A28 [92,93,94]				*						*		*		*				*			*
A29 [95]			*							*						*	*				*
A30 [96,97]	*		*							*					*		*		*		
A31 [98]			*								*					*	*				
A32 [99,100,101]				*			*				*					*	*			*	
A33 [102]				*						*				*				*	*		
A34 [103]		*										*		*				*			*
A35 [104]				*							*			*				*		*	
A36 [105]	*					*															
A37 [106,107,108]				*									*			*	*				
A38 [109]	*											*			*			*	*		
A39 [110]				*			*			*				*			*			*	
A40 [111]					*																
A41 [112]	*	*																			
A42 [113]			*							*					*		*			*	
A43 [114]	*										*				*			*			*
A44 [115]	*										*	*		*			*				*
A45 [116]	*												*	*			*				*
A46 [117]		*										*		*				*		*	

Table 6 Classification of model merging conflict management approaches (A47 to A69)

Approach	Conflict Specification				Conflict Prevention					Conflict Detection				Conflict Resolution					Conflict Awareness		
														Method			Time				
	Formalism	Pattern-based	Natural language	Implicit Spec	Lock-based	Operation ordering	Conflict-free Algo.	Sync Modifications	Change Awareness	Constraint violation	Change overlapping	Pattern matching	Formal methods	Manual	Semi-automatic	Automatic	On-the-fly	Post-analysis	Notification	Conflict Highlighting	Conflict Report
A47 [118]				*							*					*	*				
A48 [119]					*																
A49 [120]			*							*						*	*				
A50 [121]	*		*							*						*	*				*
A51 [122]				*							*	*		*				*		*	
A52 [123]			*		*				*												
A53 [124]			*										*	*				*			*
A54 [125]				*							*			*			*		*		
A55 [126,127,128,129,130,131]				*					*	*				*			*		*		
A56 [132]			*							*					*		*				*
A57 [133,134]	*									*			*	*				*			*
A58 [135]				*		*															
A59 [136]				*						*						*	*				
A60 [137]				*							*		*		*		*				*
A61 [138]	*		*							*				*				*			*
A62 [139]				*						*						*	*				
A63 [140,141]	*										*		*	*				*			*
A64 [142]				*							*			*			*				*
A65 [143,144]		*									*		*	*			*				*
A66 [145,146]	*							*													
A67 [147]	*												*	*				*			*
A68 [148]	*												*	*				*			*
A69 [149]		*										*				*	*				

Table 7 Classification of model merging conflict management approaches (A01 to A23)

Approach	Conf. Type				Model Type					Repo. Arch.		Ver. Strat.		Collab. Mech.		Comp. Tech.			Merging Tech.			Ord. Feat.	
	Syntactical	Semantical			Any	UML models	EMF-based	Workflow models	Other	Centralized	Distributed	Optimistic	Pessimistic	Online	Offline	State-based	Operation-based	Hybrid	Raw merging	Two-way merging	Three-way merging	Supported	Not supported
		Static	Behavioral	Equivalence																			
A01 [45,46,47]	*						*				*	*			*		*			*		*	
A02 [48]	*					*				*		*			*		*				*		*
A03 [49,50]	*				*					*		*			*		*			*			*
A04 [51,52,53]	*				*						*	*		*			*			*			*
A05 [54,55]	*	*			*					*		*		*				*			*	*	
A06 [56]	*	*						*		*		*			*		*			*			*
A07 [57,58]	*	*			*					*		*	*	*	*			*			*		*
A08 [59,60,61]	*			*				*		*		*			*		*				*	*	
A09 [62]	*	*				*	*			*		*			*	*					*		*
A10 [63]	*					*	*			*		*			*	*					*		*
A11 [64]	*	*					*			*		*			*	*					*		*
A12 [65]	*					*	*			*		*			*		*				*	*	
A13 [66]	*					*	*				*	*			*		*			*			*
A14 [67]	*	*		*		*	*			*		*			*	*					*		*
A15 [68]	*	*		*		*	*			*		*			*	*					*		*
A16 [69]	*	*					*			*		*			*			*		*			*
A17 [70]	*					*				*		*			*		*				*		*
A18 [71]	*					*				*		*			*	*			*				*
A19 [72]	*	*				*				*			*	*	*		*				*		*
A20 [73,74]	*				*					*		*			*		*				*		*
A21 [75]	*	*		*	*					*		*		*	*	*					*	*	
A22 [76,77,78,79,80]	*	*	*			*				*		*			*		*				*	*	
A23 [81]	*						*			*		*			*	*					*		*

Table 8 Classification of model merging conflict management approaches (A24 to A46)

Approach	Conf. Type				Model Type					Repo. Arch.		Ver. Strat.		Collab. Mech.		Comp. Tech.			Merging Tech.			Ord. Feat.	
	Syntactical	Semantical			Any	UML models	EMF-based	Workflow models	Other	Centralized	Distributed	Optimistic	Pessimistic	Online	Offline	State-based	Operation-based	Hybrid	Raw merging	Two-way merging	Three-way merging	Supported	Not supported
		Static	Behavioral	Equivalence																			
A24 [82,83,84]	*	*					*		*	*		*			*	*	*				*	*	
A25 [85,86,87,88]	*	*					*			*		*			*	*					*	*	
A26 [89]	*	*				*	*			*		*			*		*				*		*
A27 [90,91]	*								*		*	*		*			*			*			*
A28 [92,93,94]	*	*	*	*			*			*		*			*	*					*		*
A29 [95]	*								*	*		*			*	*					*	*	
A30 [96,97]	*		*					*		*		*			*		*				*		*
A31 [98]				*	*					*		*			*		*			*		*	
A32 [99,100,101]	*	*			*						*	*		*			*		*				*
A33 [102]	*					*				*		*			*	*					*		*
A34 [103]	*				*					*		*			*		*			*			*
A35 [104]	*					*				*		*			*	*					*		*
A36 [105]	*	*					*				*	*		*			*		*			*	
A37 [106,107,108]	*	*			*					*		*			*		*				*	*	
A38 [109]	*							*		*		*		*			*			*			*
A39 [110]	*						*			*		*		*		*	*			*			*
A40 [111]	*	*							*	*			*		*		*			*			*
A41 [112]	*	*	*	*		*	*			*		*				*					*		*
A42 [113]	*						*		*	*		*			*	*					*	*	
A43 [114]	*								*	*		*			*		*			*			*
A44 [115]	*					*				*		*			*		*				*		*
A45 [116]				*		*				*		*			*	*					*		*
A46 [117]	*					*	*			*		*			*			*			*		*

Table 9 Classification of model merging conflict management approaches (A47 to A69)

Approach	Conf. Type				Model Type					Repo. Arch.		Ver. Strat.		Collab. Mech.		Comp. Tech.			Merging Tech.			Ord. Feat.	
	Syntactical	Semantical			Any	UML models	EMF-based	Workflow models	Other	Centralized	Distributed	Optimistic	Pessimistic	Online	Offline	State-based	Operation-based	Hybrid	Raw merging	Two-way merging	Three-way merging	Supported	Not supported
		Static	Behavioral	Equivalence																			
A47 [118]	*					*				*		*			*		*				*		*
A48 [119]	*	*				*	*			*			*		*	*				*			*
A49 [120]		*						*		*		*			*	*				*			*
A50 [121]	*								*	*		*			*	*				*			*
A51 [122]	*	*				*					*	*			*		*			*			*
A52 [123]	*				*					*			*	*			*		*				*
A53 [124]	*								*	*		*			*	*				*			*
A54 [125]	*					*				*		*			*		*				*		*
A55 [126,127,128,129,130,131]	*				*						*	*		*			*		*				*
A56 [132]	*			*	*					*		*			*	*				*			*
A57 [133,134]		*			*					*		*			*	*				*			*
A58 [135]	*					*				*		*			*	*				*			*
A59 [136]		*							*	*		*			*	*				*			*
A60 [137]				*		*				*		*			*	*				*			*
A61 [138]	*					*				*		*			*		*		*				*
A62 [139]	*					*				*		*		*			*		*				*
A63 [140,141]	*								*	*		*			*	*					*	*	
A64 [142]	*					*				*		*			*		*			*			*
A65 [143,144]	*					*				*		*			*	*				*			*
A66 [145,146]	*				*					*		*			*	*				*			*
A67 [147]	*								*	*		*			*	*				*			*
A68 [148]	*								*	*		*			*	*				*			*
A69 [149]	*								*		*	*		*			*			*			*

15. C. Masson, J. Corley, and E. Syriani, "Feature Model for Collaborative Modeling Environments," in *Proceedings of the 2nd international workshop on collaborative modelling in MDE at MODELS*, 2017, pp. 164–173.
16. F. Hojaji, T. Mayerhofer, B. Zamani, A. Hamou-Lhadj, and E. Bousse, "Model execution tracing: a systematic mapping study," *Software and Systems Modeling*, vol. 18, no. 6, pp.3461–3485, 2019.
17. T. Mens, "A state-of-the-art survey on software merging," *IEEE transactions on software engineering*, vol. 28, no. 5, pp. 449–462, 2002.
18. M. Sharbaf, B. Zamani, and B.T. Ladani, "Towards automatic generation of formal specifications for UML consistency verification," in *Proceedings of the 2nd International Conference on Knowledge-Based Engineering and Innovation*, 2015, pp. 860–865.
19. M. Chechik, S. Nejati, and M. Sabetzadeh, "A relationship-based approach to model integration," in *Innovations in Systems and Software Engineering*, vol. 8, no. 1, pp. 3–18, 2012.
20. S. Barrett, P. Chalin, and G. Butler, "Model merging falls short of software engineering needs," in *Proceedings of the 2nd Workshop on Model-Driven Software Evolution*, 2008, pp. 13–18.
21. R. Conradi and B. Westfechtel, "Version models for software configuration management," in *ACM Computing Surveys (CSUR)*, vol. 30, no. 2, pp.232–282, 1998.
22. S. Ram and V. Ramesh, "Collaborative conceptual schema design: a process model and prototype system," *ACM Transactions on Information Systems (TOIS)*, vol. 16, no. 4, pp. 347–371, 1998.
23. K. Altmanninger and A. Pierantonio, "A categorization for conflicts in model versioning," *Elektrotechnik und Informationstechnik*, vol. 128, no. 11-12, pp. 421–426, 2011.
24. P. Zhang, S. Neema, and T. Bapty, "A study of collaborative efforts and proposed visualizations in domain-specific modeling environment," in *Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, 2016, pp. 146–152.
25. M. Stephan and J.R. Cordy, "A Survey of Model Comparison Approaches and Applications," in *Proceedings of the 1st International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*, 2013, pp. 265–277.
26. L. Gonçales, K. Farias, M. Scholl, T.C. Oliveira, and M. Veronez, "Model Comparison: a Systematic Mapping Study," in *Proceedings of the International Conference on Software Engineering and Knowledge Engineering*, 2015, pp. 546–551.
27. L.Addazi and A. Cicchetti, "Systematic Evaluation of Model Comparison Algorithms using Model Generation," *Journal of Object Technology*, vol. 19, no. 1, pp. 1–22, 2020.
28. H. Bruneliere, E. Burger, J. Cabot, and M. Wimmer, "A feature-based survey of model view approaches," *Software & Systems Modeling*, vol. 18, no. 3, pp. 1931–1952, 2019.
29. P.Y. Schobbens, P. Heymans, J.C. Trigaux, and Y. Bontemps, "Generic semantics of feature diagrams," *Computer networks*, vol. 51, no. 2, pp.456–479, 2007.
30. S. Edded, S.B. Sassi, R. Mazo, C. Salinesi, and H.B. Ghezala, "Collaborative configuration approaches in software product lines engineering: A systematic mapping study," *Journal of Systems and Software*, vol. 158, pp. 110422, 2019.
31. P. Brereton, B.A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *Journal of Systems and Software*, vol. 80, no. 4, pp. 571–583, 2007.
32. K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology*, vol.64, pp. 1–18, 2015.
33. K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, 2008, pp. 1–10.
34. Á.J. Cuadros López, C. Galindres, and P. Ruiz, "Project maturity evaluation model for SMEs from the software development sub-sector," *AD-minister*, vol, 29, pp. 147–162, 2016.
35. R. Wieringa, N. Maiden, N. Mead, and C. Rolland, "Requirements engineering paper classification and evaluation criteria: a proposal and a discussion," *Requirements engineering*, vol. 11, no. 1, pp. 102–107, 2006.

36. V.R. Basili, G. Caldiera and H. D. Rombach, "The goal question metric approach," *Encyclopedia of software engineering*, pp. 528–532, 1994.
37. L. Chen, M.A. Babar, and H. Zhang, "Towards an evidence-based understanding of electronic data sources," in *Proceedings of the 14th International Conference on Evaluation and Assessment in Software Engineering*, 2010, pp. 1–4.
38. B. Kitchenham and S. Charters, "Guidelines for Performing Systematic Literature Reviews in Software Engineering," *Technical report, Ver. 2.3*, School of Computer Science and Mathematics, Keele University,, 2007.
39. M. Kuhrmann, D.M. Fernández, and M. Daneva, "On the pragmatic design of literature studies in software engineering: an experience-based guideline," *Empirical software engineering*, vol.22, no. 6, pp. 2852–2891, 2017.
40. C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, 2014, pp. 1–10.
41. B. Kitchenham and P. Brereton, "A systematic review of systematic review process research in software engineering," *Information and software technology*, vol. 55, no, 12, pp. 2049–2075, 2013.
42. D. Spencer, *Card sorting: Designing usable categories*, New York, NY, USA : Rosenfeld Media, 2009.
43. D.S. Cruzes and T. Dybå, "Research synthesis in software engineering: A tertiary study," *Information and Software Technology*, vol. 53, no. 5, pp. 440–455, 2011.
44. S. Shafiq, A. Mashkoor, C. Mayr-Dorn, and A. Egyed, "Machine Learning for Software Engineering: A Systematic Mapping," *arXiv preprint*, arXiv:2005.13299, 2020.
45. A. Koshima, V. Englebert, and P. Thiran, "Distributed collaborative model editing framework for domain specific modeling tools," in *Proceedings of the 6th International Conference on Global Software Engineering*, 2011, pp. 113–118.
46. A. Koshima and V. Englebert, "Collaborative Editing of EMF/Ecore Meta-models and Models," in *Proceedings of the 2nd International Conferenceon Model-Driven Engineering and Software Development (MODELSWARD-2014)*, 2014, pp. 55–66.
47. A. Koshima and V. Englebert, "Collaborative editing of EMF/Ecore meta-models and models: Conflict detection, reconciliation, and merging in DiCoMEF," *Science of Computer Programming*, vol. 113, pp. 3–28, 2015.
48. H. Chong, R. Zhang, and Z. Qin, "Composite-based conflict resolution in merging versions of UML models," in *Proceedings of the 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 2016, pp. 127–132.
49. M. Koegel, H Jonas, and S. Stephan, "Operation-based conflict detection and resolution," in *Proceedings of the ICSE Workshop on Comparison and Versioning of Software Models*, 2009, pp. 43–48.
50. M. Koegel, M. Herrmannsdoerfer, O. Wesendonk, and J. Helming, "Operation-based conflict detection," in *Proceedings of the 1st International Workshop on Model Comparison in Practice*, 2010, pp. 21–30.
51. J.Y. Bang, Y. Brun, and N. Medvidović, "Collaborative-Design Conflicts: Costs and Solutions," *IEEE Software*, vol. 35, no. 6, pp. 25–31, 2018.
52. J.Y. Bang and N.Medvidović, "Proactive detection of higher-order software design conflicts," in *Proceedings of the 12th Working IEEE/IFIP Conference on Software Architecture*, 2015, pp. 155–164.
53. J.Y. Bang, Y. Brun, and N. Medvidović, "Continuous analysis of collaborative design," in *Proceedings of the International Conference on Software Architecture (ICSA)*, 2017, pp. 97–106.
54. H.K. Dam, A. Reder, and A. Egyed, "Inconsistency resolution in merging versions of architectural models," in *Proceedings of the IEEE/IFIP Conference on Software Architecture*, 2014, pp. 153–162.
55. H.K. Dam, A. Egyed, M. Winikoff, A. Reder, and R.E. Lopez-Herrejon, "Consistent merging of model versions," *Journal of Systems and Software*, vol. 112, pp. 137–155, 2016.
56. S. Erol and G. Neumann,"Handling concurrent changes in collaborative process model development: a change-pattern based approach," in *Proceedings of the 17th IEEE International Enterprise Distributed Object Computing Conference Workshops*, 2013, pp. 250–257.

57. C. Debreceni, G. Bergmann, I. Ráth, and V. Dániel, "Property-based locking in collaborative modeling," in *Proceedings of the 20th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, 2017, pp. 199–209.
58. C. Debreceni, I. Ráth, V. Dániel, X. De Carlos, X. Mendialdua, and S. Trujillo, "Automated model merge by design space exploration," in *Proceedings of the International Conference on Fundamental Approaches to Software Engineering*, 2016, pp. 104–121.
59. C. Gerth, J.M. Küster, M. Luckey, and G. Engels, "Detection and resolution of conflicting change operations in version management of process models," *Software & Systems Modeling*, vol. 12, no. 3, pp. 517–535, 2013.
60. C. Gerth, J.M. Küster, M. Luckey, and G. Engels, "Precise detection of conflicting change operations using process model terms," in *Proceedings of the International Conference on Model Driven Engineering Languages and Systems*, 2010, pp. 93–107.
61. J.M. Küster, C. Gerth, and G. Engels, "Dependent and conflicting change operations of process models," in *Proceedings of the European Conference on Model Driven Architecture-Foundations and Applications*, 2009, pp. 158–173.
62. C. Bartelt, "Conflict analysis at collaborative development of domain specific models using description logics," in *Proceedings of the 44th Hawaii International Conference on System Sciences*, 2011, pp. 1–9.
63. C. Bartelt and B. Schindler, "Technology Support for Collaborative Inconsistency Management in Model Driven Engineering," in *Proceedings of the 43rd Hawaii International Conference on System Sciences*, 2010, pp. 1–7.
64. T. Kehrer, U. Kelter, and G. Taentzer, "Consistency-preserving edit scripts in model versioning," in *Proceedings of the 28th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2013, pp. 191–201.
65. P. Sriplakich, X. Blanc, and M.P. Gervais, "Supporting collaborative development in an open MDA environment," in *Proceedings of the 22nd IEEE International Conference on Software Maintenance*, 2006, pp. 244–253.
66. A. Mougenot, X. Blanc, and M.P. Gervais, "D-praxis: A peer-to-peer collaborative model editing framework," in *Proceedings of the International Conference on Distributed Applications and Interoperable Systems*, 2009, pp. 16–29.
67. M. Sharbaf and B. Zamani, "A UML profile for modeling the conflicts in model merging," in *Proceedings of the 4th IEEE International Conference on Knowledge-Based Engineering and Innovation*, 2017, pp. 197–202.
68. M. Sharbaf and B. Zamani, "Configurable three-way model merging," *Software: Practice and Experience*, vol. 50, no. 8, pp. 1565–1599, 2020.
69. S.C. Barrett, P. Chalin, and G. Butler, "Table-driven detection and resolution of operation-based merge conflicts with mirador," in *Proceedings of the European Conference on Modelling Foundations and Applications*, 2011, pp. 329–344.
70. M. Zerrouk, A. Anwar, and I. Benelallam, "Managing Model Conflicts in Collaborative Modeling Using Constraint Programming," in *Proceedings of the 5th International Congress on Information Science and Technology (CiSt)*, 2018, pp. 117-123.
71. A. De Lucia, F. Fasano, G. Scanniello, and G. Tortora, "Concurrent fine-grained versioning of UML models," in *Proceedings of the 13th European Conference on Software Maintenance and Reengineering*, 2009, pp. 89–98.
72. M. Cataldo, C. Shelton, Y. Choi, Y.Y. Huang, V. Ramesh, D. Saini, and L.Y. Wang, "Camel: A tool for collaborative distributed software design," in *Proceedings of the 4th International Conference on Global Software Engineering*, 2009, pp. 83–92.
73. M. Kessentini, W. Werda, P. Langer, and M. Wimmer, "Search-based model merging," in *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, 2013, pp. 1453–1460.
74. U. Mansoor, M. Kessentini, P. Langer, M. Wimmer, S. Bechikh, and K. Deb, "MOMM: Multi-objective model merging," *Journal of Systems and Software*, vol. 103, pp. 423–439, 2015.
75. P. Brosch, P. Langer, M. Seidl, K. Wieland, and M. Wimmer, "Colex: a web-based collaborative conflict lexicon," in *Proceedings of the 1st International Workshop on Model Comparison in Practice*, 2010, pp. 42–49.
76. P. Brosch, H. Kargl, P. Langer, M. Seidl, K. Wieland, M. Wimmer, and G. Kappel, "Conflicts as first-class entities: a UML profile for model versioning," in *Proceedings of Models in Software Engineering Workshop at MODELS*, 2010, pp. 184–193.

77. P. Brosch, M. Seidl, and G. Kappel, "A recommender for conflict resolution support in optimistic model versioning," in *Proceedings of the ACM international conference companion on Object oriented programming systems languages and applications*, 2010, pp. 43–50.
78. P. Brosch, G. Kappel, P. Langer, M. Seidl, K. Wieland, and M. Wimmer, "An introduction to model versioning," in *Proceedings of the International School on Formal Methods for the Design of Computer, Communication and Software Systems*, 2012, pp. 336–398.
79. P. Brosch, U. Egly, S. Gabmeyer, G. Kappel, M. Seidl, H. Tompits, M. Widl, and M. Wimmer, "Towards semantics-aware merge support in optimistic model versioning," in *Proceedings of the International Conference on Model Driven Engineering Languages and Systems*, 2011, pp. 246–256.
80. P. Brosch, M. Seidl, M. Wimmer, and G. Kappel, "Conflict Visualization for Evolving UML Models," *Journal of Object Technology*, vol. 11, no. 3, pp. 1–30, 2012.
81. K. Wieland, P. Langer, M. Seidl, M. Wimmer, and G. Kappel, "Turning conflicts into collaboration," *Computer Supported Cooperative Work (CSCW)*, vol. 22, no. 2-3, pp. 181–240, 2013.
82. G. Taentzer, C. Ermel, P. Langer, and M. Wimmer, "A fundamental approach to model versioning based on graph modifications: from theory to implementation," *Software & Systems Modeling*, vol. 13, no. 1, pp. 239–272, 2014.
83. G. Taentzer, C. Ermel, P. Langer, and M. Wimmer, "Conflict detection for model versioning based on graph modifications," in *Proceedings of the International Conference on Graph Transformation*, 2010, pp. 171-186.
84. H. Ehrig, C. Ermel, and G. Taentzer, "A formal resolution strategy for operation-based conflicts in model versioning using graph modifications," in *Proceedings of the International Conference on Fundamental Approaches to Software Engineering*, 2011, pp. 202–216.
85. F. Schwägerl, S. Uhrig, and B. Westfechtel, "Model-based tool support for consistent three-way merging of EMF models," in *Proceedings of the workshop on ACadeMics Tooling with Eclipse*, 2013, pp. 1–10.
86. F. Schwägerl, S. Uhrig, and B. Westfechtel, "A graph-based algorithm for three-way merging of ordered collections in EMF models," *Science of Computer Programming*, vol. 113, pp. 51–81, 2015.
87. B. Westfechtel, "Merging of EMF models," *Software & Systems Modeling*, vol. 13, no. 2, pp. 757–788, 2014.
88. B. Westfechtel, "A formal approach to three-way merging of EMF models," in *Proceedings of the 1st International Workshop on Model Comparison in Practice*, 2010, pp. 31–41.
89. A. Cicchetti, D. Di Ruscio, and A. Pierantonio, "Managing model conflicts in distributed development," in *Proceedings of the International Conference on Model Driven Engineering Languages and Systems*, 2008, pp. 311–325.
90. E. Kuiter, S. Krieter, J. Krüger, T. Leich, and G. Saake, "Foundations of collaborative, real-time feature modeling," in *Proceedings of the 23rd International Systems and Software Product Line Conference-Volume A*, 2019, pp. 257–264.
91. E. Kuiter, S. Krieter, J. Krüger, G. Saake, and T. Leich, "variED: an editor for collaborative, real-time feature modeling," *Empirical Software Engineering*, vol. 26, no. 2, pp. 1–47, 2021.
92. K. Altmanninger, "Models in conflict–towards a semantically enhanced version control system for models," in *Proceedings of the International Conference on Model Driven Engineering Languages and Systems*, 2007, pp. 293–304.
93. K. Altmanninger, A. Bergmayr, W. Schwinger, and G. Kotsis, "Semantically enhanced conflict detection between model versions in SMoVer by example," in *Proceedings of the International Workshop on Semantic-Based Software Development at OOPSLA*, 2007.
94. K. Altmanninger, W. Schwinger, and G. Kotsis, "Semantics for accurate conflict detection in smover: Specification, detection and presentation by example," *International Journal of Enterprise Information Systems (IJEIS)*, vol. 6, no. 1, pp. 68–84, 2010.
95. A. Baqasah, E. Pardede, and W. Rahayu, "A new approach for meaningful XML schema merging," in *Proceedings of the 16th International Conference on Information Integration and Web-based Applications & Services*, 2014, pp. 430–439.

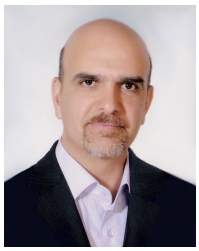
96. S. Mafazi, W. Mayer, and M. Stumptner, "Conflict resolution for on-the-fly change propagation in business processes," in *Proceedings of the 10th Asia-Pacific Conference on Conceptual Modelling-Volume 154*, 2014, pp. 39–48.
97. G. Grossmann, S. Mafazi, W. Mayer, M. Schrefl, and M. Stumptner, "Change propagation and conflict resolution for the co-evolution of business processes," *International Journal of Cooperative Information Systems*, vol. 24, no. 01, pp. 1540002, 2015.
98. A. Boronat, Artur, J.Á. Carsí, I. Ramos, and P. Letelier, "Formal model merging applied to class diagram integration," *Electronic Notes in Theoretical Computer Science*, vol. 166, pp. 5–26, 2007.
99. P. Nicolaescu, M. Rosenstengel, M. Derntl, R. Klamma, and M. Jarke, "Near real-time collaborative modeling for view-based web information systems engineering," *Information Systems*, vol. 74, pp. 23–39, 2018.
100. M. Derntl, P. Nicolaescu, S. Erdtmann, R. Klamma, and M. Jarke, "Near real-time collaborative conceptual modeling on the web," in *Proceedings of the International Conference on Conceptual Modeling*, 2015, pp. 344–357.
101. P. Nicolaescu, M. Rosenstengel, M. Derntl, R. Klamma, and M. Jarke, "View-based near real-time collaborative modeling for information systems engineering," in *Proceedings of the International Conference on Advanced Information Systems Engineering*, 2016, pp. 3–17.
102. L. Murta, H. Oliveira, C. Dantas, L.G. Lopes, and C. Werner, "Odyssey-SCM: An integrated software configuration management infrastructure for UML models," *Science of Computer Programming*, vol. 65, no. 3, pp. 249–274, 2007.
103. A. Rajbhoj and S. Reddy, "A graph-pattern based approach for meta-model specific conflict detection in a general-purpose model versioning system," in *Proceedings of the International Conference on Model Driven Engineering Languages and Systems*, 2013, pp. 422–435.
104. K. Phalp, F. Grimm, and L. Xu, "Supporting Collaborative Work by Preserving Model Meaning When Merging Graphical Models," in *Proceedings of the Working Conference on Virtual Enterprises*, 2012, pp. 262–269.
105. G. Sunyé, "Model consistency for distributed collaborative modeling," in *Proceedings of the European Conference on Modelling Foundations and Applications*, 2017, pp. 197–212.
106. A. Rossini, A. Rutle, Y. Lamo, and U. Wolter, "A formalisation of the copy-modify-merge approach to version control in MDE," *The Journal of Logic and Algebraic Programming*, vol. 79, no. 7, pp. 636–658, 2009.
107. A. Rutle, A. Rossini, Y. Lamo, and U. Wolter, "A category-theoretical approach to the formalisation of version control in MDE," in *Proceedings of the International Conference on Fundamental Approaches to Software Engineering*, 2009, pp. 64–78.
108. A. Rossini, A. Rutle, Y. Lamo, and U. Wolter, "Handling constraints in model versioning," in *Proceedings of the 3rd International workshop on collaborative modelling in MDE at MODELS*, 2018, pp. 1–9.
109. H.M. Cai, X.F. Ji, and F.L. Bu, "Research of consistency maintenance mechanism in real-time collaborative multi-view business modeling," *Journal of Shanghai Jiaotong University (Science)*, vol. 20, no. 1, pp. 86–92, 2015.
110. P. Zweihoff, S. Naujokat, and B. Steffen, "Pyro: Generating domain-specific collaborative online modeling environments," in *Proceedings of the International Conference on Fundamental Approaches to Software Engineering*, 2019, pp. 101–115.
111. S. Kelly, "Collaborative modelling with version control," in *Proceedings of the Federation of International Conferences on Software Technologies: Applications and Foundations*, 2017, pp. 20–29.
112. M. Sharbaf, B. Zamani, and G. Sunyé, "A Formalism for Specifying Model Merging Conflicts," in *Proceedings of the 12th System Analysis and Modelling Conference*, 2020, pp. 1–10.
113. J. Schröpfer, F. Schwägerl, and B. Westfechtel, "Consistency Control for Model Versions in Evolving Model-Driven Software Product Lines," in *Proceedings of the 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, 2019, pp. 268–277.

114. T. Mens, G. Taentzer, and O. Runge, "Detecting structural refactoring conflicts using critical pair analysis," *Electronic Notes in Theoretical Computer Science*, vol. 127, no. 3, pp. 113–128, 2005.
115. Z. Zhang, R. Zhang, and Z. Qin, "Composite-level conflict detection in uml model versioning," *Mathematical Problems in Engineering*, vol. 2015, pp. 1–9, 2015.
116. V.O. Costa, J.M.B. Junior, and L.G.P. Murta, "Semantic Conflicts Detection in Model-driven Engineering," in *Proceedings of the 25th International Conference on Software Engineering and Knowledge Engineering*, 2013, pp. 656–661.
117. H.G. Schaathun and A. Rutle, "Model-Driven Software Engineering in the Resource Description Framework: a way to version control," in *Norsk IKT-konferanse for forskning og utdanning*, 2018, pp. 1–12.
118. N.M.S. Iswari and F.N. Azizah, "Improvement of Adaptable Model Versioning (AMOR) framework for software model versioning using critical pair analysis," in *Proceedings of the International Conference on Data and Software Engineering (ICODSE)*, 2014, pp. 1–5.
119. K. Jahed, M. Bagherzadeh, and J. Dingel, "On the benefits of file-level modularity for EMF models," *Software and Systems Modeling*, vol. 20, no. 1, pp. 267–286, 2021.
120. A. Hachemi and M. Ahmed-Nacer, "Conflict resolution in process models merging," in *Proceedings of the Computational Methods in Systems and Software*, 2020, pp. 336–345.
121. S. Edded, S. Sassi, R. Mazo, C. Salinesi, and H. Ghézala, "Preference-based Conflict Resolution for Collaborative Configuration of Product Lines," in *Proceedings of the International Conference on Evaluation of Novel Approaches to Software Engineering*, 2020, pp. 1–8.
122. K. Farias,TC. de Oliveira, LJ. Gonçales, and V. Bischoff, "UML2Merge: a UML extension for model merging," *IET Software*, vol. 13, no. 6, pp. 575–586, 2019.
123. N. Kanagasabai, O. Alam, and J. Kienzle, "Towards online collaborative multi-view modelling," in *Proceedings of the 10th International Conference on System Analysis and Modeling*, 2018, pp. 202–218.
124. B. Kallweit, P.O. Antonino, J. Jahic, T. Kuhn, and P. Liggesmeyer, "Detection of conflicts and inconsistencies between architecture solutions," in *Proceedings of the 13th European Conference on Software Architecture-Volume 2*, 2019, pp. 183–189.
125. M. Foucault, S. Barbier, and D. Lugato, "Enhancing version control with domain-specific semantics," in *Proceedings of the 5th International Workshop on Modeling in Software Engineering (MiSE)*, 2013, pp. 31–36.
126. M.A. Tröls, A. Mashkoor, and A. Egyed, "Multifaceted consistency checking of collaborative engineering artifacts," in *Proceedings of the 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, 2019, pp. 278–287.
127. M.A. Tröls, A. Mashkoor, and A. Egyed, "Live and global consistency checking in a collaborative engineering environment," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, 2019, pp. 1776–1785.
128. M.A. Tröls, A. Mashkoor, and A. Egyed, "Collaboratively enhanced consistency checking in a cloud-based engineering environment," in *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, 2019, pp. 1–6.
129. M.A. Tröls, A. Mashkoor, and A. Egyed, "Timestamp-based Consistency Checking of Collaboratively Developed Engineering Artifacts," in *Proceedings of the International Conference on Software and System Processes*, 2021, pp. 1–11.
130. M.A. Tröls, A. Mashkoor, and A. Egyed, "Hierarchical Distribution of Consistency-relevant Changes in a Collaborative Engineering Environment," in *Proceedings of the International Conference on Software and System Processes*, 2021, pp. 1–11.
131. M.A. Tröls, A. Mashkoor, and A. Egyed, "Instant distribution of consistency-relevant change information in a hierarchical multi-developer engineering environment," in *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, 2021, pp. 1572–1575.
132. R.A. Pottinger and P.A. Bernstein, "Merging models based on given correspondences," in *Proceedings of the 29th International conference on Very large data bases*, 2003, pp. 862–873.
133. P. Stünkel, H. König, Y. Lamo, and A. Rutle, "Towards Multiple Model Synchronization with Comprehensive Systems," in *Proceedings of the International Conference on Fundamental Approaches to Software Engineering*, 2020, pp. 335–356.

134. P. Stünkel, H. König, Y. Lamo, and A. Rutle, "Comprehensive Systems: A formal foundation for Multi-Model Consistency Management," *Formal Aspects of Computing*, pp. 1–48, 2021.
135. W. Assunçã, SR. Vergilio, and R.E. Lopez-Herrejon, "Discovering software architectures with search-based merge of UML model variants," in *Proceedings of the International Conference on Software Reuse*, 2017, pp. 95–111.
136. G. Perrouin, E. Brottier, B. Baudry, and Y.Le Traon, "Composing models for detecting inconsistencies: A requirements engineering perspective," in *Proceedings of the International Working Conference on Requirements Engineering: Foundation for Software Quality*, 2009, pp. 89–103.
137. S. Nejati, "Formal support for merging and negotiation," in *Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering*, 2005, pp. 456–460.
138. A. Demuth, M. Riedl-Ehrenleitner, and A. Egyed, "Efficient detection of inconsistencies in a multi-developer engineering environment," in *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2016, pp. 590–601.
139. H.K. Dam and A. Ghose, "An agent-based framework for distributed collaborative model evolution," in *Proceedings of the 12th International Workshop on Principles of Software Evolution and the 7th annual ERCIM Workshop on Software Evolution*, 2011, pp. 121–130.
140. M. Sabetzadeh and S. Easterbrook, "An algebraic framework for merging incomplete and inconsistent views," in *Proceedings of the 13th IEEE International Conference on Requirements Engineering*, 2005, pp. 306–315.
141. M. Sabetzadeh and S. Easterbrook, "View merging in the presence of incompleteness and inconsistency," *Requirements Engineering*, vol. 11, no. 3, pp. 174–193, 2007.
142. J.P. Jyani and N.C. Barwar, "A Collaborative Versioning Framework for Model-Based Version Control Systems," in *Proceedings of the International Conference on Communication and Computational Technologies*, 2021, pp. 623–639.
143. M. Sabetzadeh, S. Nejati, S. Easterbrook, and M. Chechik, "Global consistency checking of distributed models with TReMer+," in *Proceedings of the 30th International Conference on Software Engineering*, 2008, pp. 815–818.
144. M. Sabetzadeh, S. Nejati, S. Liaskos, S. Easterbrook, and M. Chechik, "Consistency checking of conceptual models via model merging," in *Proceedings of the International Requirements Engineering Conference*, 2007, pp. 221–230.
145. H. König and Z. Diskin, "Efficient consistency checking of interrelated models," in *Proceedings of the European Conference on Modelling Foundations and Applications*, 2017, pp. 161–178.
146. H. König and Z. Diskin, "Advanced local checking of global consistency in heterogeneous multimodeling," in *Proceedings of the European Conference on Modelling Foundations and Applications*, 2016, pp. 19–35.
147. M. Sabetzadeh and S. Easterbrook, "Analysis of inconsistency in graph-based viewpoints: a category-theoretical approach," in *Proceedings of the International Conference on Automated Software Engineering*, 2003, pp. 12–21.
148. S. Easterbrook and M. Chechik, "A framework for multi-valued reasoning over inconsistent viewpoints," in *Proceedings of the 23rd International Conference on Software Engineering*,2001, pp. 411–420.
149. L. Fritsche, J. Kosiol, A. Möller, A. Schürr, and G. Taentzer, "A precedence-driven approach for concurrent model synchronization scenarios using triple graph grammars," in *Proceedings of the 13th International Conference on Software Language Engineering*, 2020, pp. 39–55.
150. M. Sharbaf, B. Zamani, and G. Sunyé, "Automatic Resolution of Model Merging Conflict using Quality-Based Reinforcement Learning," *Journal of Computer Languages*, vol. 70, pp. 101123, 2022.
151. T. Frühwirth, A. Herold, V. Küchenhoff, T.L. Provost, P. Lim, E. Monfroy, and M. Wallace Constraint logic programming," in *Logic Programming Summer School*, 1992, pp. 3–35.
152. H. Zhu and L. Shan, "Well-formedness, consistency and completeness of graphic models," in *Proceedings of the 9th International Conference on Computer Modelling and Simulation*, 2006, pp. 47–53.

153. M. Shapiro, N. Preguiça, C. Baquero, and M. Zawirski, "Conflict-free replicated data types," in *Symposium on Self-Stabilizing Systems*, 2011, pp. 386–400.
154. L. Mandel and M.V. Cengarle, "On the expressive power of OCL," in *International Symposium on Formal Methods*, 1999, pp. 854–874.
155. D.S. Kolovos, R.F. Paige, and F.A. Polack, "On the evolution of OCL for capturing structural constraints in modelling languages," in *Rigorous Methods for Software Construction and Analysis*, 2009, pp. 204–218.
156. M. Sharbaf, B. Zamani, and G. Sunyé, "Towards Personalized Change Propagation for Collaborative Modeling," in *Proceedings of the 24th International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, 2021, pp. 3–7.
157. S. Pérez-Soler, E. Guerra, and J. de Lara, "Collaborative modeling and group decision making using chatbots in social networks," *IEEE Software*, vol. 35, no. 6, pp. 48–54, 2018.
158. I. David, K. Aslam, S. Faridmoayer, I. Malavolta, E. Syriani, and P. Lago, "Collaborative model-driven software engineering: a systematic update," in *Proceedings of the 24th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, 2021, pp. 273–284.

Mohammadreza Sharbaf is a Ph.D. candidate in Computer Science. He is currently enrolled in a cotutelle program between the University of Isfahan, Iran, and the University of Nantes, France. He received his B.Sc. from the Isfahan University of Technology in 2013, and his M.Sc. from the University of Isfahan in 2016, both in Software Engineering. His research interest includes Collaborative Modeling, Model Merging, Model Transformation, and Domain-Specific Modelling Languages. He is an active member of the MDSE Research Group at the University of Isfahan and the NaoMod Team at the University of Nantes. His current research is focused on Conflict Management in Collaborative Modeling.



Bahman Zamani holds a Ph.D. in Computer Science from Concordia University, Montreal, QC, Canada for his work on the Pattern Language Verification. Currently, he is an Associate Professor in the Department of Software Engineering, University of Isfahan, Isfahan, Iran. His main research interest is Model-Driven Software Engineering (MDSE). He is the founder and director of the MDSE Research Group at the University of Isfahan.

Gerson Sunyé is an associate professor at the University of Nantes in the domain of software engineering and distributed architectures and the head of the Nantes Software Modeling Group. He received the Ph.D. degree in Computer Science from the University of Paris 6, France, in 1999. From 1999 to 2001 he was a postdoctoral researcher at the IRISA Computer Science laboratory. He has 4 years of industry experience in software development. He received his Habilitation in 2014. He is author of several papers in international conferences and journals in software engineering. His research interests include software testing, design patterns and large-scale distributed systems.