UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Software Engineering Curriculum

Orkhan Jamaladdinov

Systematic Literature Review on the Extent of Conflict Identification and Resolution Research and Practice in Requirements Engineering of Socio-technical Systems: Where are we now?

Master's Thesis (30 ECTS)

Supervisor: Ishaya Peni Gambo, PhD

Tartu 2022

Systematic Literature Review on the Extent of Conflict Resolution Research and Practice in Requirements Engineering of Socio-technical Systems: Where are we now?

Abstract:

Conflict identification and resolution are fundamental requirements engineering (RE) issues that can positively affect many application areas, especially in sociotechnical systems (STSs). For this reason alone, requirements conflicts are a crucial area of study for the RE community, particularly as stakeholders have different views and concerns and frequently pursue mismatching goals. However, despite the increasing studies in theoretical and experimental research for identifying and resolving conflicts during the RE process, a systematic study for evaluating and aggregating the extent of research efforts from both practitioner and research perspectives is missing. Therefore, this research provides a comprehensive review of the significant efforts undertaken along this line of research by investigating and summarizing findings in the context of the formulated research questions. This is with the view of having a clearer and broader perspective on how conflicts in the RE process are handled in practice to improve the quality of STSs and achieve stakeholders' satisfaction. We use search terms with relevant keywords to identify the main research relating to conflict identification and resolution from journal articles, conference papers, seminars, symposiums, book chapters, and IEEE bulletins. Overall, we extracted 1720 papers from the literature search after running several queries, and after applying the exclusion, inclusion, and quality criteria, 66 articles were selected for the research. The study proves that some research has been done on conflict management, but there are fewer practices in conflict resolution than identification. The results also found that stakeholders are not actively involved in decision-making. Also, the existing practices have a bunch of drawbacks, such as requiring expertise and not being completely automated. It is concluded that, even though there is some downside to current practices, they are actively used in practice to discover and handle conflicts. Moreover, new techniques have been implemented to identify and/or resolve conflicts in RE and also to avoid existing practice gaps.

Keywords:

conflict identification, conflict resolution, requirement engineering, systematic literature review

CERCS: P170 Computer science, numerical analysis, systems, control

Süstemaatiline kirjanduse ülevaade konfliktide tuvastamisel ja lahendamisel kasutusel olevate uurimistööde ja tavade määrast tarkvaraarenduse nõuete analüüs sotsio-tehnilistes süsteemides. Milline on hetkeseis?

Lühikokkuvõte:

Tarkvaraarenduse nõuete analüüs (NA) olulisteks osadeks on konfliktide tuvastamine ja lahendamine. Konfliktide õigeaegsel avastamisel ja lahendamisel on rakendust erinevates valdkondades, eriti sotsio-tehnilistes süsteemides. Sel põhjusel on nõuete vaheliste konfliktide uurimine ka tähtsal kohal nõuete koostamise arendamisega tegelevates ringkondades. Põhjalikuma uuringu vajadust rõhutavad ka olukorrad, kus sidusrühmadel on erinevad arusaamad lahendusest ning sellel põhjusel tihti püstitatakse konfliktseid eesmärke. Nii teoreetilisest kui eksperimentaalsest arengust nõuete koostamise uuringute valdkonnas vaatamata, on puudujääk süstemaatilisest uuringust, mis koondaks ja hindaks üheselt teoreetilisi ja eksperimentaalseid uuringuid. Antud uurimistöö koondab erinevad uuringud ning annab laiaulatusliku ülevaate tarkvaraarenduse NA valdkonnast. Antud töö on püstitatud sotsio-tehniliste süsteemide kvaliteedi parandamise vaatenurgast, rõhuga sidusrühmade rahulolu parendamisele. Seda eesmärgiga omandada laiemat ja põhjalikumat vaatenurka NA protsessile. Eesmärgi saavutamiseks kasutatud materjali kogumiseks on kasutatud NA protsessile vastavaid otsingutermineid. Kasutatud materjalide alla kuuluvad artiklid, konverentsiettekanded, seminarid, sümpoosiumid, raamatupeatükid ja IEEE publikatsioonid. Algse uuringu tulemusel eraldati 1720 allikat, mille hulgast peale detailsemate nõuete rakendamist valiti 66 allikat. Antud töö tulemused näitavad, et kuigi konfliktihalduse valdkonda on mõneti uuritud, on vähem teada olevaid lahendamine praktikaid kui tuvastatud konflikte. Samuti näitavad antud töö tulemused, et sidusrühmad ei ole aktiivselt seotud otsuste vastuvõtmisega. Olemasolevatel tavadel on ka mitmeid puuduseid, nagu olemasoleva kompetentsi nõudmine ning puudujäägid täieliku automatiseeringu rakendamisel. Kokkuvõtvalt näitab antud uuring, et sõltumata mõningatest puudujääkidest on hetke olemasolevad tavad laialt kasutuses, ning tarkvaraarenduse NA valdkonda võetakse pidevalt kasutusele uusi tehnikaid eesmärgiga vähendada olemasolevaid lünki praktikas.

Võtmesõnad:

konfliktide tuvastamine, konfliktide lahendamine, nõuete analüüs, süstemaatiline kirjanduse ülevaade

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Contents

1 Introduction

Requirements engineering (RE), which deals with gathering the needs and goals of stakeholders to define, record, and maintain requirements, is one of the most important parts of creating a software project [1]. The stakeholders' involvement is essential while defining the requirements but handling stakeholders' different wishes and objectives can result in conflicting or inconsistent requirements [2]. If the conflicting stakeholders' requirements, desires, expectations, goals, and views are not adequately handled promptly, the resultant effect will engender dissatisfaction, eventually leading to system failure. Notably, that can be expensive and a waste of time [3]. In addition, it will disappoint stakeholders, and they will lose confidence in the product [2]. Thus, identifying and resolving conflicts in the early stage of the RE process, especially at the elicitation and analysis phase, is one of the essential aspects of the software development process.

In the literature, there are increasing studies in theoretical and experimental research for identifying and resolving conflicts during the RE process. Because of that, it is essential to use a systematic strategy to evaluate and aggregate the extent of research efforts done so far in this direction from both practitioner and research perspectives. Therefore, in this research, we provide an overview of the efforts researchers have made in addressing conflicts that arise in RE. The systematic approach seeks to engender a thorough analysis of existing studies published to identify reliable literature sources containing the techniques and practices proposed, developed, and implemented to achieve the research goal.

Also, the research seeks to abridge and explicate the existing evidence concerning (i) the specified conflict identification and/or resolution techniques in RE, (ii) the current practices in the industry for identifying and resolving conflicts in a software development project, (iii) the current practices that support stakeholders' decision-making process amidst conflicting situation, (iv) the gaps or drawbacks of existing techniques (iv) the classification of these techniques if any, (v) the associated Software Engineering(SE) processes; and (vi) the collective and increasing interest of researchers and practitioners in addressing conflicts arising during a development project. To the best of our knowledge, few or no existing SLRs focus on conflict identification and resolution concerning the development of socio-technical systems (STSs) from the RE perspective.

1.1 Research Goal

The overarching goal of this research is three-fold. First, to find out existing conflict management research and practices, to create an overview of these practices, and also define the taxonomy between noted procedures. Secondly, to present a fair and impartial empirical overview of the research evidence in reality and encourage further research and development for a more robust approach to conflict management. Finally, to demonstrate the essential areas software engineers and requirements engineers need to enhance in handling conflicts during requirements elicitation and analysis to create high-quality software. To address the goals mentioned above, we reviewed the literature.

1.2 Research Contribution

The current research goal is to get an overview of existing practices in conflict management, which is a gap in a research area. The thesis can be an excellent sample for the research community to understand and appreciate the extent of work done focusing on conflict identification and resolution concerning the development of STSs from the RE perspective. In addition, the study can be used to implement new practices for conflict identification and/or resolution with more benefits by taking into account a summary of existing approaches and downsides of these practices from the results. In practice, the RE practitioners and researchers can leverage this study to find appropriate methods or techniques for identifying and resolving conflicts.

1.3 Thesis Outline

The remaining part of this thesis is organized as follows. Chapter 2 shows details of current facts in similar or related works. Details of the research process such as research approaches, questions, strategy, criteria, data synthesis, and extraction are represented in Chapter 3. Chapter 4 discusses threats to the validation of the work. The analysis and results from selected research papers are described in Chapter 5. The results are discussed in Chapter 6. Findings from this research are specified in Chapter 7. Finally, Chapter 8 is about the conclusion and future work.

2 State of the art

The analysis of the existing SLR related to conflict identification and resolution in RE was done before starting the research. However, it has been found that there is a significant gap in the research area related to that topic. There was no similar research for conflict identification and/or resolution in the RE community. That makes our research more valuable.

Only one review has been done by Aldekhail et al. [4]. Still, it is not precisely systematic in nature, as research questions, approaches, and queries are not defined to justify what is expected of a systematic review based on the guidelines in [5], [6], [7]. Unfortunately, the purpose of the review in [4] was to find a summary of existing research on conflict identification and its limitations. Even with that, the study only covers the years from 2001 to 2014 and only included 20 papers. On the downside of that research, recent research in the last 10 years was not covered. The study has found only a limited number of existing approaches, thereby excluding all possible existing practices.

Moreover, the study in [4] classified existing solutions such as manual, automatic, and general frameworks. Compared with our study, previous reviews did not cover current practices, especially in the last decade. Also, the few studies in this direction from literature did not focus on stakeholders' involvement and did not clearly describe setbacks of the existing practices in a more pragmatic way. Moreover, it is impossible to understand which software engineering processes the current approaches to identifying and resolving conflicts are associated with. Differently in this thesis, we extended our review to provide coverage of the recent years and described the new practices available for managing conflicts in the RE process.

3 Methodology

This thesis' methodology section describes the procedures used to conduct the systematic literature review (SLR). In this context, SLR provides an organized, repeatable method for evaluating, understanding, analyzing, comprehending, and summarizing all available research relevant to a particular research question, topic area, or phenomenon of interest [8], [9]. This method was observed by Kitchenham [10], and Biolchini et al. [5].

We follow the three-step process of performing an SLR: *planning, execution (i.e., carrying out the research), and result analysis* [10]. Figure 1 presents the process's pictorial representation, as further refined by Biolchini et al. [5]. As Figure 1 reflect, Biolchini et al. [5] proposal leverage Kitchenham's strategy for conducting a systematic review where the *"packaging"* process was included to store all the results obtained in the first three steps proposed by Kitchenham [10]. In this thesis, we use the *"planning"* phase to engender the list of research objectives and define the protocols used for the review. The formulated research questions and the execution strategies for the review are laid forth in the protocol. Finding primary studies, choosing and evaluating them in accordance with the inclusion and exclusion criteria outlined in the review protocol, and executing them are all parts of the *"execution"* stage. Finally, the data from the primary investigations are retrieved and summarized during the *"result analysis"* phase. All the results acquired are simultaneously saved in the *"packaging"* phase. Biolchini et al. [5] observed that after completing the planning and execution phases, two checkpoints depicted by the diamond symbols in Figure 1 are used for detecting possible errors.

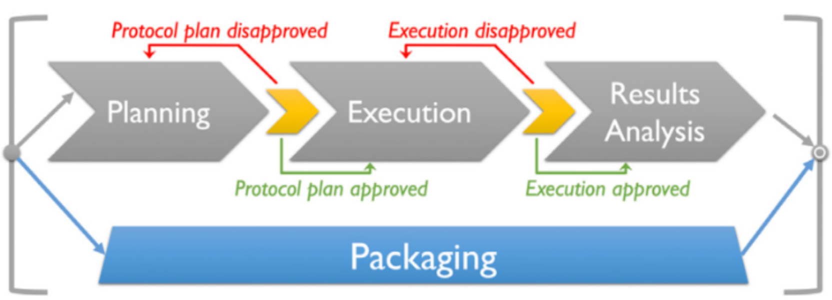

Figure 1. The systematic review process based on [10] and [5]
Source: [8]

Notably, this section goes into detail about the research approach, study-specific research questions, search strategy, study selection and quality assessment, data synthesis, and data extraction techniques.

3.1 Research Approach

We adopted the guidelines and review protocols for conducting systematic literature review in software engineering [5], [6], [7], as shown in the stages of review protocol in Figure 2. The thesis systematically identified, analyzed, and interpreted studies on conflict identification and resolution in the requirements engineering process for developing STSs. As Figure 2 reflects, we defined the aim of the review, which the formulated research questions achieved. In essence, the research questions are formulated based on the aim of this study. Next is identifying a search strategy, where search terms and electronic sources (research databases or resources) were specified. After that, we designed search strategies based on the research questions formulated. The study selection followed the identified search strategy, where the data extracted were collated. The extracted data were further refined by scrutinizing the titles to determine relevant papers. We then formulated the quality assessment criteria to evaluate the scrutinized papers further. The final process is the data synthesis, where the final list of papers was selected for analysis.

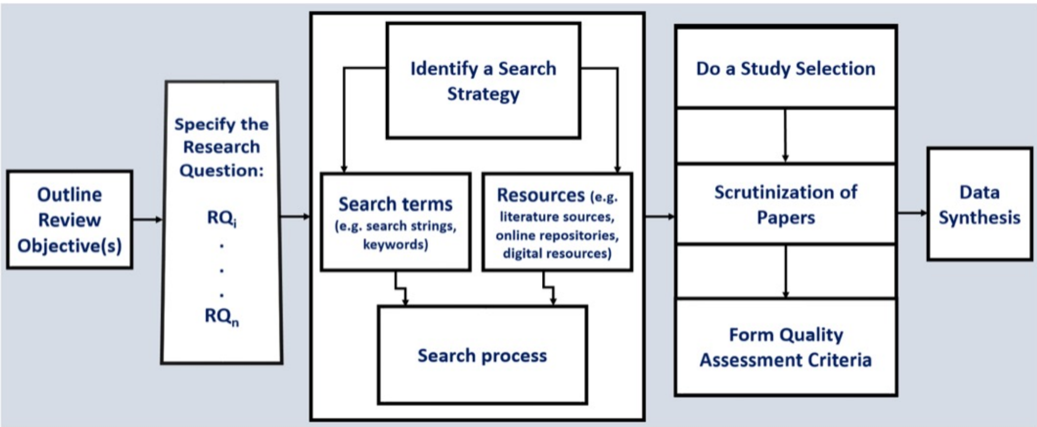

Figure 2. Stages of the review protocol

3.2 Research Questions

The main objective of this SLR is to find articles that have information about existing practices for identifying and/or resolving conflicts in RE. We focus on getting information about stakeholders' involvement in the decision-making process, finding patterns between existing RE solutions, and defining the taxonomy or classification according to these patterns. In addition, the research seeks to determine the most appropriate the Software Development Life Cycle (SDLC) phase existing practices for managing conflicts are associated with. Moreover, the research aspires to get an overview of gaps or setbacks in existing practices. For that, we explore in the study the existence and need of newly offered or developed methods for detecting and/or handling inconsistencies in the various research articles to make informed findings for the software engineering (SE) practitioners, especially in the RE stage. As a result, the following research questions (RQs) are formulated to fulfill the objective of the research:

- RQ1: How much of the existing literature specified conflict identification or/and resolution techniques or practices in RE?
- RQ2: What are the current practices(or already proposed approaches) in the industry for identifying and resolving conflicts in a software project?
- RQ3: Do the current practices involve stakeholders in the decision-making process?
- RQ4: What taxonomy or classification of these techniques exist in the literature?
- RQ5: What are the SDLC phase the existing solutions are associated with?
- RQ6: What are the gaps or setbacks in the literature on the reported techniques/ methods for conflict identification and/or resolution?
- RQ7: What are new practices offered in the papers for identifying and resolving the conflicts?

3.3 Search Strategy

We designed and outlined our search strategies based on the formulated research questions. The focus was on using search terms, search strings (or keywords), and online repositories during the search process. Below is an in-depth explanation of the search processes.

3.3.1 Search strings

The search strings are the keywords identified with which we were able to harvest literature (or download relevant articles) from the online repositories. We followed the criteria outlined in Kitchenham and Charters [6] to formulate our search terms. In particular, we considered the various terms contained in the research questions, the keywords in relevant papers or books, additional spellings and synonyms for significant and related terms, and finally, the use of Boolean operators "OR" and "AND" to either integrate additional spellings and synonyms or establish the connection with main terms. For example, we had the Boolean operation - "resolution OR technique OR method OR practices."

IEEE Xplore Digital Library, Scopus, and ACM Digital Library have been selected as databases to get relevant articles that have the answer to research questions. We adjusted the search query to get better relevant articles from the databases (data sources). The following tables: Table 1, Table 2, and Table 3 show the context of the query for each data source.

Table 1. IEEE Xplore search query

("All Metadata":requirement OR "All Metadata":"requirements engineering") AND ("All Metadata":"conflict resolution" OR "All Metadata":conflict OR "All Metadata":inconsistency OR "All Metadata":dispute OR "All Metadata": disagreement) AND ("All Metadata":identify OR "All Metadata":recognize OR "All Metadata":capture OR "All Metadata":resolve) AND ("All Metadata": resolution OR "All Metadata":technique OR "All Metadata":method OR "All Metadata":practices)

Table 2. Scopus search query

(TITLE-ABS-KEY (requirement OR requirements AND engineering) AND TITLE-ABS-KEY ("conflict resolution" OR conflict OR inconsistency OR dispute OR disagreement) AND ALL (identify OR recognize OR capture OR resolve) AND ALL (resolution OR technique OR method OR practices))

Table 3. ACM Digital library search query

[[Abstract: "requirement"] OR [Abstract: "requirements"] OR [Abstract: "requirements engineering"]] AND [[Abstract: "conflict resolution"] OR [Abstract: conflict] OR [Abstract: inconsistency] OR [Abstract: dispute] OR [Abstract: disagreement]] AND [[All: identify] OR [All: recognize] OR [All: capture] OR [All: resolve]] AND [[All: resolution] OR [All: technique] OR [All: method] OR [All: practices]]

Further, we added "requirement", and "requirements engineering" as keywords to the search query to get relevant articles connected with requirement engineering. To get papers connected with conflict or inconsistency, the following keywords were attached to the query: "conflict resolution", "conflict", "inconsistency", "dispute", and "disagreement." The query also has the "identify", "recognize", "capture", "resolve", "resolution", "technique", "method", and "practices" keywords to get articles that mentioned conflict identification and/or resolution practices.

3.3.2 Literature resources

We identified three (3) electronic sources of relevance to software engineers and the research topic, as observed by Brereton et al. [7] to perform an in-depth search. The electronic sources include the ACM digital library, IEEE Xplore libraries, and Scopus database. More specifically, the search strings were used on the IEEE Xplore, ACM Digital library, and Scopus electronic sources to harvest data (the various articles) for synchronizations. The rationale behind the ACM digital library, IEEE Xplore digital libraries, and Scopus database are to enable the author to provide a more detailed and broader view of researchers' efforts on the extent of research and practical consideration in dealing with conflict identification and resolution in a development project, especially for STSs. Remarkably, the IEEE Xplore, ACM Digital library, and Scopus electronic sources are the major databases to source and extract relevant, reputable, and high indexed articles pertinent to the study. IEEE Xplore and ACM digital library include journal papers, conference proceedings, workshops, symposiums, book chapters, and bulletins in computer science and software engineering SLR studies [6]. Scopus database includes citations of academic articles from ScienceDirect, SpringerLink, and Web of Science. The search duration is from September 2021 to July 2022.

3.3.3 The search process

After running queries in the 3 data sources, one thousand, seven hundred and twenty (1720) papers were found without filtering. Four hundred ninety-four (494) of these articles were from IEEE Xplore, four hundred fifty-nine (459) from Scopus, and the remaining seven hundred and sixty-seven (767) articles were from the ACM digital library.

3.4 Study Selection and Quality Assessment

Overall, 1720 papers were received after running queries. Next, we define the inclusion, exclusion, and quality criteria applied to get more suitable papers and reduce the number by filtering the 1720 papers.

3.4.1 Exclusion Criteria

Seven (7) exclusion criteria (EC) have been defined to eliminate unrelated articles. To get the most recent papers, EC1 was specified. There were books, short papers, and tutorials in the results. EC2 was defined to get only conference materials, journals, articles, or reviews. Based on EC3 and EC4, some papers were not in English, while some were not accessible. Notably, Scopus is the database for research articles, but it was missing some links for the research articles. As a result, EC5 was determined to remove these papers from the received articles. Respectively, EC6 was implemented to remove duplicate. After applying exclusion criteria, nine hundred and ninety-four (994) papers remained.

- EC1: Papers published before 2000 (1534 remaining)
- EC2: Papers that are not conference materials, journals, articles, or reviews (1248 remaining)
- EC3: Papers that are not in the English language (1245 remaining)
- EC4: Papers that are not accessible (1179 remaining)
- EC5: Some Scopus papers were missing links for the full text (1107 remaining)
- EC6: Papers that are duplicates (994 remaining)

3.4.2 Inclusion Criteria

To include the most suitable papers in the selected research papers list, two inclusion criteria (IC) have been specified. The first IC, IC1, aims to include only articles with details about conflict identification or/and resolution techniques or practices. Some articles offered new conflict management practices without comparing or mentioning previous approaches. For that, we define the second IC, IC2, to acquire papers that describe recent articles and also mention existing methods. After filtering with the two IC, sixty (60) papers were left.

- IC1: Papers that have information about conflict identification or/and resolution techniques or practices
- IC2: Papers that offer new conflict identification or/and resolution techniques or practices(but at the same time mention or compare previous techniques, approaches, methods or practices)

3.4.3 Quality Criteria

After implementing exclusion and inclusion criteria, one quality criteria (QC) was applied to get the best-qualified paper for the research.

- QC1: Are the papers connected to the topic as their title, abstract, conclusion, or related (i.e., earlier) work section?

As a result, 60 papers were selected after applying all the criteria. Six (6) papers that were found or attracted attention in the other articles' references were manually added to the list of the selected papers. In total, sixty-six (66) papers were chosen for the research.

Further, we formulated five (5) quality assessment questions (QAQs) based on the RQs to help score the selected papers. In particular, the QAQs were used to analyse the selected papers' integrity, comprehensiveness, and applicability. To score each paper, we provided a weighting or scoring strategy. For that, each of the QAQs was judged with the following answers: "Yes", "Partly", or "No". Consequently, "Yes" = 1, "Partly" = 0.5, and "No" =0. Thus, the grades used for the scoring are either 0, 0.5, or 1. For example, if the QAQ is not addressed in any of the selected papers, the score will be 0, which means "No". If it is partially addressed, the score will be 0.5, meaning it is "Partly" addressed. Finally, if fully addressed, the score will be 1, which means "Yes". This was applied to all the final selected papers. The quality score results are based on the QAQs for each of the selected papers, and the total sum of these scores is in the Appendix section depicted in Table 26. The goal of scoring the papers is to prove that the papers used to answer the RQs are relevant and contextual and cover the elements associated with the RQs. The QAQs include:

- QAQ1: the existing solutions for identifying and/or resolving conflicts clearly mentioned in the paper?
- QAQ2: Is stakeholders' participation in the current practice clearly reported?
- QAQ3: Are the papers clearly reporting which SDLC phase the practice is suitable?
- QAQ4: Are the gaps or setbacks clearly described?
- QAQ5: Are newly offered conflict management practices clearly explained?

3.5 Data Synthesis

As mentioned in the search process, we initially got 1720 papers after applying our search queries in the 3 data sources. After that, we used the inclusion, exclusion, and quality criteria to get valuable papers for the research.

Table 4 shows the number of papers remaining in each data source and the total after applying each inclusion and exclusion criteria. After running the search queries, we got 494 papers from IEEE Xplore, 459 from Scopus, and 767 from ACM digital library, respectively. After applying all criteria, 31 articles were selected from IEEE Xplore, 22 from Scopus, and 7 from ACM digital library, and the total came to 60 articles.

Table 4. Papers selection process

Filter	IEEE Xplore	Scopus	ACM Digital library	Total
Total	494	459	767	1720
EC1	428	418	688	1534
EC2	413	377	458	1248
EC3	413	375	457	1245
EC4	410	313	456	1179
EC5	410	241	456	1107
EC6	408	166	420	994
IC1	52	59	49	160
IC2	31	22	7	60
QC1	31	22	7	60
Selected	31	22	7	60

In Figure 3, we have visualised the entire elimination process presented in Table 4. Consequently, Figure 3 gives an overview of the study selection process, indicating the number of papers left in total after applying each criterion. In total, 60 articles were left after the filtering and scrutinization process of removing unrelated papers and identifying redundancy. 6 papers from IEEE Xplore were manually added to the list, which makes a total of 66 papers selected for the review and analysis. As Figure 3 shows, 37 of these 66 articles were from IEEE Xplore, 7 were from ACM digital library, and 22 were selected from Scopus. Notably, Scopus is a database for the research articles where we got 8 out of 22 Scopus articles from SpringerLink, 6 out of 22 were from ScienceDirect, 5 out of 22 were from IEEE Xplore, and 3 out of 22 were from other sources.

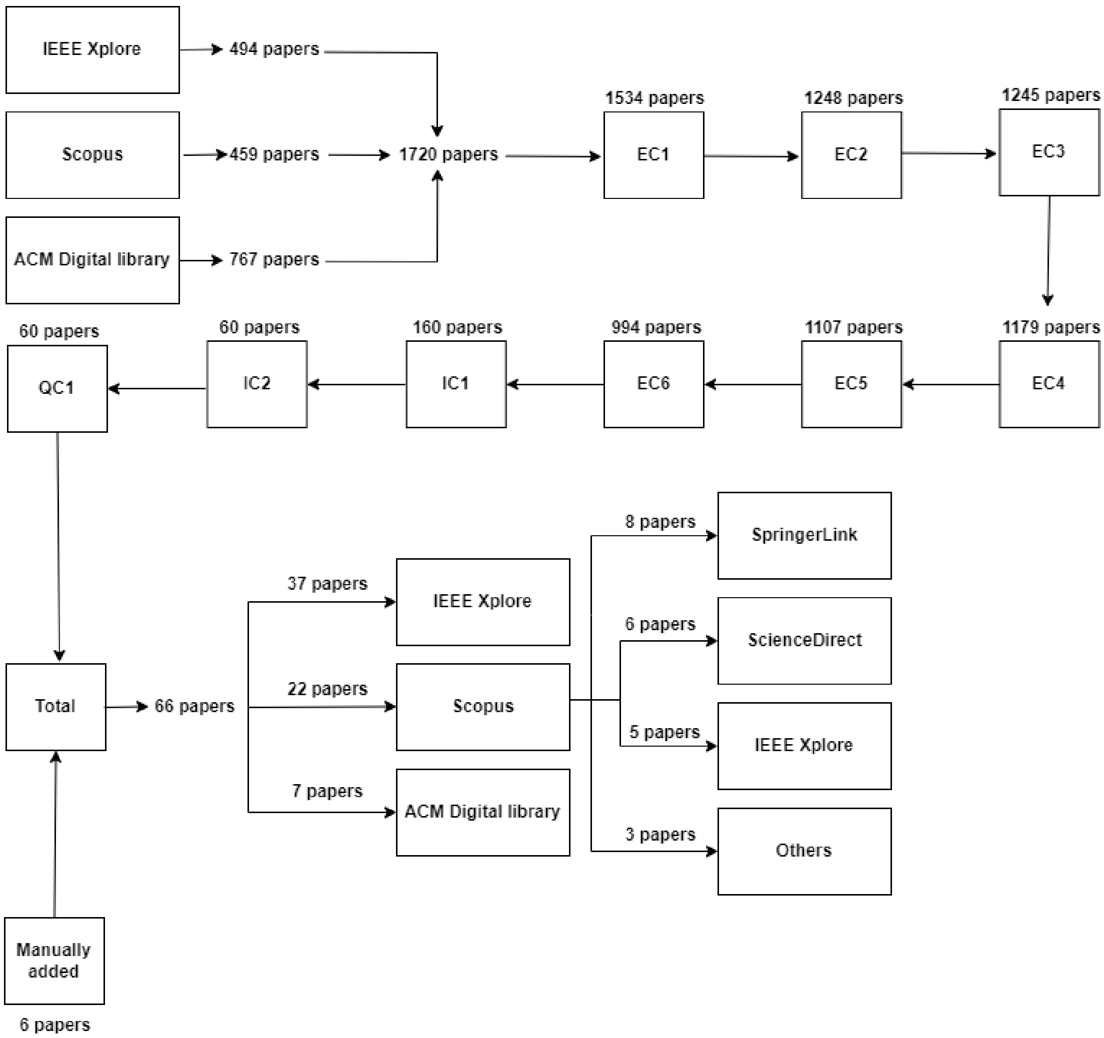

Figure 3. An overview of the study selection process

3.6 Data Extraction

Data extraction form was concocted while reading several resources relevant to the research. Each resource had its fair share in permeating clarity to conflict identification and/or resolution practices in RE. The provided form(see Table 5) contains several paramount factors for resource differentiation purposes. These factors are as stated: identifier, article link, article title, author name(s), publication year, existing practices, stakeholders' involvement, taxonomy or classification, software engineering process, gaps or setbacks, and new practices.

Useful information from the selected papers are combined in the Data Extraction Spreadsheet [1]. To answer defined research questions, the results in the spreadsheet will be used.

As shown in the Data Extraction Spreadsheet, some factors have responses to questions of overall content, and some only have responses for a small number of questions. For example, factors such as Identifier, Article link, Article title, Author name(s), Publication years, and Existing practices are answered in all selected papers. In contrast, factors such as Stakeholders' involvement, Software engineering process, Gaps or setbacks, and New practices are responded to only a particular number of research papers.

Table 5. Data Extraction Form

Data Source	**Value**	**RQ**
Identifier	Unique identification number	
Article link	Link to article	
Article title	Name of the article	
Authors name(s)	Set of names of the authors	
Publication Year	The year article was published	RQ1
Existing practices	Current practices for identifying and resolving conflicts or inconsistencies	RQ2
Stakeholders' involvement	Stakeholders' involvement in decision-making process	RQ3
Taxonomy or classification	Taxonomy or classification of existing techniques	RQ4
Software engineering process	Software engineering processes which existing solutions are associated with	RQ5
The gaps or setbacks	The gaps or setbacks of existing techniques	RQ6
New practices	New practices offered in the papers for identifying and resolving	RQ7

[1] Data Extraction Spreadsheet details can be found here

4 Threat to Validity

This section is about the discussion of threats to the validity of our research. We have discovered 3 threads, and detail can be found below.

- *Lack of access to related articles:* One burden in SLR is getting all related articles from the defined and selected repositories, which is almost impossible. However, this was tried to handle by defining the research approach, questions, and strategy carefully and clearly to receive all related articles. Specifically, 6 papers are manually added to the research to get the best result.

- *Limited number of articles:* The number of the selected 66 papers used for the SLR can be considered small. However, the research approach, questions, and strategy have been defined carefully and clearly to receive as many articles as possible. Notably, not much work has been done on conflict management, especially identifying and resolving conflicts; as a result, we got that amount of literature and could further make informed analysis to attract research attention in the SE and RE community.

- *Reliability of the research:* The final threat to validity borders on justifying the dependability of the research. The big question is, will a similar SLR get the same result in this direction? To mitigate this, we provided a detailed description by strictly following the guideline for conducting SLR in SE. Regarding that, we explain the research approach required to fit the scope of the study and achieve the research goal. In addition, we formulated the research questions necessary and described the study selection process, quality assessment criteria, and data synthesis. Optimistically, similar research, if conducted, will get matching results.

5 Analysis and Results

In this chapter, we present the results of the SLR based on the formulated research questions.

5.1 RQ1: Overview for existing solutions

Overall, sixty-six (66) papers were selected for the research. Figure 4 shows the number of papers published each year.

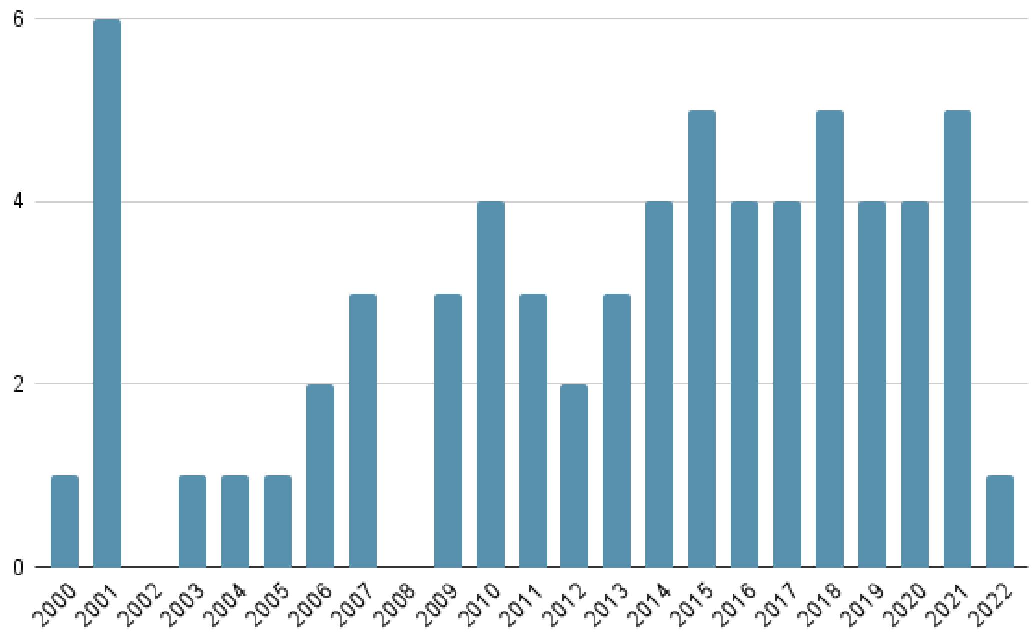

Figure 4. Number of articles published each year

5.2 RQ2: Current practices for identifying and resolving conflicts or inconsistencies

There are quite many practices for identifying and resolving conflicts or inconsistencies. We have identified different terminologies used in the literature and industry for qualifying these practices. They include *tools, approaches, frameworks, techniques, methods, methodologies, models or modelling languages, and other practices*. The following subsections describe the different terminologies.

5.2.1 Tools

Tools are divided into 3 different groups by their purposes, such as conflict identification, resolution, and both identification and resolution tools.

Table 6 shows the name of the tools that are used to identify conflict in requirement engineering. 22 of 66 papers mentioned 28 different tools that are developed or offered for the identification of tools. Tools are sorted from most used to less-used in the table. Only 33% of articles had information about inconsistency detection tools. Remarkably, only one-third of the papers mentioned tools for conflict detection, but considering every aspect, there were quite enough tools in practice for this purpose.

Table 6. Tools for conflict identification

Tool	**Solution**	**Reference**	**Number**
QARCC	identify	[11], [12], [13], [14]	4
SpeAR tool (Specification and Analysis of Requirements)	identify	[15], [16], [17]	3
QuARS	identify	[18], [19], [20]	3
EA-Analyser tool	identify	[18], [21], [22]	3
BTC Embedded Platform tool	identify	[16]	1
MaramaAIC tool (Automated Inconsistency Checker)	identify	[17]	1
SREE	identify	[18]	1
Requirements Scout	identify	[20]	1
QVscribe tool	identify	[20]	1
HILA	identify	[21]	1
REInDetector	identify	[21]	1
UC (use case)supporting tools	identify	[21]	1
Software Cost Option Strategy Tool (S-COST)	identify	[23]	1
Inconsistency detection tools	identify	[24]	1
VERIDEV	identify	[25]	1
VUML	identify	[25]	1
CDET	identify	[25]	1
VIEWINTEGRA	identify	[25]	1
Requirements Analysis Tool (RAT)	identify	[26]	1
Computer-Assisted Requirements Evolution Toolset (CARET)	identify	[27]	1
Argosim	identify	[28]	1
SAVELife	identify	[29]	1
JITTAC	identify	[29]	1
ArchTrace	identify	[29]	1
ACTool	identify	[29]	1
jRMTool	identify	[29]	1
WikiWinWin	identify	[30]	1
XML file consistency checking tool	identify	[31]	1

Furthermore, there were a few papers that mentioned tools for the resolution of conflicts. 3 different tools are mentioned in 3 different papers. Table 7 provides more details about resolution tools. In practice, there is a relatively limited number of tools for conflict resolution.

Table 7. Tools for conflict resolution

Tool	**Solution**	**Reference**	**Number**
Synoptic tool	resolve	[2]	1
HERMES	resolve	[32]	1
Design advice tool	resolve	[33]	1

Some tools can be used for identifying and resolving conflicts as well. Table 8 provides details of these tools. 10 tools are mentioned in 9 articles. The tools are sorted from most mentioned to less mentioned in the table. There are several tools that are used in practice for both conflict identification and resolution. However, the number of them more is than resolution tools.

Table 8. Tools for conflict identification and resolution

Tool	**Solution**	**Reference**	**Number**
Software Cost Option Strategy Tool (S-COST)	identify and resolve	[12], [34]	2
QARCC	identify and resolve	[34], [35]	2
Tradeoff analysis tools	identify and resolve	[36], [37]	2
Distributed Collaboration Priorities Tool (DCPT)	identify and resolve	[36], [37]	2
AS-SERT™ tool	identify and resolve	[15]	1
HERMES	identify and resolve	[38]	1
gIBIS (graphical IBIS)	identify and resolve	[38]	1
Teleologic DOORS	identify and resolve	[38]	1
QuestMap	identify and resolve	[38]	1
Analytical tool	identify and resolve	[39]	1

Figure 5 shows the comparison of tools by their purpose. There are 28 tools for identification, 3 for resolution, and 10 for identification and resolution of conflicts. As shown in Figure 5, most of the tools are developed for conflict detection or identification. One-fourth of the tools are useful for both inconsistency identification and handling. Only 7.3% of tools mentioned in the articles are for resolving conflicts.

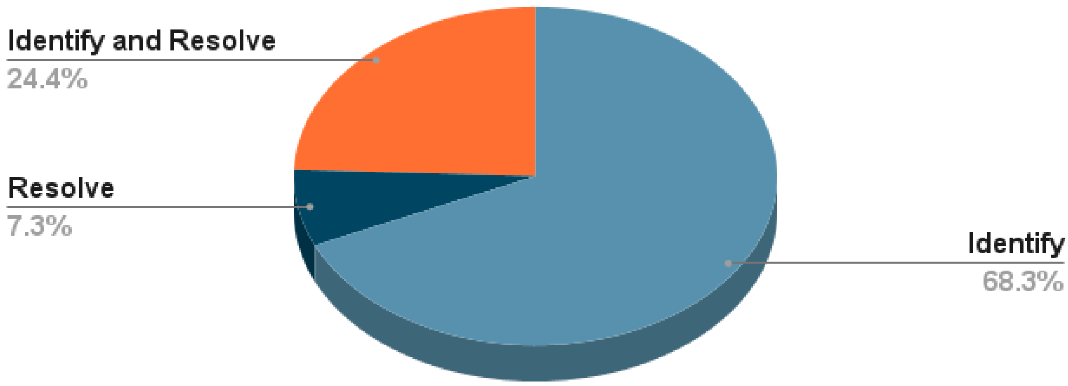

Figure 5. Comparison of existing tools by their purposes

5.2.2 Approaches

Similar to what we have in subsection 5.2.1, we classify the approaches based on their purpose of use. For that, we grouped them into identification, resolution, and both identification and resolution, as shown in Tables 9, 10, and 11.

Table 9 provides information about approaches mentioned in papers for conflict identification. We observed that some papers mentioned approaches without specifying their direct name. These approaches are called *"other approaches"* in Table 9. In total, 23 papers had details about 22 approaches. Results are sorted from most used to less used. Only 34.8% of articles had mentioned approaches for conflict detection. Table 9 proves that many approaches for conflict identification exist in the literature. This means much effort has been put into identifying conflicts during the RE process of STSs.

Table 9. Approaches for conflict identification

Approach	Solution	Reference	Number
Ontology based approaches	identify	[19], [26], [40], [41]	4
Pattern-based approach	identify	[42], [43], [44]	3
Natural Language processing(NLP) Based approach	identify	[19], [44]	2
Automated approaches	identify	[3]	1
Review approach	identify	[16]	1
Rule-based approach	identify	[18]	1
Statistical approaches	identify	[18]	1
Model composition mechanism	identify	[21]	1
Dependency Structure Matrices	identify	[29]	1
Source Code Query Languages	identify	[29]	1
Exception handling approach	identify	[33]	1
Tableaux-based approach	identify	[42]	1
Word2Vec approach	identify	[44]	1
Goal-conflict analysis approach	identify	[45]	1
Win Win approach	identify	[46]	1
Conventional approaches	identify	[47]	1
Use case/ Scenario based Approach	identify	[48]	1
Adaptable approaches	identify	[48]	1
Composition Frame approach	identify	[49]	1
Process-NFL approach	identify	[50]	1
CDNFR approach	identify	[50]	1
Other approaches	identify	[22], [30] [43], [51]	4

In contrast to approaches for conflict identification, only a few papers mentioned approaches for conflict resolution in the literature. Table 10 provide detailed information about these approaches. Thus, only 3 papers had information about 3 different approaches to resolving conflicts. This means that researchers and practitioners have put in fewer efforts in terms of the approaches for resolving conflicts in the RE process of STSs.

Table 10. Approaches for conflict resolution

Approach	Solution	Reference	Number
Win Win approach	resolve	[2]	1
Component based Requirements Engineering Approach	resolve	[48]	1
Rule-based approach	resolve	[52]	1

Further, 11 approaches are mentioned in 15 articles for both identification and resolution of conflicts in the literature. Table 11 gives a detailed overview of these approaches. Approaches in which the names are not clearly identified in papers are grouped under the *"other approaches."* The approaches are sorted from most noted to less noted. Compared to identification approaches, fewer approaches are suitable for both identification and resolution simultaneously. However, the number of them is way more than the resolution approaches.

Table 11. Approaches for conflict identification and resolution

Approach	Solution	Reference	Number
Win Win approach	identify and resolve	[12], [34], [35] [36], [37], [53]	6
Natural Language processing(NLP) Based approach	identify and resolve	[20], [40]	2
Ontology based approaches	identify and resolve	[28], [54]	2
Oz approach	identify and resolve	[35]	1
Resilient design approach	identify and resolve	[39]	1
Environment-based design (EBD)	identify and resolve	[39]	1
Viewpoint Oriented Approach	identify and resolve	[48]	1
Goal Oriented Approach	identify and resolve	[48]	1
Multidimensional Separation of Concern	identify and resolve	[48]	1
Hybrid planning approach	identify and resolve	[55]	1
Other approaches(the exact name was not mentioned)	identify and resolve	[30], [56]	2

In Figure 6, we show the comparison of approaches by their purpose. Most of the approaches noted in the literature are for identifying conflicts. Approaches for identification and resolution are almost 2 times less than the identification approaches. Unfortunately, there are few approaches in practice that are used for the resolution of conflicts.

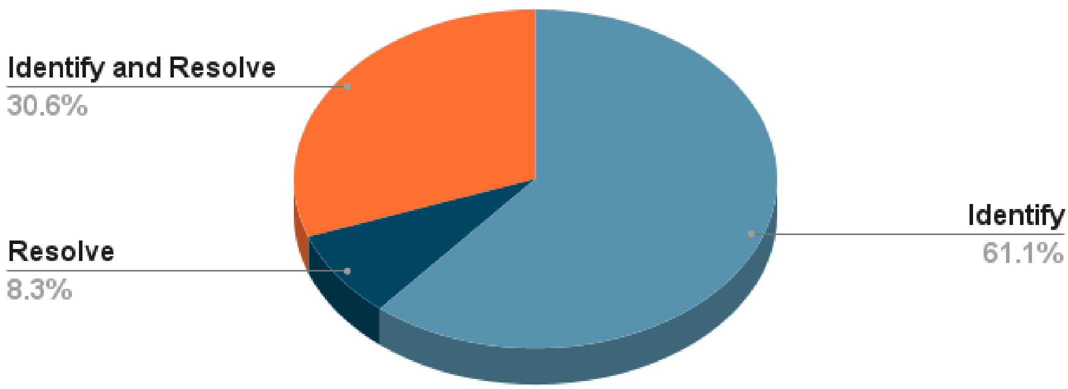

Figure 6. Comparison of existing approaches by their purposes

5.2.3 Frameworks

Another main terminology used in practice for conflict identification and/or resolution is frameworks. Compared with tools and approaches, less work has been done in frameworks field. That's why the results are combined in Table 12. Table 12 shows detailed information about the frameworks used for conflict identification and/or resolution in the literature. Noticeably, the Non-Functional Decomposition (NFD) framework is mentioned as a resolution framework in one paper and as an identification framework in another. That's why this framework is mentioned in one row, but both of the purposes - identify and resolve- are specified in the solution column. Some papers mentioned conflict identification and/or resolution frameworks used in practice without specifying their exact name. As a result, these frameworks are grouped under *"other frameworks."* The results are sorted from most-used to less-used. In addition, "other frameworks" are placed at the end of the table. In total, 13 frameworks are noted in 23 papers.

Table 12. Frameworks for conflict management

Framework	Solution	Reference	Number
Non-Functional Requirements (NFR) framework	identify	[24], [46]	2
Non-Functional Decomposition (NFD) framework	resolve identify	[23] [32]	1 1
KAOS goal modelling framework	identify	[19], [57]	2
UML-based transformation framework	identify	[21]	1
Repair framework	identify	[21]	1
Viewpoints and Inconsistency Management (VIM) framework	identify	[24]	1
FLAME framework	identify	[29]	1
PBURC framework	identify	[44]	1
Probabilistic framework	identify	[45]	1
PERSUADER framework	resolve	[58]	1
A formal graph based framework	resolve	[59]	1
MAVO framework	identify	[60]	1
Other frameworks (the exact name not mentioned)	identify identify and resolve resolve	[27],[35],[53] [61], [62], [63] [52], [55], [64] [32]	6 3 1

Figure 7 is created to show the comparison of frameworks by their purposes in practice. As in tools and approaches, more work has been done for conflict identification. One-fourth of the frameworks are for conflict handling, which is much better than tools and approaches. However, 6.3% of frameworks are appropriate for conflict identification and resolution.

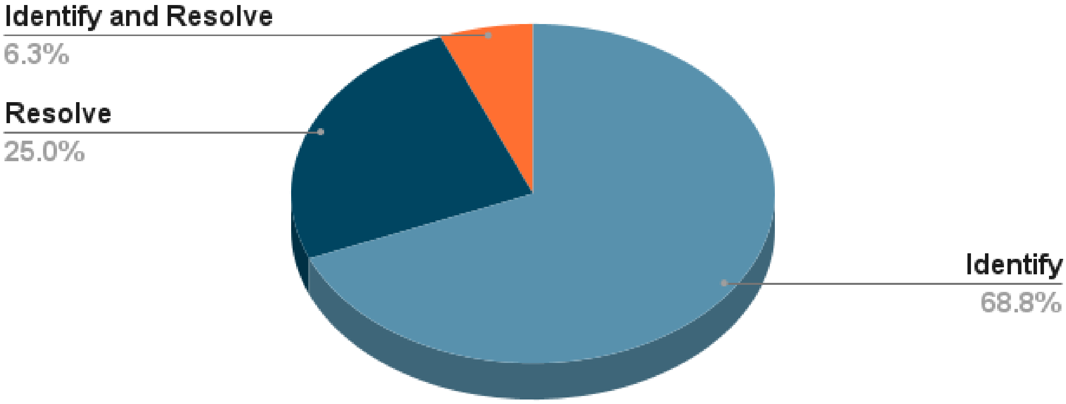

Figure 7. Comparison of existing frameworks by their purposes

5.2.4 Techniques

Additional terminology that is preferable in the industry is the techniques. Several of them are noted in the selected articles used during the SLR.

The techniques that name wasn't specified in the literature are combined under "other techniques." 18 techniques are mentioned in the 17 papers. Table 13 provides a detailed summary of techniques. They are sorted from more noted to less noted, and *"Other techniques"* is placed in the last row. Only 25.7% of papers had information about techniques which proves that it is not one of the standard practices in the industry for conflict identification and/or resolution.

Table 13. Techniques for conflict management

Technique	Solution	Reference	Number
Trace analyzer of the requirements traceability	identify	[11], [13]	2
A technique that adopts a hierarchical constraint logic programming approach	identify	[11], [13]	2
Manual technique	identify	[3]	1
NLP technique	resolve	[20]	1
Model-based validation and inconsistency detection technique	identify	[22]	1
Unified Modeling Language (UML)-Based technique	identify and resolve	[40]	1
Analysis of Competing Hypotheses technique	resolve	[43]	1
Latent Semantic Indexing (LSI) technique	identify	[44]	1
Pareto front technique	identify	[46]	1
Multiple Criteria Decision Making (MCDM) technique	identify	[49]	1
VWPL(Viewpoints Language) technique	identify	[53]	1
Conceptual modeling technique	identify	[58]	1
Techniques to systematically transform goals	resolve	[58]	1
Automated traceability technique	identify	[61]	1
Regressing Negated Assertions technique	identify	[65]	1
Using Divergence Patterns technique	identify	[65]	1
Theorem-proving and model-checking technique	identify	[65]	1
Other techniques(the exact name was not mentioned)	identify identify and resolve	[19], [51] [56] [3]	3 1

Figure 8 provides information about the comparison of techniques by their objectives—approximately three-fourths of the techniques mentioned in the papers are for identifying conflicts. Only a few works have been done in conflict resolution and conflict identification and resolution.

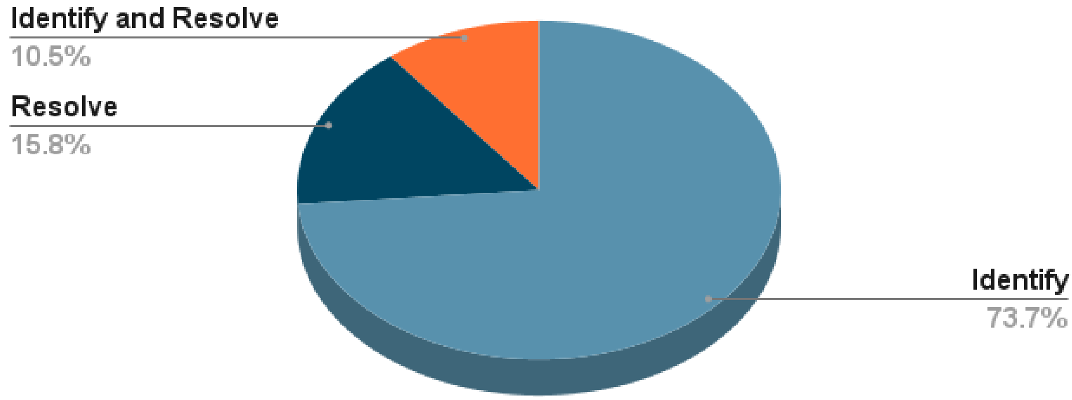

Figure 8. Comparison of existing techniques by their purposes

5.2.5 Methods

Methods are another terminology beneficial to practice for identifying and/or resolving inconsistency in the literature.

Some of the methods weren't clearly named in the papers. As a result, we mentioned them as "other methods." The negotiation method is noted as a resolution method in several papers. However, one of the articles [61] suggested it as an identification method. Informal or semi-formal methods are specified as conflict identification methods in some papers, but in one of the articles [66] are recognized as a resolution method.

We sorted the result from most to less used, and "other methods" were used to indicate methods without a defined name. We found 23 methods mentioned in 23 papers, which is around 34.8% of articles. It explains that methods are one of the preferable practices in the RE process of STSs at the industry level. A detailed overview of methods is shown in Table 14.

Table 14. Methods for conflict management

Method	Solution	Reference	Number
Negotiation method	identify resolve	[61] [2], [22],[33], [58], [67], [68]	1 6
Informal or semi-formal methods	identify resolve	[43], [51] [66]	2 1
Formal method	identify	[19], [69]	2
Prioritization method	resolve	[59], [65]	2
Intelligent method	identify and resolve	[3]	1
Multiple Criteria Decision Making method	identify	[22]	2 1
Semi-automatic method	identify	[28]	1
Hierarchy method	resolve	[32]	1
REMAP (REpresentation and MA-) intenance of Process knowledge)	identify and resolve	[38]	1
EVOLVE release planning method	identify	[57]	1
CBAM architecture decision method	identify	[57]	1
Search-based method	identify	[57]	1
Analyze obstacles method	resolve	[58]	1
Cooperative method	resolve	[66]	1
Competitive method	resolve	[66]	1
Thirdparty method	resolve	[66]	1
Risk analysis method	identify	[69]	1
ReCAPs method	identify and resolve	[70]	1
Description Logic based method	identify	[70]	1
Logic-based method	identify	[71]	1
Goal-based method	resolve	[71]	1
Architecture Tradeoff Analysis Method (ATAM)	identify	[72]	2 1
Other methods(the exact name was not mentioned)	identify	[32], [56]	2

Compared to other practices, there are more methods for conflict resolution; 36% of methods noted in papers are used for resolving conflicts. Around half of the methods is for identifying inconsistency. Figure 9 clearly establishes a comparison of the purpose of the methods.

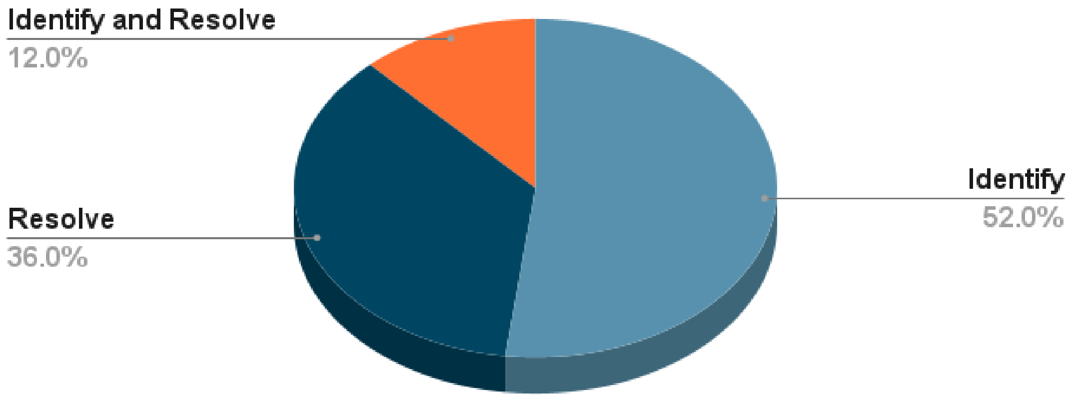

Figure 9. Comparison of existing methods by their purposes

5.2.6 Methodologies

The term "methodologies" is mentioned in the literature for identifying and/or resolving conflicts in practice. Compared with other practices, the term "methodologies" is not preferable in the industry.

As shown in Table 15, 5 methodologies are noted in 7 articles. GORE methodologies are mentioned as an identification practice in 2 papers and as an identification and resolution in one of the papers.

Table 15. Methodologies for conflict management

Methodology	Solution	Reference	Number
GORE methodologies	identify identify and resolve	[42], [57] [43]	2 1
View integration methodology	resolve	[35]	1
Parameter analysis methodology	identify	[39]	1
Soft system methodology	identify	[46]	1
Contemporary design methodologies	identify and resolve	[52]	1

Figure 10 shows a percentage comparison of methodologies by their objectives. Half of the methodologies are for identifying conflicts.

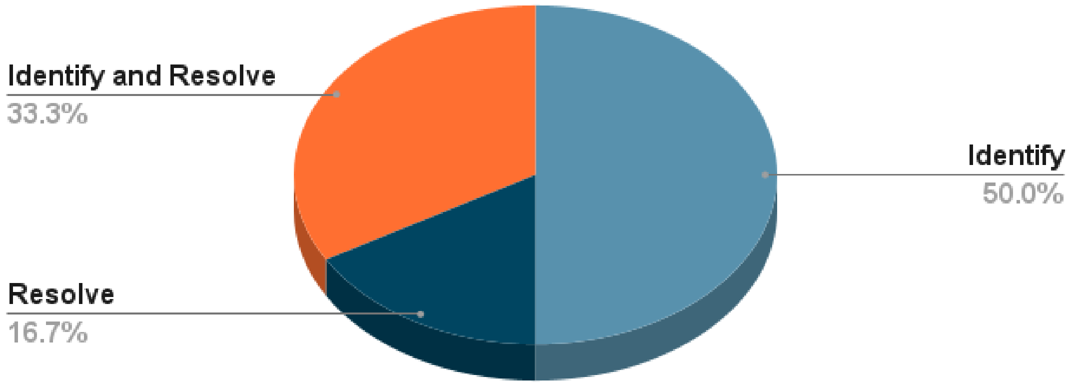

Figure 10. Comparison of existing methodologies by their purposes

5.2.7 Models or Modelling languages

Model or modelling languages is another terminology used in practice and the industry for identifying and/or resolving inconsistency.

Table 16 shows detailed information about *"models or modelling languages"* for conflict identification and/or resolution, which are noted in selected papers. 15 *models or modelling languages* are mentioned in 12 different papers. These practices are not preferable in the RE process of STSs, as only 18% of the selected papers had information about them. Two articles mentioned the *model* of argumentation practice, and one of them [38] suggested it as a conflict identification and resolution practice. Still, the other one is [32] mentioned it as a resolution practice.

Table 16. Models or Modelling languages for conflict management

Model or Modelling language	**Solution**	**Reference**	**Number**
Model of argumentation	resolve identify and resolve	[32] [38]	1 1
Global Model	identify	[21]	1
Essential user interface (EUI) prototype model	identify	[21]	1
Reflexion Modelling	identify	[29]	1
Computational market model	resolve	[33]	1
Conceptual design model	identify and resolve	[39]	1
Multi-layer graph model	resolve	[39]	1
Decision support system (DSS) design models	resolve	[39]	1
Modeling crosscutting concerns as scenarios	identify	[49]	1 1
Requirements dialog meta-model	identify	[55]	1
UML metamodel	identify	[55]	1
Quantitative probabilistic models	identify and resolve	[57]	1
Techne modeling language	resolve	[58]	1
Business process modeling language (BPMN modeling language)	identify	[61]	1
Tropos goal models	identify	[73]	1

Figure 11 shows the comparison of *models or modelling languages* by their purpose. The model of argumentation was mentioned in the papers as an identification and resolution model and resolution model. That's why while creating Figure 11, this *model* was counted separately as an identification and resolution *model*, and also as a resolution *model*. As in other practices, most *models or modelling languages* are helpful for the identification of conflicts.

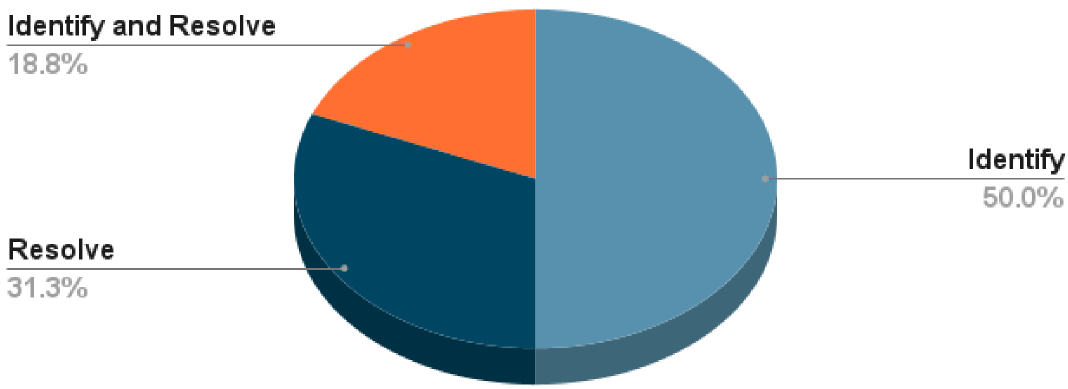

Figure 11. Comparison of existing models or modelling languages by their purposes

5.2.8 Others Practices

Some practices were used as terminologies that were hard to combine under one group. That's why they are grouped under the *"Other Practices"* subsection. It includes algorithms, patterns, policies, and so on.

Table 17 gives a detailed overview of *"Other Practices."* They have been sorted from most mentioned to less mentioned. Heuristics were noted as an identification practice [2], [39], [47] and also resolution practice [31] at the same time.

Table 17. Others Practices for conflict management

Name	Solution	Reference	Number
Aspect-Oriented Requirements Engineering (AORE)	identify	[14], [48], [49] [74]	4
Heuristics	identify resolve	[2], [39], [47] [31]	3 1
Essential use cases (EUCs) pattern	resolve	[17], [21]	2
Automated assistance	identify	[12]	1
CMU-SEI	identify	[12]	1
English grammar based pattern	identify	[16]	1
ASSERT language	identify	[16]	1
QuadREAD project (Quality-Driven Requirements Engineering and Architectural Design)	identify	[17]	1
Model-checking-based analysis	identify	[19]	1
XLinkit service	identify	[25]	1
Domain Specific Language (DSL)	identify	[29]	1
Using naming conventions	identify	[29]	1
Middleware infrastructure	identify and resolve	[31]	1
Design patterns	resolve	[41]	1
Genetic algorithm	identify	[43]	1
Obstacle analysis	identify	[43]	1
Unsupervised machine learning algorithm - k-means	identify	[44]	1
Feature Driven Development	identify and resolve	[48]	1
Theory W	identify	[52]	1
Intelligent agent	identify	[52]	1
Taxonomies	identify	[54]	1
Secure Tropos	identify	[58]	1
Xipho	resolve	[58]	1
Timing strategies	identify	[59]	1
Using catalogs	identify	[62]	1
QoS policies	identify	[63]	1
A delta integration mechanism	identify	[64]	1
Graph theory	identify	[65]	1
Bayesian Belief Networks (BBN)	identify	[65]	1
Propositional logic	resolve	[66]	1
SCR (Software Cost Reduction) tabular notation	identify	[73]	1

Figure 12 compares the *"Other Practices"* by their objectives. Heuristics were counted as identification and resolution practices separately while creating Figure 12, as it was mentioned in 3 of the papers [2], [39], [47] as an identification practice and one of the papers [31] as a resolution practice. The main part of these practices (78.1%) is for conflict detection. Few of them are useful for resolving or identifying and resolving conflicts.

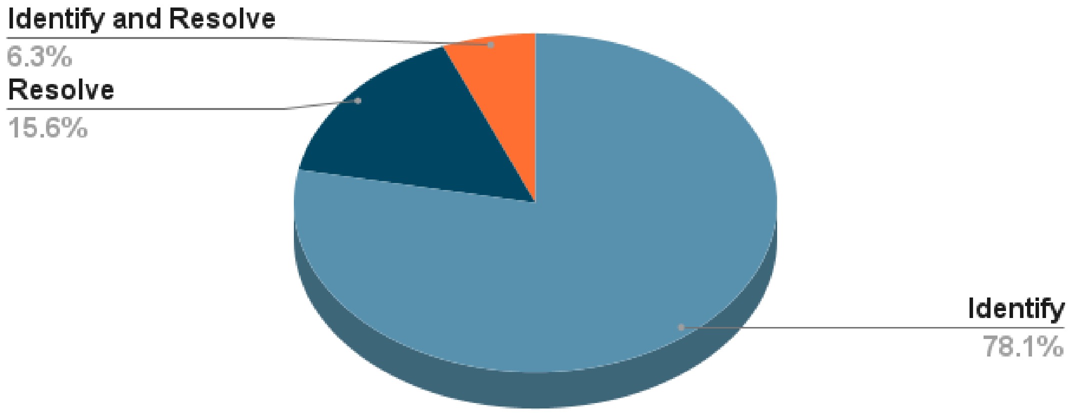

Figure 12. Comparison of existing *"Other Practices"* by their purposes

5.3 RQ3: Stakeholders' involvement in decision-making process

Table 18 provides information on the involvement of stakeholders in the decision-making process during the conflict identification and/or resolution in RE. 23 of the selected articles had information about stakeholders' involvement. One of the papers [66] mentioned that stakeholders are involved in some of the existing conflict management practices but not the others.

Table 18. Stakeholders involvement in decision-making process

Involve stakeholders?	**Reference**	**Total number**
Yes	[2], [12], [22], [23], [32], [33], [34], [35], [36], [37], [41], [47], [48], [52], [58], [61], [66] [67], [68]	19
No	[20], [24], [53], [66], [70]	5

5.4 RQ4: Taxonomy or classification of existing techniques

The existing practices are classified according to their types (used the same classification defined in the section 5.2). Details can be found in Table 19. One of the most preferable is Tools. 38 different tools are mentioned in the selected papers. All types are sorted from most noted to less noted.

Table 19. Procedures

Procedure	**Total number**
Tools	38
Others	31
Approaches	30
Methods	23
Techniques	18
Models or Modelling languages	15
Frameworks	13
Methodologies	5

In the study, the mentioned practices for identifying and/or resolving conflicts can be classified as follows:

1. Formal practices
2. Natural language-based practices
3. Ontology-based practices
4. Goal-oriented based practices
5. Machine learning based practices

Figure 13 gives an overview of general taxonomy and examples of practices related to each classification.

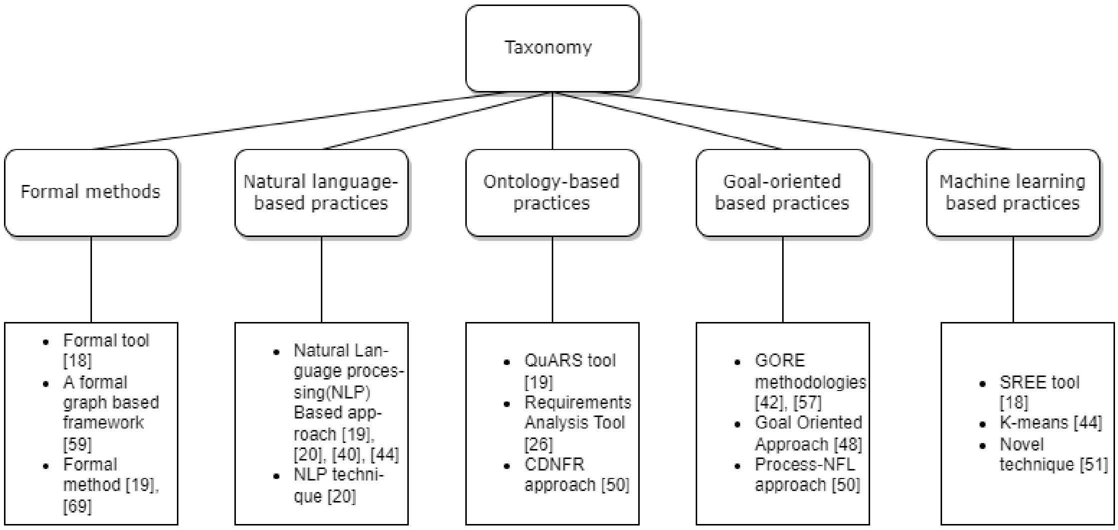

Figure 13. Taxonomy of existing practices

5.5 RQ5: Processes in software engineering associated with existing solutions

Table 20 provides an overview SDLC phases in which the existing practices are used. Most of the existing practices are operated during the requirement analysis phase. Some practices can be used to identify and resolve conflicts in the design and implementation phases.

Table 20. Existing practices are associated with which the SDLC phase

Software engineering process	Reference	Number
Requirement analysis	[3], [12], [13], [15], [16], [17], [18], [19], [20], [21], [22], [25], [27], [40], [41], [42], [44], [47], [49], [51], [53], [54], [58], [59], [66], [67], [69], [70], [71], [72], [73], [74]	32
Design	[15], [32], [33], [35], [38], [39], [46], [48], [52]	9
Software Implementation	[12], [13], [29], [50], [68]	5

5.6 RQ6: The gaps or setbacks of existing techniques

Table 21 shows a detailed outline of the main gaps in existing practices for identifying and/or resolving conflicts in the RE process of STSs. The results are sorted from most mentioned to less-mentioned.

Table 21. Gaps or setbacks of existing practices

New	Reference	Number
Require expertise	[15], [19], [21], [22], [24], [25], [40], [45], [58], [66]	10
Not completely automated and requires human intervention	[14], [19], [20], [21], [22], [40], [47], [57], [63]	9
Lack on NFR	[12], [13], [23], [29], [35], [61], [62], [69]	8
Suitable for small projects	[16], [34], [36], [37], [38]	5
Manual process - time-consuming and error-prone	[3], [50], [74]	3
Not provide expected satisfaction for stakeholders	[2], [11]	2
Solve only actual conflicts but unable to resolve potential conflicts	[2], [70]	2
Results are not always accurate, desirable	[17], [66]	2
Lack of stakeholders' involvement	[20], [53]	2
Not consider resolving conflicts, only identify	[29], [61]	2
Not conflicting pairwise but conflicting in three	[56], [65]	2
Less practical and not productive for multiple stakeholders	[53]	1

5.7 RQ7: New practices offered in the papers for identifying and resolving conflicts

New practices are offered in most of the papers to identify and/or resolve conflicts in RE. The new practices are divided into 3 groups according to their purposes such as 1. identification practices, 2. resolution practices, and 3. identification and resolution practices.

Table 22 provides an overview of offered new practices for detecting inconsistencies. Several frameworks [11],[13],[17],[23],[40] are suggested in the selected papers, the ones without the name specified clearly [17],[23],[40] are grouped under *Framework*. Also, new approaches [22],[49] or methods [47],[51] without names are grouped as *novel approaches* and *novel methods*. Overall, 24 new practices are offered in papers for conflict identification.

Table 22. New practices for conflict identification

New	**Reference**	**Number**
Framework	[17], [23], [40]	3
SureCM framework	[11], [13]	2
Novel approach	[22], [49]	2
Novel Method	[47], [51]	2
Using quality attributes and requirements traceability	[14]	1
Novel technique	[16]	1
Using argumentation theory as a formal tool	[18]	1
MaramaAIC	[21]	1
Automated prototype tool	[25]	1
EDTL-patterns	[28]	1
SpaceSolver	[33]	1
Genetic algorithm	[42]	1
Goal-Conflict Likelihood Assessment Based on Model Counting	[45]	1
Hierarchical clustering technique	[46]	1
Semantic Web Rule Language (SWRL) (SWRL)	[50]	1
The argumentation methodology	[52]	1
RADAR	[57]	1
Order of Conflict concept	[65]	1
EA-Analyzer	[74]	1

Also, 7 new practices are offered for conflict resolution in 7 articles. Table 23 provides brief details of them.

Table 23. New practices for conflict resolution

New	**Reference**	**Number**
ConsView approach	[31]	1
Framework	[39]	1
DeepCoref method	[44]	1
Decision-making and tradeoff analysis using goal models	[60]	1
Optima model	[67]	1
Changing communication media	[68]	1
Improved Access Control Model	[70]	1

In Table 24, we found that 26 new practices are mentioned in the selected papers for conflict identification and resolution in the literature. We use the term *frameworks* to represent the practices that are not clearly specified in the literature.

Table 24. New practices for conflict management

New	Reference	Number
Framework	[3], [19], [32], [55], [61], [63], [64]	7
Stakeholder-Centric Clustering Methods framework	[2]	1
Jigsaw puzzle metaphor	[24]	1
Novel method	[27]	1
Language Extended Lexicon (LEL) model	[30]	1
New approach	[35]	1
Tool based on a computational argumentation method	[38]	1
Method for eliciting trust-related software features	[41]	1
Post-processing framework (PPFc)	[43]	1
Joint framework (JFc)	[43]	1
Semi-formal Ontology Driven Domain-Specific Requirements Specification Language	[53]	1
Crossover service goal convergence method	[56]	1
Arg-ACH approach	[58]	1
Comprehensive method	[59]	1
Propose a catalog of conflicts	[62]	1
Semi-formal mechanism	[66]	1
Integrated requirement analysis method	[69]	1
Ternary logic based state machine	[71]	1
Conflict-centric approach	[72]	1
Automated analysis techniques	[73]	1

Figure 14 shows comparison of new offered practices by their purposes. Most of the new practices are for conflict identification and resolution. 42.1% of them are useful for detecting inconsistency. Little research is done for the resolution of conflicts.

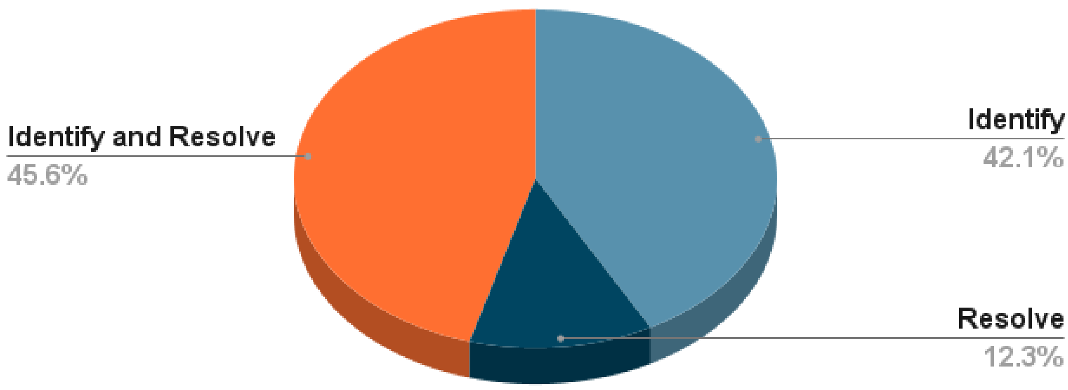

Figure 14. Comparison of new practices by their purposes

6 Discussion

This section discusses the implications of the study based on the results.

6.1 RQ1: Overview for existing solutions

As mentioned in Section 3, 1720 papers were received without filter after running queries in the selected data sources. After applying inclusion, exclusion, and quality criteria, 60 papers were obtained, and 6 articles were manually added. In total, 66 papers have been chosen for the research, which have specified details about existing practices for conflict resolution and/or identification. More information on the elimination process can be found in Section 3.

Figure 4 gives information about how many papers are published each year between 2000 and 2022. Interestingly, most papers were published in 2001. The majority of papers were published the previous decade - after 2010. Eighteen (18) papers - around 27.2% of articles were published between 2000 and 2009. Thirty-eight (38) articles - approximately 57.6% - were printed between 2010 and 2019. It is the beginning of the current decade - only a few articles (10 articles) have been published in 2020, 2021, and 2022. As shown in Figure 4, there wasn't much interest in conflict identification and/or resolution in the RE process of STSs in the 2000's years. After the 2010s, more research and further study were done in this direction.

6.2 RQ2: Current practices for identifying and resolving conflicts or inconsistencies

6.2.1 Tools

The use of the term *Tools* is one of the preferable practices in the industry for conflict management. Many tools are mentioned in the selected papers.

Table 6 shows the summary of tools for conflict identification mentioned in the selected papers. QARCC is one of the most used conflict identification tools in the industry. It was mentioned in a total of 4 papers. QARCC tool is based on a win-win approach [11], [12], [13].

Other preferable tools are the SpeAR (Specification and Analysis of Requirements), QuARS, and EA-Analyser tools, which are noted in 3 articles. SpeAR tool using LTL(linear temporal logic) to find inconsistencies [17]. QuARS can determine linguistics conflicts in requirements [19].

Only 3 tools are mentioned in the papers for resolving conflicts, and details are reflected in Table 7. These tools are Synoptic [2], HERMES [32] and Design advice tool [33]. Synoptic is a computer-based negotiation system, but stakeholders' fulfillment is not ensured [2]. Design advice tools help to resolve inconsistencies in collaborative design, and it also helps in negotiation [33].

Table 8 gives an overview of existing tools for conflict identification and resolution. Some papers mentioned QARCC [11], [12], [13], [14] and S-COST [23] as inconsistency identification tools only, but in some papers they QARCC [34], [35] and S-COST [12], [34] are noted as identification and resolution tools. QARCC and S-COST involve negotiating conflicts, identifying inconsistencies in functional and quality requirements, and suggesting some resolution options [34].

Tradeoff analysis tool and Distributed Collaboration Priorities Tool (DCPT) are other preferable tools in the RE process of STSs to identify and resolve conflicts. Tradeoff analysis tools help stakeholders define win conditions, specify conflicting win conditions, and offer different options to handle these inconsistencies [36], [37].

In [32], Hermes was reported as a resolution tool, but in [38], it was mentioned as an identification and resolution tool. There are several tools that were mentioned in only one article for identifying and resolving inconsistencies such as the AS-SERT™ tool[15], HERMES[38], gIBIS (graphical IBIS) [38], Teleologic DOORS [38], QuestMap [38], and Analytical tool [39].

As shown in Figure 5, the main percentage of tools noted in selected papers for conflict identification and/or resolution is for the identification of inconsistencies. Not much work has been done for conflict resolution. Only 3 tools are noted for conflict resolution. Compared with resolution tools, more tools were developed for identifying and resolving conflicts - 24.4% of all tools are for identifying and resolving conflicts.

6.2.2 Approaches

A considerable number of approaches were documented in the articles for identifying and resolving conflicts in Requirement Engineering. They are divided into 3 groups according to their purposes in practice such as identification, resolution, and identification and resolution approaches.

As shown in Table 9, the ontology-based approach is one of the preferable approaches in RE for identifying conflicts. This approach was mentioned in 4 papers [19], [26], [40], [41]. However, some of these ontology-based approaches are not fully automated and require expertise involvement [40]. Ontology-based approaches can be used to identify semantic conflicts[19]. Pattern-based approaches are another most used approach for conflict identification. Natural language processing-based approaches help to find ambiguous terms [44].

Four (4) approaches are mentioned in the selected articles for identifying conflicts in RE without specifying their name, and these approaches are grouped under *"Other approaches,"* one of these approaches that finds inconsistency between the two requirements is offered by Harel et al [43].

Some other approaches for conflict identification such as the Rule-based approach

[18], Tableaux-based approach [42], Goal-conflict analysis approach [45], and Win Win approach [46] are noted in only one article.

As in tools, there are only 3 approaches for conflict resolution such as Win Win approach [2], Component-based RE approach [48], Rule-based approach [52]. Table 10 gives a summary of these approaches. Compared to the number of conflict identification approaches, it is too little. It proves that less work has been done in the resolution of conflicts.

Table 11 shows approaches for the identification and resolution of conflicts. The win-win approach was mentioned as an identification practice in one of the papers [46], and the other one [2] noted it as a resolution practice. However, in 6 of the selected articles [12], [34], [35], [36], [37], [53] it was reported as an identification and resolution approache. Although Win-win is the most noted approach for conflict management (identify and resolve) in the selected papers, it is appropriate only for small projects. When more stakeholders are involved, negotiation will be harder [36], [37] and scalability will become a major issue.

Besides being reported as an identification approach [19], [44], Natural language processing-based approaches were also noted as an identification and resolution practice [20], [40]. Ontology-based approaches were the most mentioned conflict identification approach. However, it was also reported as a conflict identification and resolution approach [28], [54].

Figure 6 shows the comparison of approaches mentioned in the selected papers by their purposes. As in tools, most of the approaches are for detecting conflicts in RE. Approximately 1 out of 3 approaches were developed for identifying and resolving conflicts. Figure 6 demonstrates a significant gap in resolving conflicts during the RE process of STSs.

6.2.3 Frameworks

The use of framework is also another operated practice in the industry. Only a few frameworks are documented in the literature compared to tools and approaches. That's why all frameworks are grouped in one and only Table 12.

Non-Functional Requirements (NFR) framework [24], [46] and Non-Functional Decomposition (NFD) framework [23], [32] are developed to manage conflicts in non-functional requirements. NFR framework is for identifying conflicts, and the NFD framework was reported as a resolution framework in [23] and in [32], it was mentioned as a framework for identifying. KAOS goal modeling framework is another most reported framework that uses linear temporal logic (LTL) for conflict identification, but LTL demands engineers' expertise to pinpoint requirements [19] accurately.

Six (6) frameworks [27], [35], [53], [61], [62], [63] for identification, one framework [32] for resolution, and 3 frameworks [52], [55], [64] for identification and resolution of conflicts are documented with undefined name. For that, they are grouped as *"Other* *frameworks."*

As Figure 7 shows, most of the frameworks are for conflict identification. One-fourth of the frameworks were developed for handling resolution, which is better compared to tools and approaches. A few frameworks can identify and resolve inconsistencies at the same time.

6.2.4 Techniques

Several techniques are used in the industry for conflict management. Table 13 shows the summary of these techniques.

Trace analyzer of the requirements traceability technique and a technique that adopts a hierarchical constraint logic programming approach is the most mentioned technique in selected papers for conflict identification [11], [13]. Some articles documented techniques without exact names defined and they are considered as "Other techniques" [3], [19], [51], [56]. Identifying and resolving conflicts state-of-the-art techniques are mentioned in [3]. An automatic technique was created to provide consistency in [19].

Some of the techniques are only reported in one paper such as Model-based validation and inconsistency detection technique [22], Unified Modeling Language (UML)-Based technique [40], Analysis of Competing Hypotheses technique [43], the Pareto front technique [46], Multiple Criteria Decision Making (MCDM) technique [49], VWPL (Viewpoints Language) technique [53], Automated traceability technique [61], Theorem-proving and model-checking technique [65].

As shown in Figure 8, 73.7% of techniques are for the detection of conflicts in RE. A few works have been done for inconsistency resolution or inconsistency identification and resolution.

6.2.5 Methods

The methods are another preferable practice in the industry for identifying and/or resolving conflicts in RE. The methods mentioned in the papers are reflected in Table 14.

The negotiation method is the most documented in the selected articles. One of the papers [61] is mentioned as a conflict identification practice but in 6 papers [2], [22], [33], [58], [67], [68] is reported as a conflict resolution practice. Stakeholder and requirement engineers discuss requirements to find possible inconsistencies [61]. In the negotiation method, engineers communicate with stakeholders to resolve conflicts. However, it is impossible to satisfy all stakeholders, as noted in [2].

Further, informal and semi-formal methods have been developed to identify conflicts [43], [51]. Informal methods also can be used to resolve inconsistencies [66]. Also, formal methods have been introduced for identifying conflicts [19], [69]. One of them helps to find inconsistencies in security and safety requirements [69]. Requirement

Prioritization is another preferable method in the industry for handling conflicts. Another method was proposed to find inconsistencies in non-functional requirements in [32].

Methods such as Multiple Criteria Decision-Making method [22], REMAP (REpresentation and MAintenance of Process knowledge) [38], Risk analysis method [69], Description Logic based method [70], Goal-based method [71], Architecture Tradeoff Analysis Method (ATAM) [72] are reported only in one paper.

As shown in Figure 9, 36% of the methods mentioned in selected papers have been introduced for resolving conflicts in RE. This percentage is a lot better compared to tools, approaches, frameworks, and techniques. Approximately half of the methods are for detecting inconsistencies. Only a few methods have been mentioned for identifying and resolving conflicts in the literature.

6.2.6 Methodologies

Five (5) methodologies are noted from the selected paper for conflict management. Table 15 shows the summary of these methodologies.

Notably, Goal-Oriented Requirements Engineering (GORE) process was reported as a conflict identification methodology in [42], [57]. However, in [43], it was mentioned as a conflict identification and resolution methodology.

Also, Parameter analysis [39] and Soft system [46] methodologies were developed for conflict identification, while Contemporary design methodologies [52] were mentioned as an identification and resolution methodology. View integration methodology [35] was implemented for resolving conflicts.

To sum it up, Figure 10 shows that half of the methodologies mentioned in the selected papers are identification methodologies. One-third of the mentioned methodologies were developed for conflict identification and resolution. As in most of the practices, less work has been done for conflict resolution in the RE process of STSs.

6.2.7 Models or Modelling languages

Models or Modelling languages that are reported in the selected articles are mirrored in Table 16. Thus, 15 Models or Modelling languages are mentioned in 12 articles.

However, only Model of argumentation are mentioned in more than one article [32], [38]. In [32], the Model of argumentation was suggested as a resolution model, while in [38] it was considered for as identification and resolution of conflicts in practice. Remarkably, several software and systems are implemented based on the model of argumentation to manage conflicts [38].

To identify and resolve conflicts during the design phase Conceptual design model was implemented [39]. Techne modeling language was implemented to resolve conflict, which gives options to select requirements and guarantees that the selected requirement is not conflicting [58]. Additionally, an extension of the Business process modeling language (BPMN) was developed to identify inconsistencies [61].

Computational market model [33], Multi-layer graph model [39], Decision support system (DSS) design models [39] were mentioned in the selected papers which are capable to resolve inconsistencies.

Half of the Models or Modelling languages are mentioned as an identification practice. Approximately 31.3% of models or modelling languages are developed for resolving conflicts in RE. Figure 11 shows that compared with tools, approaches, frameworks, techniques, and methodologies, more work is done for conflict resolution in models or modelling languages.

6.2.8 Others Practices

Some other practices which are not possible to group with other practices are combined all in Other Practices. Table 17 shows an overview of these 31 *other practices*.

For example, Aspect-Oriented Requirements Engineering (AORE) practice is one of the most mentioned in [14], [48], [49], [74]. This practice is useful for identifying conflicts in RE. Heuristics is another preferable practice that is mentioned as an identification practice in [2], [39], [47] and as a resolution practice in [31].

Also. Patterns were noted for conflict resolution such as Essential use cases (EUCs) patterns [17], [21] and Design patterns [41]. For inconsistency identification, English grammar-based patterns [16] were reported. Algorithms can also be used for conflict identification in industry. Genetic algorithm [43] and Unsupervised machine learning algorithm - k-means [44] have been reported as other strategies for identifying and resolving conflicts in the literature.

Moreover, Middleware infrastructure [31] and Feature Driven Development [72] are used in practice to both identify and resolve conflicts. Languages such as Domain Specific Language (DSL) [16] and ASSERT language [44] are appropriate practices to detect inconsistencies.

In addition, Automated assistance [12], QuadREAD project (Quality-Driven Requirements Engineering and Architectural Design) [17], Model-checking-based analysis [19], Taxonomies [54], Secure Tropos [58], Timing strategies [59], Delta integration mechanism [64], Graph theory [65], Bayesian Belief Networks (BBN) [65], SCR (Software Cost Reduction) tabular notation [73], and others are developed to identify conflicts or inconsistencies in RE.

As shown in Figure 12, most of the practices - around 78.1% that are grouped under *Other practices*- are for detecting conflicts. It also shows that less research has been done on resolving conflict and identifying and resolving conflicts in the RE process of STSs.

In general, a lot of practices are mentioned in the selected papers for this study. To analyze and visualize the results, the practices are divided into 8 groups: tools, approaches, frameworks, techniques, methods, methodologies, models or modeling languages, and other practices. Figures 5, 6, 7, 8, 9, 10, 11 and 12 give information that most practices are for identifying conflicts. There is a significant gap between identification practices and resolution or identification and resolution practices. Identification and resolution practices can be used to detect and handle conflicts at the same time, but their count is not many. This means the resolution of conflicts in the RE process of STSs is still in its infancy stage and requires a lot of attention by the RE researchers and practitioners.

6.3 RQ3: Stakeholders' involvement decision-making process

Another main question in this research is to determine if stakeholders' participation in decision-making while analyzing requirements to identify and/or resolve conflicts is guaranteed. Details about it have been mentioned in 23 of the selected articles (Table 18). This means that only 34.8% of the selected papers mentioned stakeholders' involvement, proving that not much research has been done in this regard. Therefore, there is a wide gap in the research regarding engendering stakeholders' satisfaction, trust, and confidence in the development process, which the RE community, especially researchers and practitioners, need to fill.

It is noted in only one of the selected papers, some of the practices involve stakeholders, but some don't [66]. In informal methods, negotiation with stakeholders takes place to find and resolve conflicts, but stakeholders don't directly affect formal methods [66]. In reality, the formal method is unsuitable for industrial practices, especially when large and various stakeholders are involved, and their participation is essential in determining the success of the development project.

18 of 23 articles reported that stakeholders participate in the decision-making process, 4 of 23 papers highlighted that mentioned practices for conflict identification and/or resolution do not involve stakeholders in the decision-making process. 1 of the 23 papers mentioned that some practices involve stakeholders, and some don't include them directly.

One of the most mentioned methods for conflict resolution is the Negotiation method [2], [22], [33], [58], [67], [68] which is reflected in Table 14. Negotiation methods involve discussing with stakeholders during the conflict resolution process to remove inconsistencies. Win-Win's approach also involves stakeholders negotiating options to get an agreement [36].

Overall, we observed that the lack of participation of stakeholders could result in inconsistencies and ambiguity in requirements [20].

6.4 RQ4: Taxonomy or classification of existing techniques

This research question sought to categorize existing conflict identification and resolution strategies by identifying commonalities among them.

In subsection 5.2, existing solutions were divided into different clusters according to their functionality, such as *tools, approaches, frameworks, techniques, methods, methodologies, models or modelling languages, and other practices*. The existing practices can be classified according to these clusters. Table 19 shows an overview of this taxonomy in detail. The majority of the existing practices fall into the *tools* cluster. There were a lot of practices in *other practices* (31 practices), *approaches* (30 practices), *methods* (23 practices), and *techniques* (28 practices) groups.

Figure 13 shows the general classification of the existing practices for conflict identification and/or resolution and examples for each classification. These classifications are as follows: *Formal methods* (Formal method [19], [69], Formal tool [18], A formal graph-based framework [59]), *Natural language-based practices* (Natural Language processing (NLP) Based approach [19], [20], [40], [44], NLP technique [20]), *Ontology-based practices* (QuARS tool [19], Requirements Analysis Tool [26], CDNFR approach [50]), *Goal-oriented based practices* (GORE methodologies [42], [57], Goal Oriented Approach [48], Process-NFL approach [50]), *Machine learning based practices* (SREE tool [18], K-means [44], Novel technique [51]).

6.5 RQ5: Processes in software engineering associated with existing solutions

Software Development Life Cycle (SDLC) mainly has 5 phases: Requirement Analysis, Software Design, Software Implementation, Software Testing, and Software Maintenance [75]. The primary purpose of RQ5 was to find out which phase of SDLC the existing practices for conflict identification and/or resolution can be used and which phase should be improved.

We found the answer to this question in 43 of 66 papers - approximately 65% of the selected papers. Existing practices are suitable only for the Requirement Analysis, Software Design, and Software Implementation phase. None of the papers mentioned a practice to identify and/or resolve conflicts in Software Testing or Software Maintenance phase. Nonetheless, it is possible that conflicting views and expectations during software maintenance in a development project may occur. This can be an interesting research plan to consider. However, in [15], identifying and resolving conflicts can be executed during the Requirement Analysis and Design phases. Some practices in [12], [13] are sufficient in the Requirement analysis phases. Some others can be used in the Software Implementation phase.

Table 20 shows that most of the papers mentioned practices are useful in the Requirement Analysis phase. In total, 32 papers have information about the Requirement Analysis phase. There are also practices that is for managing conflicts in Design phase - 9 papers mentioned these practices such as Design advice tool [33], Conceptual design model [39], Contemporary design methodologies [52], and so on. Several articles report practices for Software Implementation phases, such as Reflexion Modelling [29], Process-NFL [50] and so on.

6.6 RQ6: The shortcomings of existing techniques

There are a lot of practices which has been developed or implemented for conflict management. However, existing practices have some gaps and disadvantages. Table 21 shows an overview of these shortcomings.

10 papers [15], [19], [21], [22], [24], [25], [40], [45], [58], [66] mentioned that some of the existing practices require expertise. Some methods depend on the analysts' prior knowledge [58], experience [22] and require computing backgrounds [24]. Some ontology-based approaches are not fully automated and require experts in the domain, so results cannot be trustable [40]. Before using the EA-Analyzer, training should be completed [21].

Another problem with existing approaches is that they are not fully automated, and require human intervention [14], [19], [20], [21], [22], [40], [47], [57], [63]. However, it can be error-prone, and results are not entirely reliable. This gap was reported in 9 papers. Also, existing tools require human attention to check the results of consistency checks while trying to identify inconsistencies [21]. Some processes cannot be entirely automated as resolving conflicts take more time [20], and humans' dynamic or changing nature seems complex for perfect automation.

In addition, 8 articles reported a lack in the non-functional requirements [12], [13], [23], [29], [35], [61], [62], [69]. Most of the existing practices are for functional requirements. For example, there is a gap in conflict management in security and privacy requirements [61]. As observed in [23], there is no current attempt to identify and resolve conflicts in NFR. Moreover, automated assistance practice does not take into account stakeholders' concerns [12].

Interestingly, we observed that some of the existing practices are only suitable for small projects [16], [34], [36], [37], [38]. Win-Win is one of them, when more stakeholders are involved, negotiation becomes difficult [34], [36], [37] and complex. Also, the negotiation method is one of the preferable practices for conflict resolution. Overall, stakeholders' satisfaction is not fulfilled [2].

It was reported that manual processes can be time-consuming and error-prone [3], [50], [74]. There are a lot of existing solutions for identification, and some of the papers mentioned that the existing solution only identifies conflicts and does not take resolving into account [29], [61].

There are other setbacks in existing practices such as a lack of stakeholders' involvement [20], [53], solving only actual conflicts but being unable to resolve potential conflicts [2], [70], not conflicting pairwise but conflicting in three [56], [65], results not always accurate [17], [66], less practical and not productive for multiple stakeholders [53].

Overall, some gaps in existing solutions are not reflected in Table 21. For instance, logical inconsistency is not identified [19], insufficient [40], very expensive, and even impractical [43], the complexity of method [57], stakeholders' ideas being incomplete [58], the need for some formal specification of requirements [74] and etc. are some of the shortcomings of existing techniques. By implication, more research efforts are required to manage conflicts during the RE process of STSs to build high-quality software-intensive systems further.

6.7 RQ7: New practices offered in the papers for identifying and resolving

Some new practices are offered in the selected papers. They have been divided into 3 groups according to their objective: new practices for identifying, resolving, and identifying and resolving conflicts.

As shown in Table 22, in 24 papers, new practices for identification are presented. 5 new frameworks were mentioned in 5 articles, and 2 of these papers offered the SureCM framework [11], [13]. The name of the other 3 frameworks weren't specified clearly and they were grouped under Frameworks [17], [23], [40]. One of these frameworks can find conflict among non-functional requirements [23], and the other requires human expert review after finding inconsistencies [40].

A novel method was implemented to detect inconsistencies in the design phase of SDLC [47]. Requirements Description Language approach has been noted as an identification practice based on natural language [49]. Different new tools have also been developed for detection such as the use of argumentation theory as a formal tool [18], MaramaAIC [21], the automated prototype tool [25], RADAR [57], and EA-Analyzer tool [74].

As shown in Table 23, 7 new conflict resolution practices are described in the selected papers. They include, ConsView approach [31], Framework [39], DeepCoref method [44], Decision-making and tradeoff analysis using goal models [60], Optima model [67], Changing communication media [68], Improved Access Control Model [70]. Outstandingly, ConsView helps to handle conflicts in multiple applications that are running in one system [31].

Table 24 provides an overview of the newly offered conflict identification and resolution practices mentioned in the 66 selected papers. Frameworks without a specific name are grouped under *Frameworks* [3], [19], [32], [55], [61], [63], [64]. For managing conflicts between security and privacy requirements, a new framework has been developed in [60]. Other frameworks are also reported in the articles as examples: Stakeholder-Centric Clustering Methods framework [2], The post-processing framework (PPFc) [43], and Joint framework (JFc) [43]. Also, some new methods have been implemented for identifying and resolving inconsistencies, such as, a Novel method (CARL tool was developed based on this method) [27], tool based on a computational argumentation method [38], Crossover service goal convergence method [56], Comprehensive method [59], and the Integrated requirement analysis method [69].

Figure 14 shows the comparison of new practices by their objectives. Most of the new practices - 45.6% can identify and resolve conflicts. 42.1% of new practices are for identifying inconsistencies. Only a little work has been done to resolve inconsistencies - 12.3% of new practices can be used to resolve conflicts.

A comparison of existing practices has been made in Section 6.2. We found a big difference between existing and new practices in their purposes. Most of the existing practice is only for identifying conflicts. However, in new practices, more work has been done to develop practices for identifying and resolving conflicts. Similarities between existing and new practice are that few of these practices is for resolving conflicts. It proves that some research should be done and developed to resolve conflicts. However, further research should take into account that most of the new practices can identify and also resolve conflicts. That's why current practices have no significant gap in resolving inconsistencies.

6.8 Quality Scores of Papers

Table 26 gives detailed information about estimated quality scores for each selected article. Only 2 of the selected articles [58], [66] got a full score - 5 points out of 5. 52 of 66 selected papers are estimated at 3 or more points. This proves that the quality of the selected articles is quite satisfactory, which means at least in the 52 papers, we encountered an answer to at least 3 of 5 QAQs. To sum up, we have found the answer to QAQ1 in all the articles and QAQ5 in most of the papers.

7 Research Findings

Some of the findings from the research are listed below:

(i) No similar SLR mainly focused on conflict identification and/or resolution in practices.

(ii) Conflict management has attracted more researchers in recent years, so most of the work has been done in the last decade.

(iii) There are a few practices for the resolution of conflicts compared with identification. Therefore, research attention should be focused on how stakeholders' conflicting views, expectations, goals, and desires are resolved, especially in real-life software development projects.

(iv) Most existing practices do not involve stakeholders in the decision-making process.

(v) Most existing practices have been developed to manage conflicts in the requirement analysis phase. This means more research is required to address conflicts extensively in other stages of the SE process since the entire SE process involves human participation and collaboration.

(vi) Tools for identifying and resolving conflicts in RE of STSs are not fully automated and require expertise and manual human intervention. Therefore, there is a need for engineering suitable, reliable, and usable tools for handling inconsistencies during the RE process.

(vii) There is a lack of conflict identification and resolution in NFR. Even though few research attempts to address conflicts in NFR [23], [62], more research is expected in this direction to build further acceptable software systems that will mutually satisfy all stakeholders involved.

(viii) There are many newly developed or offered practices in the selected articles for identifying and resolving conflicts in requirements. For that, the SE and RE communities can leverage these new practices to explore better ways of identifying and resolving conflicts from a more pragmatic and realistic perspective.

Therefore, it is worth mentioning that there are limitations in the existing practices, such as fewer practices for conflict resolution and gaps in existing practices. Considering the recent increase in researchers' interest in finding suitable ways to harmonize stakeholders' viewpoints, goals, expectations, and desires in a development project, especially in the agile development methodology, more research is expected in this direction to provide a general approach to assist with a variety of design issues handling conflicting circumstances while creating and putting STSs into practice. To manage conflicts in RE, which is typically expected in many different ways, we anticipate an all-encompassing research approach that should have practical implications for various application domains. For instance, it will be beneficial to examine human psychology and adopt pertinent theories to comprehend the causes of each stakeholder's emotional motives [76].

8 Conclusion and Future Work

This SLR's main goal was to get an overview of the existing conflict identification and resolution research and practices in RE of STSs. For that purpose, 7 RQs were formulated, and search queries were formed according to these RQs. As a result, 66 papers have been selected to answer these questions.

In this thesis, we found that there are a lot of practices for conflict identification and/or resolution. Accordingly, they have been grouped in clusters according to their functionality based on the terminology used by the different researchers/authors in the selected papers. Consequently, these terminologies used in the various research include *tools, approaches, frameworks, techniques, methods, methodologies, models or modeling languages, and other practices*. However, we discovered considerable limitations in the research area for conflict resolution as most of the practices have been developed for conflict identification in RE.

Also, most of these practices do not involve stakeholders during the decision-making process. Furthermore, most of these practices are suitable for the requirement analysis phase, while some can be used in the SDLC's design and software implementation phases.

Also, there are significant drawbacks to existing conflict management practices, such as requiring expertise, previous experience, and manual human involvement. Again, there is a gap in existing practices for NFR. Some practices such as Win-Win are suitable for small projects.

RQ7 goal was to construct a summary of the newly developed or offered practices. Enough new practices have been mentioned in the selected papers. In contrast to the existing practices, most new practices are suitable for identifying and resolving conflicts. There are only a few novel conflict resolution practices, the same as in existing conflict management practices.

Future work can involve analyzing or having a case study for newly developed practices to compare these practices and find the most appropriate solutions for conflict identification and/or resolution in RE. This case study also can take into account existing solutions to see which limitations of existing practices have been improved with newly offered solutions and what gaps the newly implemented practices have.

References

[1] Alistair Mavin, Sabine Mavin, Birgit Penzenstadler, and Colin C. Venters. Towards an ontology of requirements engineering approaches. In *2019 IEEE 27th International Requirements Engineering Conference (RE)*, pages 514–515, 2019.

[2] Ishaya Gambo and Kuldar Taveter. Stakeholder-centric clustering methods for conflict resolution in the requirements engineering process. In Raian Ali, Hermann Kaindl, and Leszek A. Maciaszek, editors, *Evaluation of Novel Approaches to Software Engineering*, pages 183–210, Cham, 2022. Springer International Publishing.

[3] Maysoon Aldekhail and Marwah Almasri. Intelligent identification and resolution of software requirement conflicts: Assessment and evaluation. *Computer Systems Science and Engineering*, 40(2):469–489, 2022.

[4] Maysoon Aldekhail, Azzedine Chikh, and Djamal Ziani. Software requirements conflict identification: review and recommendations. *International Journal of advanced computer science and applications*, 7(10), 2016.

[5] Jorge Biolchini, Paula Gomes Mian, Ana Candida Cruz Natali, and Guilherme Horta Travassos. Systematic review in software engineering. *System engineering and computer science department COPPE/UFRJ, Technical Report ES*, 679(05):45, 2005.

[6] Staffs Keele et al. Guidelines for performing systematic literature reviews in software engineering. Technical report, Technical report, ver. 2.3 ebse technical report. ebse, 2007.

[7] Pearl Brereton, Barbara A. Kitchenham, David Budgen, Mark Turner, and Mohamed Khalil. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, 80(4):571–583, 2007. Software Performance.

[8] Ainhoa Aldave, Juan M. Vara, David Granada, and Esperanza Marcos. Leveraging creativity in requirements elicitation within agile software development: A systematic literature review. *Journal of Systems and Software*, 157:110396, 2019.

[9] A César C França, Tatiana B Gouveia, Pedro CF Santos, Celio A Santana, and Fabio QB da Silva. Motivation in software engineering: A systematic review update. In *15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE 2011)*, pages 154–163. IET, 2011.

[10] Barbara Kitchenham. Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004):1–26, 2004.

[11] Dewi Mairiza and Didar Zowghi. An ontological framework to manage the relative conflicts between security and usability requirements. In *2010 Third International Workshop on Managing Requirements Knowledge*, pages 1–6, 2010.

[12] Hoh In, B. Boehm, T. Rodger, and M. Deutsch. Applying winwin to quality requirements: a case study. In *Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001*, pages 555–564, 2001.

[13] Dewi Mairiza, Didar Zowghi, and Vincenzo Gervasi. Conflict characterization and analysis of non functional requirements: An experimental approach. In *2013 IEEE 12th International Conference on Intelligent Software Methodologies, Tools and Techniques (SoMeT)*, pages 83–91, 2013.

[14] A. Egyed and P. Grunbacher. Identifying requirements conflicts and cooperation: how quality attributes and automated traceability can help. *IEEE Software*, 21(6):50–58, 2004.

[15] Abha Moitra, Kit Siu, Andrew Crapo, Harsh Chamarthi, Michael Durling, Meng Li, Han Yu, Panagiotis Manolios, and Michael Meiners. Towards development of complete and conflict-free requirements. In *2018 IEEE 26th International Requirements Engineering Conference (RE)*, pages 286–296, 2018.

[16] Vincent Langenfeld, Daniel Dietsch, Bernd Westphal, Jochen Hoenicke, and Amalinda Post. Scalable analysis of real-time requirements. In *2019 IEEE 27th International Requirements Engineering Conference (RE)*, pages 234–244, 2019.

[17] Issa Atoum. A scalable operational framework for requirements validation using semantic and functional models. ICSIM 2019, page 1–6, 2019.

[18] Yehia Elrakaiby, Alessio Ferrari, Paola Spoletini, Stefania Gnesi, and Bashar Nuseibeh. Using argumentation to explain ambiguity in requirements elicitation interviews. In *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pages 51–60, 2017.

[19] Tuong Huan Nguyen, John C. Grundy, and Mohamed Almorsy. Ontology-based automated support for goal–use case model analysis. *Software Quality Journal*, 24:635–673, 2016.

[20] Assad Alzayed and Ahmed Al-Hunaiyyan. A bird's eye view of natural language processing and requirements engineering. *International Journal of Advanced Computer Science and Applications*, 12(5), 2021.

[21] Massila Kamalrudin, John Hosking, and John Grundy. Maramaaic: tool support for consistency management and validation of requirements. *Automated Software Engineering*, 24:1–45, 2017.

[22] M.J. Escalona, M. Urbieta, G. Rossi, J.A. Garcia-Garcia, and E. Robles Luna. Detecting web requirements conflicts and inconsistencies under a model-based perspective. *Journal of Systems and Software*, 86(12):3024–3038, 2013.

[23] Vishal Sadana and Xiaoqing Frank Liu. Analysis of conflicts among non-functional requirements using integrated analysis of functional and non-functional requirements. In *31st Annual International Computer Software and Applications Conference (COMPSAC 2007)*, volume 1, pages 215–218, 2007.

[24] Maria Pinto-Albuquerque and Awais Rashid. Tackling the requirements jigsaw puzzle. In *2014 IEEE 22nd International Requirements Engineering Conference (RE)*, pages 233–242, 2014.

[25] Massila Kamalrudin. Automated software tool support for checking the inconsistency of requirements. In *2009 IEEE/ACM International Conference on Automated Software Engineering*, pages 693–697, 2009.

[26] Paulo Afonso Parreira and Rosângela Aparecida Dellosso Penteado. Domain ontologies in the context of requirements engineering. In *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, pages 1–8, 2015.

[27] D. Zowghi, V. Gervasi, and A. McRae. Using default reasoning to discover inconsistencies in natural language requirements. In *Proceedings Eighth Asia-Pacific Software Engineering Conference*, pages 133–140, 2001.

[28] Natalia Garanina and Dmitry Koznov. Static checking consistency of temporal requirements for control software. In Ladjel Bellatreche, George Chernishev, Antonio Corral, Samir Ouchani, and Jüri Vain, editors, *Advances in Model and Data Engineering in the Digitalization Era*, pages 189–203, Cham, 2021. Springer International Publishing.

[29] Nour Ali, Sean Baker, Ross O'Crowley, Sebastian Herold, and Jim Buckley. Architecture consistency: State of the practice, challenges and requirements. *Empirical Software Engineering*, 23:224–258, 2018.

[30] Claudia Litvak, Leandro Antonelli, Gustavo Rossi, and Nora Gigante. Improving the identification of conflicts in collaborative requirements engineering. In *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 872–877, 2018.

[31] Haibin Yang, Chang Xu, Xiaoxing Ma, Linghao Zhang, Chun Cao, and Jian Lu. Consview: Towards application-specific consistent context views. In *2012 IEEE*

36th Annual Computer Software and Applications Conference, pages 632–637, 2012.

[32] Xiaoqing (Frank) Liu, Eric Christopher Barnes, and Juha Erik Savolainen. Conflict detection and resolution for product line design in a collaborative decision making environment. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, page 1327–1336, 2012.

[33] C Lottaz, I.F.C Smith, Y Robert-Nicoud, and B.V Faltings. Constraint-based support for negotiation in collaborative design. *Artificial Intelligence in Engineering*, 14(3):261–280, 2000.

[34] Hoh In and Barry W. Boehm. Using winwin quality requirements management tools: A case study. *Annals of Software Engineering*, 11:141–174, 2001.

[35] Jonathan Lee, Nien-Lin Xue, and Jong-Yih Kuo. Structuring requirement specifications with goals. *Information and Software Technology*, 43(2):121–135, 2001.

[36] Hoh In and S. Roy. Issues of visualized conflict resolution. In *Proceedings Fifth IEEE International Symposium on Requirements Engineering*, pages 314–315, 2001.

[37] H. In and S. Roy. Visualization issues for software requirements negotiation. In *25th Annual International Computer Software and Applications Conference. COMPSAC 2001*, pages 10–15, 2001.

[38] Xiaoqing Frank Liu, S. Raorane, Man Zheng, and Ming Leu. An internet based intelligent argumentation system for collaborative engineering design. In *International Symposium on Collaborative Technologies and Systems (CTS'06)*, pages 318–325, 2006.

[39] Xin Guo, Ying Liu, Wu Zhao, Jie Wang, and Ling Chen. Supporting resilient conceptual design using functional decomposition and conflict resolution. *Advanced Engineering Informatics*, 48:101262, 2021.

[40] Geet Sandhu and Sunil Sikka. State-of-art practices to detect inconsistencies and ambiguities from software requirements. In *International Conference on Computing, Communication Automation*, pages 812–817, 2015.

[41] Angela Borchert, Nicolas Emilio Díaz Ferreyra, and Maritta Heisel. Balancing trust and privacy in computer-mediated introduction: Featuring risk as a determinant for trustworthiness requirements elicitation. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*, ARES '20, 2020.

[42] Renzo Degiovanni, Facundo Molina, Germán Regis, and Nazareno Aguirre. A genetic algorithm for goal-conflict identification. In *2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 520–531, 2018.

[43] Weilin Luo, Hai Wan, Xiaotong Song, Binhao Yang, Hongzhen Zhong, and Yin Chen. How to identify boundary conditions with contrasty metric? In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pages 1473–1484, 2021.

[44] Yawen Wang, Lin Shi, Mingyang Li, Qing Wang, and Yun Yang. A deep context-wise method for coreference detection in natural language requirements. In *2020 IEEE 28th International Requirements Engineering Conference (RE)*, pages 180–191, 2020.

[45] Renzo Degiovanni, Pablo Castro, Marcelo Arroyo, Marcelo Ruiz, Nazareno Aguirre, and Marcelo Frias. Goal-conflict likelihood assessment based on model counting. In *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*, pages 1125–1135, 2018.

[46] Varsha Veerappa and Emmanuel Letier. Understanding clusters of optimal solutions in multi-objective decision problems. In *2011 IEEE 19th International Requirements Engineering Conference*, pages 89–98, 2011.

[47] Alejandro Salado and Roshanak Nilchiani. The tension matrix and the concept of elemental decomposition: Improving identification of conflicting requirements. *IEEE Systems Journal*, 11(4):2128–2139, 2017.

[48] S.S. Khan and M. Jaffar-ur Rehman. A survey on early separation of concerns. In *12th Asia-Pacific Software Engineering Conference (APSEC'05)*, pages 7 pp.–, 2005.

[49] Nathan Weston, Ruzanna Chitchyan, and Awais Rashid. Formal semantic conflict detection in aspect-oriented requirements. *Requirements Engineering*, 14:247, 2009.

[50] Unnati Shah, Sankita Patel, and Devesh Jinwala. An ontological approach to specify conflicts among non-functional requirements. In *Proceedings of the 2019 2nd International Conference on Geoinformatics and Data Analysis*, ICGDA 2019, page 145–149, 2019.

[51] Renzo Degiovanni, Nicolas Ricci, Dalal Alrajeh, Pablo Castro, and Nazareno Aguirre. Goal-conflict detection based on temporal satisfiability checking. In *2016*

31st IEEE/ACM International Conference on Automated Software Engineering (ASE), pages 507–518, 2016.

[52] Scott Sigman and Xiaoqing Frank Liu. A computational argumentation methodology for capturing and analyzing design rationale arising from multiple perspectives. *Information and Software Technology*, 45(3):113–122, 2003.

[53] Tan Lai Heng and Lim Tong Ming. Using multi-coordinated views with agent communication protocol to detect and resolve inconsistent requirements to improve accuracy. In *2010 International Symposium on Information Technology*, volume 2, pages 1041–1044, 2010.

[54] Arshad Ahmad, José Luis Barros Justo, Chong Feng, and Arif Ali Khan. The impact of controlled vocabularies on requirements engineering activities: A systematic mapping study. *Applied Sciences*, 10(21), 2020.

[55] Thiago Viana, Andrea Zisman, and Arosha K. Bandara. Towards a framework for managing inconsistencies in systems of systems. In *Proceedings of the International Colloquium on Software-Intensive Systems-of-Systems at 10th European Conference on Software Architecture*, SiSoS@ECSA '16, 2016.

[56] Yu Peng, Bing Li, Jian Wang, and Zhengli Liu. An approach of crossover service goal convergence and conflicts resolution. In *2020 IEEE World Congress on Services (SERVICES)*, pages 225–230, 2020.

[57] Saheed A. Busari and Emmanuel Letier. Radar: A lightweight tool for requirements and architecture decision analysis. In *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, pages 552–562, 2017.

[58] Pradeep K. Murukannaiah, Anup K. Kalia, Pankaj R. Telangy, and Munindar P. Singh. Resolving goal conflicts via argumentation-based analysis of competing hypotheses. In *2015 IEEE 23rd International Requirements Engineering Conference (RE)*, pages 156–165, 2015.

[59] Amirreza Masoumzadeh, Morteza Amini, and Rasool Jalili. Conflict detection and resolution in context-aware authorization. In *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, volume 1, pages 505–511, 2007.

[60] Jennifer Horkoff, Rick Salay, Marsha Chechik, and Alessio Di Sandro. Supporting early decision-making in the presence of uncertainty. In *2014 IEEE 22nd International Requirements Engineering Conference (RE)*, pages 33–42, 2014.

[61] Duaa Alkubaisy, Karl Cox, and Haralambos Mouratidis. Towards detecting and mitigating conflicts for privacy and security requirements. In *2019 13th International Conference on Research Challenges in Information Science (RCIS)*, pages 1–6, 2019.

[62] Rainara Maia Carvalho. Dealing with conflicts between non-functional requirements of ubicomp and iot applications. In *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pages 544–549, 2017.

[63] Ahmed Taha, Patrick Metzler, Ruben Trapero, Jesus Luna, and Neeraj Suri. Identifying and utilizing dependencies across cloud security services. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '16, page 329–340, 2016.

[64] Prawee Sriplakich, Xavier Blanc, and Marie-pierre Gervais. Supporting collaborative development in an open mda environment. In *2006 22nd IEEE International Conference on Software Maintenance*, pages 244–253, 2006.

[65] Alejandro Salado and Roshanak Nilchiani. The concept of order of conflict in requirements engineering. *IEEE Systems Journal*, 10(1):25–35, 2016.

[66] Ebrahim Bagheri and Faezeh Ensan. Consolidating multiple requirement specifications through argumentation. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, SAC '11, page 659–666, 2011.

[67] Bertrand Verlaine, Ivan J. Jureta, and Stéphane Faulkner. Optima: a domain-specific model for prioritization and conflicts management in requirements engineering for services intermediaries. *Service Oriented Computing and Applications*, 8:175–190, 2014.

[68] Huma Hayat Khan, Nauman Malik, Muhammad Usman, and Naveed Ikram. Impact of changing communication media on conflict resolution in distributed software development projects. In *2011 Malaysian Conference in Software Engineering*, pages 189–194, 2011.

[69] Tingyang Gu, Minyan Lu, and Luyi Li. Extracting interdependent requirements and resolving conflicted requirements of safety and security for industrial control systems. In *2015 First International Conference on Reliability Systems Engineering (ICRSE)*, pages 1–8, 2015.

[70] Yigong Wang, Hongqi Zhang, Xiangdong Dai, and Jiang Liu. Conflicts analysis and resolution for access control policies. In *2010 IEEE International Conference on Information Theory and Information Security*, pages 264–267, 2010.

[71] Zhang Yikun, Yin Peng, Cui Duwu, and Xia Hui. A method of requirement inconsistency analysis. In *31st Annual International Computer Software and Applications Conference (COMPSAC 2007)*, volume 1, pages 211–214, 2007.

[72] Juha Savolainen and Tomi Mannisto. Conflict-centric software architectural views: Exposing trade-offs in quality requirements. *IEEE Software*, 27(6):33–37, 2010.

[73] Raian Ali, Fabiano Dalpiaz, and Paolo Giorgini. Reasoning with contextual requirements: Detecting inconsistency and conflicts. *Information and Software Technology*, 55(1):35–57, 2013.

[74] Alberto Sardinha, Ruzanna Chitchyan, Nathan Weston, Phil Greenwood, and Awais Rashid. Ea-analyzer: Automating conflict detection in aspect-oriented requirements. In *2009 IEEE/ACM International Conference on Automated Software Engineering*, pages 530–534, 2009.

[75] Parita Jain, Arun Sharma, and Laxmi Ahuja. The impact of agile software development process on the quality of software product. In *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pages 812–815, 2018.

[76] Kuldar Taveter and Tahira Iqbal. Theory of constructed emotion meets re. In *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*, pages 383–386, 2021.

Appendix

I. The list of the papers that are used in Systematic Literature Review

Table 25 shows an overview of 66 articles that have been selected for the SLR. It consists of details such as reference number, article title, author's name(s), and publication year.

Table 25. The list of the papers that are used in Systematic Literature Review

Identifier	Article title	Authors name(s)	Publication Year
[2]	Stakeholder-Centric Clustering Methods for Conflict Resolution in the Requirements Engineering Process	Ishaya Gambo, Kuldar Taveter	2022
[3]	Intelligent identification and resolution of software requirement conflicts: Assessment and evaluation	Aldekhail M, Almasri M	2021
[11]	An ontological framework to manage the relative conflicts between security and usability requirements	Dewi Mairiza, Didar Zowghi	2010
[12]	Applying WinWin to quality requirements: a case study	Hoh In, B. Boehm, T. Rodger, M. Deutsch	2001
[13]	Conflict characterization and Analysis of Non Functional Requirements: An experimental approach	Dewi Mairiza, Didar Zowghi, Vincenzo Gervasi	2013
[14]	Identifying requirements conflicts and cooperation: how quality attributes and automated traceability can help	A. Egyed, P. Grunbacher	2004
[15]	Towards Development of Complete and Conflict-Free Requirement	Abha Moitra, Kit Siu, Andrew Crapo, Harsh Chamarthi, Michael Durling, Meng Li, Han Yu, Panagiotis Manolios, Michael Meiners	2018
[16]	Scalable analysis of real-time requirements	Vincent Langenfeld, Daniel Dietsch, Bernd Westphal Jochen Hoenicke, Amalinda Post	2019

[17]	A Scalable Operational Framework for Requirements Validation Using Semantic and Functional Models	Issa Atoum	2019
[18]	Using Argumentation to Explain Ambiguity in Requirements Elicitation Interviews	Yehia Elrakaiby, Alessio Ferrari, Paola Spoletin, Stefania Gnesi, Bashar Nuseibeh	2017
[19]	Ontology-based automated support for goal–use case model analysis	Tuong Huan Nguyen, John C. Grundy, Mohamed Almorsy	2016
[20]	A Bird's Eye View of Natural Language Processing and Requirements Engineering	Assad Alzayed, Ahmed Al-Hunaiyyan	2021
[21]	MaramaAIC: tool support for consistency management and validation of requirements	Massila Kamalrudin, John Hosking, John Grundy	2018
[22]	Detecting Web requirements conflicts and inconsistencies under a model-based perspective	M.J.Escalona, M.Urbieta, G.Rossi, J.A.Garcia-Garciaa, E. Robles Luna	2013
[23]	Analysis of Conflicts among Non-Functional Requirements Using to Integrated Analysis of Functional and Non-Functional Requirements	Vishal Sadana, Xiaoqing Frank Liu	2007
[24]	Tackling the requirements jigsaw puzzle	Maria Pinto-Albuquerque, Awais Rashid	2014
[25]	Automated Software Tool Support for Checking the Inconsistency of Requirements	Massila Kamalrudin	2009
[26]	Domain ontologies in the context of Requirements Engineering	Paulo Afonso Parreira, Rosângela Aparecida Dellosso Penteado	2015
[27]	Using default reasoning to discover inconsistencies in natural: language requirements	D. Zowghi, V. Gervasi, A. McRae	2001
[28]	Static Checking Consistency of Temporal Requirements for Control Software	Natalia Garanina, Dmitry Koznov	2021
[29]	Architecture consistency: State of the practice, challenges and	Nour Ali, Sean Baker, Ross O'Crowley, Sebastian	2018

	requirements	Herold, Jim Buckley	
[30]	Improving the identification of conflicts in collaborative requirements engineering	Claudia Litvak, Leandro Antonelli, Gustavo Rossi Nora Gigante	2018
[31]	ConsView: Towards Application-Specific Consistent Context Views	Haibin Yang, Chang Xu, Xiaoxing Ma, Linghao Zhang Chun Cao, Jian Lu	2012
[32]	Conflict detection and resolution for product line design in a collaborative decision making environment	Xiaoqing Frank Liu, Eric Christopher Barnes, Juha Erik Savolainen	2012
[33]	Constraint-based support for negotiation in collaborative design	C. Lottaz, I.F.C Smith, Y Robert-Nicoud B.V Faltings	2000
[34]	Using WinWin Quality Requirements Management Tools: A Case Study	Hoh In, Barry W. Boehm	2001
[35]	Structuring requirement specifications with goals	Jonathan Leea, Nien-Lin Xue, Jong-Yih Kuo	2001
[36]	Issues of visualized conflict resolution	Hoh In, S. Roy	2001
[37]	Visualization issues for software requirements negotiation	H. In, S. Roy	2001
[38]	An Internet Based Intelligent Argumentation System for Collaborative Engineering Design	Xiaoqing Frank Liu, S. Raorane, Man Zheng, Ming Leu	2006
[39]	Supporting resilient conceptual design using functional decomposition and conflict resolution	Guo X., Liu Y., Zhao W., Wang J., Chen L.	2021
[40]	State-of-art practices to detect inconsistencies and ambiguities from software requirements	Geet Sandhu, Sunil Sikka	2015
[41]	Balancing trust and privacy in computer-mediated introduction: featuring risk as a determinant for trustworthiness requirements elicitation	Angela Borchert, Nicolas Emilio Díaz Ferreyra, Maritta Heisel	2020
[42]	A Genetic Algorithm for Goal-Conflict Identification	Renzo Degiovanni, Facundo Molina, Germán Regis, Nazareno Aguirre	2018
[43]	How to Identify Boundary	Weilin Luo, Hai Wan,	2021

	Conditions with Contrasty Metric?	Xiaotong Song, Binhao Yang, Hongzhen Zhong, Yin Chen	
[44]	A Deep context-wise method for coreference detection in natural language requirements	Yawen Wang, Lin Shi Mingyang Li, Qing Wang, Yun Yang	2020
[45]	Goal-Conflict Likelihood Assessment Based on Model Counting	Renzo Degiovanni, Pablo Castro, Marcelo Arroyo, Marcelo Ruiz, Nazareno Aguirre, Marcelo Frias	2018
[46]	Understanding clusters of optimal solutions in multi-objective decision problems	Varsha Veerappa, Emmanuel Letier	2011
[47]	The Tension Matrix and the Concept of Elemental Decomposition: Improving Identification of Conflicting Requirements	Alejandro Salado, Roshanak Nilchiani	2015
[48]	A survey on early separation of concerns	S.S. Khan, M. Jaffar-ur-Rehman	2005
[49]	Formal semantic conflict detection in aspect-oriented requirements	Nathan Weston, Ruzanna Chitchyan, Awais Rashid	2009
[50]	An Ontological Approach to Specify Conflicts among Non-Functional Requirements	Unnati Shah, Sankita Patel, Devesh Jinwala	2019
[51]	Goal-Conflict Detection Based on Temporal Satisfiability Checking	Renzo Degiovanni, Nicolas Ricci, Dalal Alrajeh, Pablo Castro, Nazareno Aguirre	2016
[52]	A computational argumentation methodology for capturing and analyzing design rationale arising from multiple perspectives	Scott Sigman, Xiaoqing Frank Liu	2003
[53]	Using multi-coordinated views with agent communication protocol to detect and resolve inconsistent requirements to improve accuracy	Tan Lai Heng, Lim Tong Ming	2010
[54]	The impact of controlled vocabularies on requirements engineering activities: A systematic mapping study	Arshad Ahmad, José Luis Barros Justo, Chong Feng, Arif Ali Khan	2020
[55]	Towards a framework for managing	Thiago Viana, Andrea	2016

	inconsistencies in systems of systems	Zisman, Arosha K. Bandara	
[56]	An Approach of Crossover Service Goal Convergence and Conflicts Resolution	Yu Peng, Bing Li, Jian Wang, Zhengli Liu	2020
[57]	RADAR: A Lightweight Tool for Requirements and Architecture Decision Analysis	Saheed A. Busari, Emmanuel Letier	2017
[58]	Resolving goal conflicts via argumentation-based analysis of competing hypotheses	Pradeep K. Murukannaiah, Anup K. Kalia, Pankaj R. Telangy, Munindar P. Singh	2015
[59]	Conflict Detection and Resolution in Context-Aware Authorization of	Amirreza Masoumzadeh, Morteza Amini, Rasool Jalili	2007
[60]	Supporting early decision-making in the presence of uncertainty	Jennifer Horkoff, Rick Salay, Marsha Chechik, Alessio Di Sandro	2014
[61]	Towards Detecting and Mitigating Conflicts for Privacy and Security Requirements	Duaa Alkubaisy, Karl Cox	2019
[62]	Dealing with Conflicts Between Non-functional Requirements of UbiComp and IoT Applications	Rainara Maia Carvalho	2017
[63]	Identifying and Utilizing Dependencies Across Cloud Security Services	Ahmed Taha, Patrick Metzler, Ruben Trapero, Jesus Luna, Neeraj Suri	2016
[64]	Supporting Collaborative Development in an Open MDA Environment	Prawee Sriplakich, Xavier Blanc, Marie-pierre Gervais	2006
[65]	The Concept of Order of Conflict in Requirements Engineering	Alejandro Salado, Roshanak Nilchiani	2014
[66]	Consolidating multiple requirement specifications through argumentation	Ebrahim Bagheri, Faezeh Ensan	2011
[67]	Optima: a domain-specific model for prioritization and conflicts management in requirements engineering for services intermediaries	Bertrand Verlaine, Ivan J. Jureta, Stéphane Faulkner	2016
[68]	Impact of changing communication media on conflict resolution in distributed software development	Huma Hayat Khan, Nauman Malik, Muhammad Usman,	2011

	projects	Naveed Ikram	
[69]	Extracting interdependent requirements and resolving conflicted requirements of safety and security for industrial control systems	Tingyang Gu, Minyan Lu, Luyi Li	2015
[70]	Conflicts analysis and resolution for access control policies	Yigong Wang, Hongqi Zhang, Xiangdong Dai, Jiang Liu	2010
[71]	A Method of Requirement Inconsistency Analysis	Zhang Yikun, Yin Peng, Cui Duwu, Xia Hui	2007
[72]	Conflict-Centric Software Architectural Views: Exposing Trade-Offs in Quality Requirements	Juha Savolainen, Tomi Mannisto	2010
[73]	Reasoning with contextual requirements: Detecting inconsistency and conflicts	Raian Ali, Fabiano Dalpiaz, Paolo Giorgini	2013
[74]	EA-Analyzer: Automating Conflict Detection in Aspect-Oriented Requirements	Alberto Sardinha, Ruzanna Chitchyan, Nathan Weston, Phil Greenwood, Awais Rashid	2009

II. Results of Quality Scores of Papers

Table 26. Results of quality scores of selected papers

Identi-fier	QAQ1	QAQ2	QAQ3	QAQ4	QAQ5	Total
[2]	1	0.5	0	1	1	3.5
[3]	1	0	1	1	1	4
[11]	1	0	0	0.5	1	2.5
[12]	1	1	1	1	0	4
[13]	1	0	1	1	1	4
[14]	0.5	0	0	0.5	1	2
[15]	0.5	0	1	0.5	0	2
[16]	1	0	1	0.5	1	3.5
[17]	1	0	1	0.5	1	3.5
[18]	1	0	1	0	1	3
[19]	1	0	1	1	1	4
[20]	1	0.5	1	1	0	3.5
[21]	1	0	1	1	1	4
[22]	1	0.5	1	0.5	0	3
[23]	1	1	0	1	1	4
[24]	1	0.5	0	1	1	3.5
[25]	1	0	1	1	0.5	3.5
[26]	1	0	0	0	0	1
[27]	1	0	1	0	1	3
[28]	1	0	0	0	1	2
[29]	1	0	1	1	0	3
[30]	1	0	0	0	1	2
[31]	1	0	0	0.5	1	2.5
[32]	1	1	1	0	1	4
[33]	1	0.5	1	0	1	3.5
[34]	1	1	0	1	0	3
[35]	1	1	1	0.5	1	4.5
[36]	1	1	0	1	0	3
[37]	1	0.5	0	1	1	3.5
[38]	1	0	1	0.5	1	3.5
[39]	1	0.5	1	0.5	1	4
[40]	1	0	1	1	1	4
[41]	0.5	0.5	1	0	1	3

[42]	1	0	1	0	1	3
[43]	1	0	0	1	1	3
[44]	1	0	1	1	1	4
[45]	1	0	0	0.5	1	2.5
[46]	1	0	1	0.5	1	3.5
[47]	1	0.5	1	1	1	4.5
[48]	1	1	1	0	0	3
[49]	1	0	1	0.5	1	3.5
[50]	1	0	1	1	1	4
[51]	1	0	1	0.5	1	3.5
[52]	1	1	1	0.5	1	4.5
[53]	1	0.5	1	1	1	4.5
[54]	1	0	1	0	0	2
[55]	1	0	0	0	1	2
[56]	1	0	0	1	1	3
[57]	1	0	0	1	1	3
[58]	1	1	1	1	1	5
[59]	1	0	1	0.5	1	3.5
[60]	0.5	0	0	0	1	1.5
[61]	1	1	0	1	1	4
[62]	0.5	0	0	1	1	2.5
[63]	1	0	0	1	1	3
[64]	0.5	0	0	0.5	1	2
[65]	1	0	0	0.5	1	2
[66]	1	1	1	1	1	5
[67]	1	1	1	0	1	4
[68]	0.5	1	1	1	1	4.5
[69]	0.5	0	1	1	1	3.5
[70]	1	1	1	0.5	1	4.5
[71]	1	0	1	0.5	1	3.5
[72]	0.5	0	1	0.5	1	3
[73]	1	0	1	0	1	3
[74]	0.5	0	1	1	1	3.5

III. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Orkhan Jamaladdinov**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Systematic Literature Review on the Extent of Conflict Identification and Resolution Research and Practice in Requirements Engineering of Socio-technical Systems: Where are we now? ,

   supervised by Ishaya Peni Gambo.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Orkhan Jamaladdinov
07/08/2022