The Journal of Systems & Software 197 (2023) 111574

Contents lists available at ScienceDirect

The Journal of Systems & Software

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

Software engineering research on the Robot Operating System: A systematic mapping study[☆]

Michel Albonico[a], Milica Đorđević[b], Engel Hamer[b], Ivano Malavolta[b,*]

[a] Federal University of Technology, Paraná (UTFPR), Francisco Beltrão, Brazil
[b] Vrije Universiteit Amsterdam, The Netherlands

ARTICLE INFO

Article history:
Received 5 April 2022
Received in revised form 5 October 2022
Accepted 28 November 2022
Available online 2 December 2022

Keywords:
ROS
Robotic systems
Software engineering
Systematic mapping study

ABSTRACT

The *Robot Operating System* (ROS) has become the de-facto standard framework for robotics software, and a great part of commercial robots is expected to have at least one ROS package on board in the coming years. For good quality, robotics software should rely on strong software engineering principles. In this paper, we perform a systematic mapping study on several works in software engineering on ROS, published at the top software engineering and robotics venues. Our goal is to analyze and evaluate such state-of-the-art regarding its relevance to the robotics software industry. The potentially-relevant studies are subject to a rigorously defined selection process. This results in a set of 63 primary studies on software engineering research on ROS. Those primary studies are then qualitatively analyzed according to a rigorously-defined classification framework. The results are of interest to both researchers and practitioners: (i) we provide an up-to-date overview of the state of the art on software engineering research on ROS and its potential for industrial adoption, (ii) a broad discussion of the research area as a whole, and (iii) point out routes of action for a better alignment between research and industry.

1. Introduction

In the last years, we have witnessed a growth in the use of robots in both industrial and societal contexts (Jung and Lim, 2020). As the scale and scope of today's robotic systems grow, designing and developing the software commanding them is becoming increasingly difficult (Quigley et al., 2009). In robotics, mainly due to the high complexity involved, it is common to rely on frameworks and software suites, among which the *Robot Operating System* (ROS (Macenski et al., 2022)) has become a de-facto standard. ROS is a collection of open-source robotics libraries and tools centered around robotics software development. It abstracts the underlying hardware and promises a great extent of modularity through an ever-increasing number of packages in the ROS ecosystem. Presently, the ROS index lists 6,843 available packages across its 15 active distributions[1]. In addition to packages, ROS facilitates robots communication based on principles that favor loose coupling of distributed processes (Quigley et al., 2009).

While ROS has been researched in the academic community for around a decade now, its adoption in the industry is in expansion as well. The breaking changes introduced in ROS 2, the latest version of ROS, are tailored to fit the industrial standards, hence the expansion of ROS in the industrial sector is expected to grow further (Macenski, 2020). Some predictions state that nearly 55% of all commercial robots shipped by 2024 will have at least one ROS package on-board (Su, 2019).

Given the great number of ROS-related scientific studies in the literature, the **goal** of this work is to analyze the ones that focus on software engineering aspects related to ROS, and evaluate their potential for industrial adoption. To achieve such goal, we perform a **systematic mapping** of scientific studies in software engineering for ROS. We start by selecting 224 ROS-related studies published at top venues in software engineering and 55 at top venues in robotics, a total of 279 studies. After following a rigorous procedure defined a priori, we keep a set of 32 studies, and follow a snowballing procedure (Wohlin, 2014) to undercover further related studies in other venues. This results in 2747 linked studies, where only 31 were selected as studies on software engineering for ROS by following the same procedure as a priori, which makes a total of 63 primary studies to be analyzed. Finally, a team of researchers analyzes all 63 primary studies qualitatively according to a well-defined classification framework, synthesizing the extracted data and drawing the main conclusions.

[☆] Editor: Dr. Martin Pinzger.
* Corresponding author.
E-mail addresses: michelalbonico@utfpr.edu.br (M. Albonico), m.djordjevic@student.vu.nl (M. Đorđević), e.hamer@student.vu.nl (E. Hamer), i.malavolta@vu.nl (I. Malavolta).
[1] https://index.ros.org/stats/

https://doi.org/10.1016/j.jss.2022.111574
0164-1212/

Table 1
GQM research goal definition.

Purpose	Analyze
Issue	the state-of-the-art and potential for industrial adoption of
Object	software engineering research on the Robot Operating System
Viewpoint	from researchers' and practitioners' point of view.

The main **contributions** of this study are the following:

- A *systematic mapping* of the state of the art in software engineering research on ROS-based systems, valuable for both researchers and practitioners.
- A *classification framework* that can be used by researchers to contextualize future work with respect to the current research landscape on software engineering for ROS-based systems.
- A *discussion* regarding the main implications of the obtained results and concrete routes of action for both practitioners and researchers working on the software engineering aspects of ROS-based systems.
- A complete *replication package* containing the scripts and spreadsheets used in this work, which enables the independent verification and replication of this study.

The remainder of this paper is organized as follows. In Section 2, we describe in detail the design of this study. Following this, in Section 3 we first consider some demographics of the set of primary studies in our analysis, then present the results obtained from data extraction and synthesis. In Section 4, we evaluate the rigor and industrial relevance of the analyzed studies. In Section 5, we discuss the obtained results and present concrete routes of action for both researchers and practitioners. We present the threats to validity and measures to mitigate them in Section 6. We provide a brief overview of existing work related to this study in Section 7 before finally concluding the answers to the research questions in Section 8.

2. Study design

We base this study design on established guidelines for systematic mapping studies (Petersen et al., 2015; Kitchenham and Brereton, 2013; Wohlin et al., 2012). We start by defining the goal and research questions (Section 2.1), then conduct an extensive search for primary studies with previously established inclusion and exclusion criteria (Section 2.2), and extract the data from the selected primary studies (Section 2.3). A full *replication package* is publicly available on *GitHub*.[2] The replication package includes all raw data obtained throughout this study, as well as the scripts used for automating the search, data analysis, and visualization, and the spreadsheet used for the data extraction, which contains the link to the analyzed primary studies.

2.1. Goal and research questions

Table 1 defines the goal of this research according to the *Goal, question, metric*-framework (Basili et al., 1994). This goal will be achieved by answering the following research questions:

- **RQ1** – What is the **state-of-the-art** in software engineering research on ROS? In order to better scope RQ1, we further refine it into the following 4 research questions.
  - **RQ1.1** – What are the **research aspects** considered in the area of software engineering research on ROS?
  - **RQ1.2** – What are the **robots** used in the area of software engineering research on ROS?
  - **RQ1.3** – What are the **ROS ecosystem aspects** considered in the area of software engineering research on ROS?
  - **RQ1.4** – What are the **software engineering aspects** considered in the area of software engineering research on ROS?
- **RQ2** – What is the **potential industrial adoption** of existing software engineering research on ROS?

By answering RQ1 (and its sub-research questions) we support the software engineering and robotics scientific communities by providing a detailed and up-to-date overview of the landscape of software engineering research on ROS. By answering RQ2 we support both the scientific and industrial communities as we will elaborate on the adoptability of current research results in the context of real-world industrial projects.

2.2. Search and selection

The search and selection process performed for this research is outlined in Fig. 1. It consists of 4 stages, which are explained in more detail throughout this section.

Stage 1 — Venues selection
To ensure that the primary studies of this research are of high quality, we construct a list of top venues in software engineering and robotics. Only studies published at these venues are considered in subsequent stages of this study. Our approach consists of the following steps:

1. Collect all entries from the top-20 list of venues in Software Engineering[3] and Robotics[4] in Google Scholar.
2. Collect all entries of *conferences* ranked A* or A from the 2020 CORE ranking.[5]
3. Collect all entries of *journals* ranked A* or A from the 2020 CORE ranking.[6]
4. Merge the lists of venues obtained from steps 1, 2 and 3, discarding duplicate entries.

We choose to use Google Scholar and CORE since they are both well-known directories of academic publishing and find widespread usage among the research community. The venues ranked in both partially overlap and such sources complement each other. Combining them mitigates the risks associated with the usage of a single source of truth. The full list of venues resulting from this approach is included in Appendix A. This list is used in Stage 2 as a querying parameter.

Stage 2 — Automatic search
In this stage, we aim at obtaining most of the scientific publications that are of interest for answering our research questions. As publications database we consider the *dblp computer science bibliography*.[7] More specifically, we consider the publicly-available snapshot of the whole DBLP database, downloaded on

[2] https://github.com/S2-group/SLR_SE_ROS_2022

[3] https://scholar.google.com/citations?view_op=top_venues&hl=us&vq=eng_softwaresystems

[4] https://scholar.google.com/citations?view_op=top_venues&hl=us&vq=eng_robotics

[5] http://portal.core.edu.au/conf-ranks/?search=4612&by=for&source=CORE2020&sort=arank&page=1

[6] http://portal.core.edu.au/jnl-ranks/?search=0803&by=for&source=CORE2020&sort=arank&page=1

[7] https://dblp.uni-trier.de/

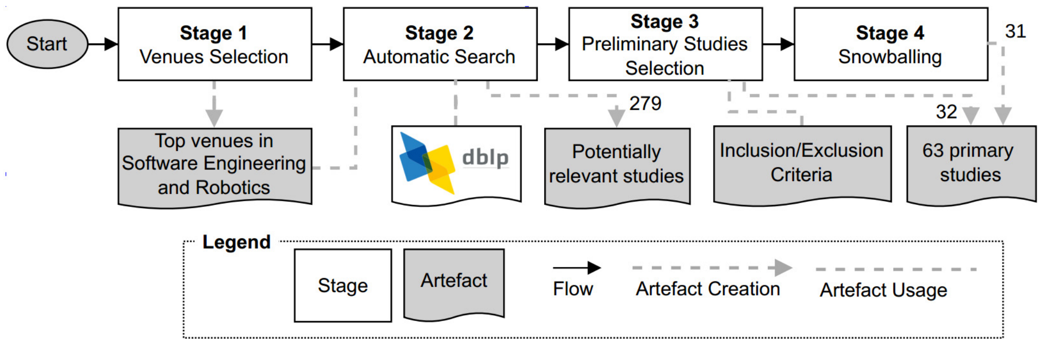


Fig. 1. Search and selection process.

February 2021 (The dblp team, 2021). Using this snapshot contributes to the reproducibility of this study, whereas entries in (general-purpose) search engines such as Google Scholar[8] are generally more volatile due to the opacity of the used search algorithms.

We queried the DBLP database by executing the following query on the titles of its contained publications:

`ROS` **OR** `robot`

We keep the search string as generic as possible in order to capture any study that may be of interest to our research questions. It is important to note that the search string we used for robotic venues does not include the term *robot* since by design studies published in robotic venues are already expected to be focussing on robotic systems. Here, `ROS` is matched case-sensitively, whereas `robot` is not; this is to discard any potentially-relevant study with a title that contains *ros* as a sub-string of another (unrelated) word, such as *across*.

We developed a dedicated script to automatically parse and search the XML dump of the DBLP database (The dblp team, 2021); the source code of our script is publicly available in the replication package of this study. We configure the script to search only for studies published at the venues selected in Stage 1, resulting in a list of 279 potentially relevant studies.

Stage 3 — Studies selection

The goal of this stage is to select those potentially-relevant studies that are relevant for answering our research questions, while discarding irrelevant ones. The main tool of this stage is a set of inclusion and exclusion criteria, which are described below.

Inclusion Criteria

I1. Studies that focus on the Robot Operating System (ROS).

I2. Studies that are written in English.

I3. Studies that are published in 2007 or later. Note that this is the year when ROS was created.[9]

I4. Studies that consider software engineering aspects *(such as software requirements, design, development, testing, or maintenance*).

Exclusion Criteria

E1. Studies that use ROS only for implementation, without providing details of the software engineering aspects.

E2. Studies that are extended by another paper in this review. In such cases, the most recent (and thus most complete) paper will be selected.

E3. Studies that are not available as full-text.

A potentially-relevant study is included (and thus proceeds to the next stage) if and only if it simultaneously satisfies *all* inclusion criteria (I1-I4) and *none* of the exclusion criteria (E1-E3). For each of the 279 primary studies from the previous stage, the above criteria are manually assessed by following three steps: (i) read the paper's title; (ii) read the paper's abstract; and (iii) read the paper's full text. A complete assessment is performed by two researchers, independently of one another. At the end, both assessments are compared for each paper and a disagreements are solved by means of a third researcher that works as an arbiter.

Out of 279 potentially relevant studies, both researchers agree to exclude 239 studies, and include 25, and had a disagreement about other 11 potentially relevant studies. The disagreements were resolved with the presence of the *arbiter*, with a final decision of including other 5 primary studies. In the end, we obtained a Cohen Kappa coefficient of 0.81 or 96.02%, which can be interpreted as an 'almost perfect agreement' (McHugh, 2012). This stage thus yields a final set of 32 studies, which are listed in Appendix B.

Stage 4 — Snowballing

From the selected studies, we conduct multiple rounds of snowballing, following the guidelines by Wohlin (2014). This improves the representativeness of our results and will potentially reveal additional studies.

In this stage, one researcher builds a dataset with studies that are related to the ones we selected in , *i.e., forward* and *backward snowballing* results. In the *backward snowballing* we insert into the dataset all the studies cited by those studies from , while in the *forward snowballing* we insert the studies that cite them. For the *forward snowballing* we consider the citations indexed by Google Scholar.

After the first round, the snowballing procedure results in 1279 potentially relevant studies. At this point, two researchers repeat *Stage 3* in parallel. They achieve an *almost perfect* agreement, with a *Cohen Kappa coefficient* of 0.96. After solving their conflicts, this round results in 25 additional.

Given the high agreement among the two researchers, only one of them proceeds with the remaining rounds. He repeats the snowballing procedure and study selection until no further study is revealed, which takes three additional rounds. Five other studies are revealed in the second round and three in the third.

All the four snowballing rounds led to a total of 2747 primary studies, 1456 with the *backward snowballing* and 1291 with the *forward snowballing*, from where we selected a total of 31 additional primary studies. This results in a total of 63 primary studies to be analyzed.

8 https://scholar.google.com
9 https://www.ros.org/history/

Table 2
The classification framework of this study.

Parameter	Type	Possible values	Description
		Research aspects (RQ1.1)	
Main contribution	Fixed	Architecture, Method, Model, Tool, Package	Most prevalent type of research contribution in a paper, adapted from Bozhinoski et al. (2019) (originally proposed in Petersen et al. (2008)).
Research strategy	Fixed	Evaluation research, Proposal of a solution, Validation research, Philosophical paper, Opinion paper, Personal experience paper	High-level perspective of a research paper, defined in Wieringa et al. (2006).
Research method	Open	Simulation-based experiment, Survey, Interview, etc.	Type of activities performed in a paper to achieve a set goal.
Future challenges and limitations	Open	Performance optimizations, Multi-language support, etc.	Future challenges and limitations of a paper explicitly specified by the authors
		Used robots (RQ1.2)	
Robotic platform	Open	Kobuki, TurtleBot, etc.	Any (simulated) robotic hardware considered in a paper.
Type of robot	Fixed	Mobile terrestrial, Mobile aquatic, Mobile airborne, Fixed	Categorization of robots used in a paper, based on their localization mechanism and operational environment as defined by Ben-Ari and Mondada (2018).
Cardinality	Fixed	Single robot, Multiple robots	Number of robots used in a paper.
		ROS ecosystem aspects (RQ1.3)	
ROS version	Fixed	ROS1, ROS2	Major ROS version that a robotic system in a paper runs.
ROS ecosystem level	Fixed	Filesystem Level, Computation Graph Level, Community Level	Categorization of all *levels*[a] of the ROS ecosystem a paper content relates to.
Communication paradigm	Open	Topics, Services, etc.	Means of inter-node communication used by a proposed solution in a paper.
		Software engineering aspects (RQ1.4)	
Knowledge Area	Fixed	Software requirements, Software design, Software construction, Software testing, Software maintenance, etc.	Knowledge areas that the main contributions of a paper belong to, defined in SWEBOK Bourque and Fairley (2004).
Application field	Open	Navigation task, Search and rescue, etc.	Categorization of tasks a robotic system performs in a paper.
Quality attributes considered	Fixed	Functional suitability, Performance efficiency, Compatibility, Usability, Reliability, Security, Maintainability, Portability	All quality attributes considered in a paper, according to the IEEE 25010 standard ISO/IEC 25010:2011 (2011).
		Potential for industrial adoption (RQ2)	
Rigor	Fixed	Low, Medium, High	Quantitative assessment of academic rigor of a paper, based on context, study design and validity as defined in Ivarsson and Gorschek (2011).
Industrial relevance	Fixed	0-4	Quantitative assessment of industrial relevance of a paper, based on subjects, context, scale and research methods applied as defined in Ivarsson and Gorschek (2011).

[a] http://wiki.ros.org/ROS/Concepts.

2.3. Data extraction

Once all the primary studies are selected, we perform a *content analysis* (manually reading their full text) for categorizing and coding the primary studies under broad thematic categories. To facilitate structured analysis and comparison of the selected primary studies, two researchers conduct this analysis by following a predefined *classification framework*. The *classification framework* is divided in 5 main groups: *(i) Research aspects*: main contribution of the paper, research strategy, research method, and future challenges and limitations; *(ii) Used robots*: robotic platform, type of robot, and cardinality; *(iii) ROS ecosystem aspects*: ROS version, ROS ecosystem level, and communication paradigm; *(iv) Software Engineering aspects*: knowledge area, application field, and quality attributes; *(v) Potential for industrial adoption*: rigor and industrial relevance. Table 2 shows the final classification framework, including the type of each parameter, examples of possible values, and a short description.

2.4. Data synthesis

After performing the data extraction, we synthesize the obtained data into quantitative and qualitative results. More specifically, we perform a *narrative synthesis* (Popay et al., 2006). This approach involves explaining the characteristics of primary studies through textual narrative summaries. The goal here is to provide evidence which can be used to substantiate answers to our research questions. Since researchers have a great agreement in the previous phases, here we divide the analysis among three researchers, where each one analyzes a portion of the primary studies. Then, we randomly validate each other analysis with a peer-reviewed process, correcting possible misinterpretations iteratively. We present the results of this synthesis in Section 3.

3. Results

In this section, we present the results from our analysis of the selected primary studies. We start with the primary study

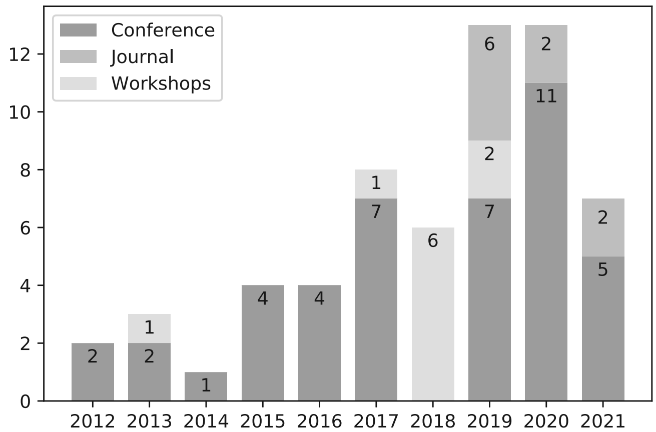


Fig. 2. Distribution of primary studies by year and type of venue.

demographic, then present the results of each parameter as defined in the classification framework (see Table 2). All the data analysis code and data is publicly available in the replication package[10], where it is also possible to see a complete list of studies classifications.

3.1. Demographics

Fig. 2 depicts the distribution of all the primary studies by year and the type of venue where they have been published at. The complete list of all 63 primary studies is presented in Appendices Appendix B and Appendix C. The majority of the primary studies has been published in recent years, namely, between 2017 and 2021 (49), with the peak of 15 publications in 2019. The growing trend in software engineering publications on ROS reflects can be seen as a reaction to the emerging challenges faced by researchers and practitioners when designing, developing, and testing complex robotic systems. The distribution by the type of venue shows that most of the primary studies are conference primary studies, with only 10 primary studies being published in journals and 10 primary studies in workshops.

Table 4 gives an overview of the distribution of the primary studies among their publication venues. For illustration reasons, this table is divided into two parts. The first part presents those venues whose themes and origin are in the Software Engineering area, while the second part presents those venues related to the Robotics area. In the table (and in the remainder of this primary study) we use the "P" prefix for primary studies emerging from the automatic search, while we use the "S" prefix for those primary studies emerging from the snowballing. We see that Software Engineering venues (37 primary studies) tend to be slightly more targeted than Robotics venues (26 primary studies). Among Software Engineering venues, the International Workshop on Robotics Software Engineering (RoSE) are the most frequently targeted venue, followed by other well-established venues, such as the International Conference on Software Engineering (ICSE) and the Journal of Systems and Software (JSS). Among Robotics venues (lower part of the table), we highlight the presence of the IEEE International Conference on Robotics Computing (IRC), the IEEE International Conference on Robotics and Automation (ICRA) and the International Conference on Intelligent Robots and Systems (IROS), with respectively 6, 6, and 5 publications each. It is important to state that ICRA and IROS are top venues in robotics, which indicates the interest of internationally-recognized robotics research groups in robotics software engineering.

[10] https://github.com/S2-group/SLR_SE_ROS_2022

3.2. Research aspects (RQ1.1)

In this section we discuss the main research aspects we extracted from the primary studies, namely: their main contributions, research strategies, applied research methods, and identified future challenges and limitations.

3.2.1. Main contribution

To gain insight into the current focus of research related to ROS, we consider the main contribution of the analyzed primary studies. Table 3 lists the primary study classification by their main contribution. Each primary study is categorized as one of the types of research contributions adapted from Bozhinoski et al. (2019) (and originally proposed in Petersen et al. (2008)), as follows:

- **Architecture**: Presents the fundamental concepts or properties of a robotic system using ROS embodied in its elements, relationships, and in the principles of its design and evolution (ISO, 2011).
- **Method**: Presents general concepts and working procedures to address specific concerns regarding ROS.
- **Model**: Presents information, representations, and abstractions to be used in the context of ROS.
- **Tool**: Presents any kind of developed tool or prototype related to ROS.
- **Package**: Presents a new package available for the ROS ecosystem.

More than half of the primary studies (35) propose a *method* as their main contribution. As an example, some methods describe guidelines for working with ROS or making design decisions (P8), whereas others propose reusable model-based techniques (S27) or other development practices in the ROS workflow (P2).

The second most prevalent contribution type is *architecture*, occurring in 13 out of 63 primary studies. This can be, for example, a robotic system for a use case which leverages functionality of ROS middleware (P19), or a novelty composition of ROS packages that together achieve new or improved functionality (P15).

A total of 12 primary studies present a *model* as their primary contribution. For instance, an empirical study on the community and ecosystem dynamics (P12), an analysis of ROS repositories for the existence of dependency bugs (P5), or a ranked list of the most frequent usage patterns of ROS functionalities and primitives (S17).

While the architectures, models, and methods rely largely on existing entities in the ROS ecosystem (such as packages or community collaboration), contributions to the ecosystem itself are less common. The development of a new *tool* occurs only in 7 primary studies, and only one primary study contributes with a ROS *package*.

3.2.2. Research strategy

We categorized the primary studies by the research strategies applied. For this purpose, we follow the classification defined by Wieringa et al. (2006). Even though the authors present their classification method in the context of requirements engineering, it can be applied to other fields without loss of generality. At the time of writing, the study by Wieringa et al. is cited 850 times on Google Scholar and their classification framework is reused in respectable systematic mapping studies in various areas, ranging from micro-service architectures (Alshuqayran et al., 2016), cloud migration (Jamshidi et al., 2013) to gamification (Sardi et al., 2017).

Fig. 3(a) shows that more than half of the primary studies (43) propose a *solution* for an identified problem. Given that the ROS ecosystem is relatively recent and in its expansion, it is

Table 3
Main contribution.

Contribution type	#primary studies	primary studies
Method	34	P2, P4, P6, P8, P10, P11, P16, P18, P20, P22, P23, P25, P26, P28, P31, S1, S2, S3, S4, S5, S10, S12, S18, S19, S21, S22, S23, S24, S25, S26, S27, S28, S29, S30, S31
Architecture	13	P1, P3, P7, P9, P14, P15, P19, P21, P24, P27, S12, S13, S16
Model	12	P5, P12, P13, P29, P32, S1, S8, S12, S16, S17, S20, S27
Tool	7	P17, P30, S6, S7, S9, S14, S15
Package	1	S23

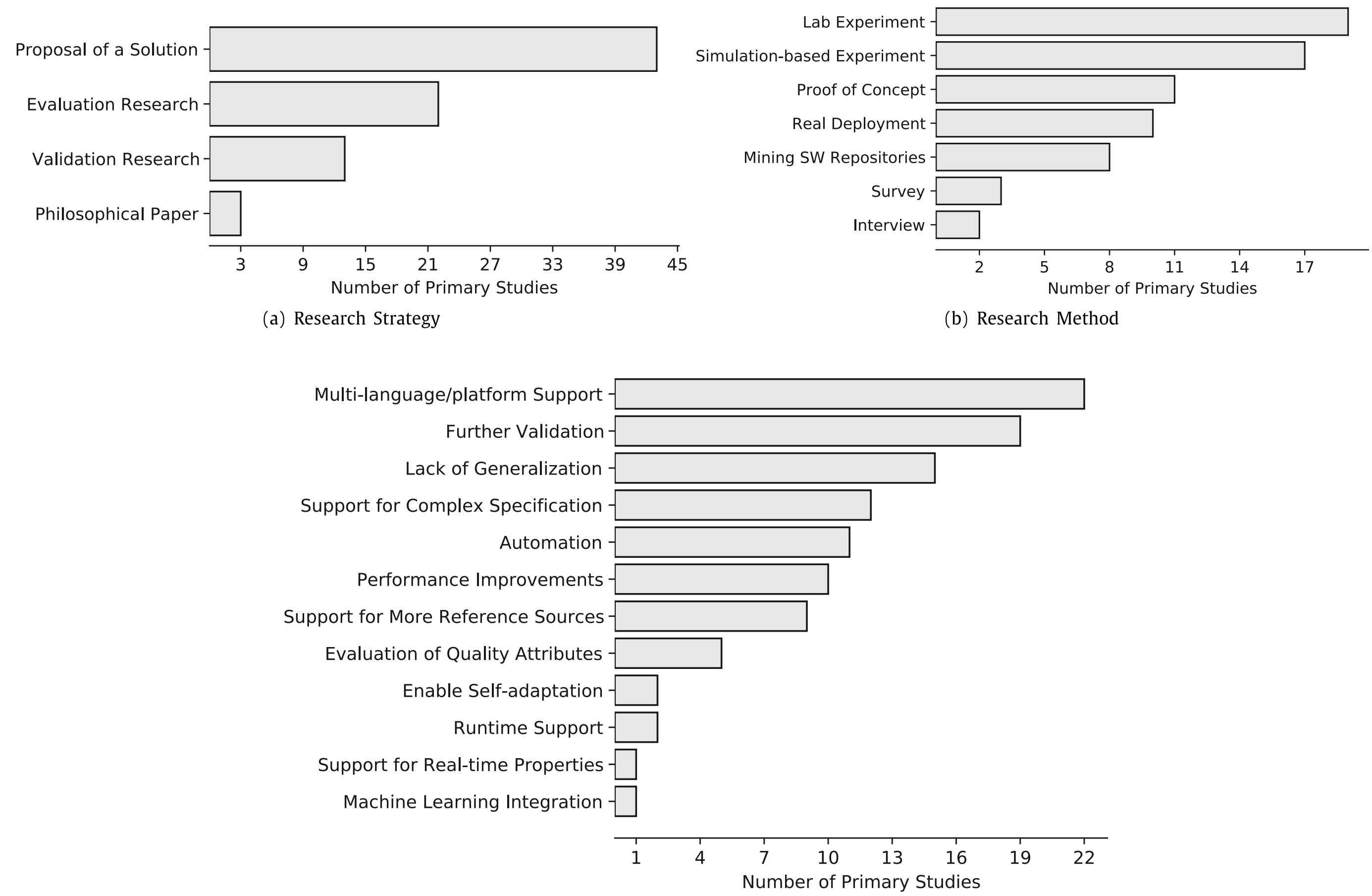
Table 4
Publication venues of the primary studies.

Venue	Acronym	#primary studies	primary studies
Software Engineering venues			
International Workshop on Robotics Software Engineering	RoSE	5	P3, P4, P9, P10, P11
International Conference on Software Engineering	ICSE	3	P5, P8, S6
Journal of Systems and Software	JSS	3	P15, P16, S13
International Conference on Model Driven Engineering Languages and Systems	MODELS	2	P19, P20
International Conference on Mining Software Repositories	MSR	2	S4, S12
IEEE SoutheastCon	SoutheastCon	2	S8, S23
International FME Workshop on Formal Methods in Software Engineering	FormaliSE	1	P6
International Symposium on Software Engineering for Adaptive and Self-Managing Systems	SEAMS	1	P7
ACM SIGSOFT International Workshop on Automating TEST Case Design	A-TEST	1	P22
ACM SIGSOFT International Symposium on Foundations of Software Engineering	SIGSOFT	1	P23
International Conference in Business Process Management	BPM	1	P1
IEEE International Conference on Engineering of Complex Computer Systems	ICECCS	1	P2
IEEE International Conference on Software aggregated results for the rigor and industrial relevanceenance and Evolution	ICSM	1	P12
Information and Software Technology	INFSOF	1	P13
International Symposium on Software Testing and Analysis	ISSTA	1	P14
International Workshop on Explainable Software	EXPLAIN	1	P17
ACM Workshop on Domain-specific Modeling	DSM	1	P21
IEEE Software	SOFTWARE	1	P24
IEEE Transactions on Reliability	TR	1	P25
International Artificial Intelligence and Data Processing Symposium	IDAP	1	S11
International Conference on Runtime Verification	RV	1	S18
International Conference of the Chilean Computer Science Society	SCCC	1	S21
ACM/IEEE International Conference on Human-Robot Interaction	HRI	1	S24
IEEE International Conference on Emerging Technologies and Factory Automation	ETFA	1	S3
Procedia Computer Science	CompSci	1	S1
International Workshop on Model-Driven Engineering Tools	MDETools	1	P18
Robotics venues			
IEEE International Conference on Robotic Computing	IRC	6	S2, S9, S10, S16, S26, S27
IEEE International Conference on Robotics and Automation	ICRA	6	P28, P29, P30, P31, S19, S28
International Conference on Intelligent Robots and Systems	IROS	5	P26, P27, S5, S15, S17
International Conference on Simulation, Modeling, and Programming for Autonomous Robots	SIMPAR	2	S14, S30
International Conference on Embedded Software	EMSOFT	1	S20
European Conference on Mobile Robots	ECMR	1	S22
Drone Systems Engineering and Rapid Simulation and Performance Evaluation: Methods and Tools	DroneSE	1	S7
Iberian Robotics Conference	ROBOT	1	S25
Robotics Open Access Journal MDPI	Robotics	1	S29
Journal of Intelligent and Robotic Systems	JIRS	1	P32
ACM Transactions on Embedded Computing Systems	TECS	1	S31

understandable that new solutions are being proposed. However, 13 primary studies complement their solution proposal with a thorough *validation* in a research setup. On the other hand, 22 *evaluation studies* observe different aspects of the ROS ecosystem and evaluate proposed solutions and techniques. The notable presence of such studies is an indicator of the ecosystem's maturity. For instance, a primary study analyses the dependency bugs in ROS (P5), another primary study evaluates best architectural practices (P8), and another primary study evaluates the practices on architecting robotics software (S13). Finally, there are only 3 *philosophical* primary studies that present reflections about the present and future of ROS, and no occurrences of (i) opinion and (ii) personal experience primary studies.

3.2.3. Research method

To complement the research strategy parameter presented in Section 3.2.2, we extracted the research methods that were applied in the context of validation and evaluation studies. Indeed, the primary studies applying either the validation or evaluation research strategies follow different research methods for assessing their proposed approaches; for example, some primary studies are based on practitioners' interviews (*e.g.*, P23), others



(a) Research Strategy

(b) Research Method

(c) Future Challenges

Fig. 3. Research aspects mapping.

focus on mining software repository techniques (*e.g.,* P8), others carry out in-the-lab controlled experiments (*e.g.,* P15), etc. According to our study design (see Section 2.3), the possible values for the research method parameter emerged in a bottom-up fashion during our content analysis sessions. The research methods applied in the analyzed primary studies are shown in Fig. 3(b).

Several primary studies (19) validate a solution in a *laboratory experiment*, either using real robots in a controlled environment (*e.g.,* P26) or by assigning programming tasks to human subjects (*e.g.,* P13). Such environments consist of relatively small rooms in the lab with custom obstacles that do not represent well real-world settings to a large extent. Additionally, 11 primary studies are validated in *real deployments* of robotic environment. Some studies are conducted on a commercial model of an electric wheelchair (*e.g.,* P3), whereas others are based on a commercial agriculture robot (AGROB V14) that operates in outdoor mountain vineyards (S29). Prior to the real deployment, some researchers perform multiple simulation-based experiments and analyze the differences between the results obtained in the simulation versus the real-world results (*e.g.,* P14). This is the typical flow that industry practitioners follow when developing a robotic system and it is rather unfortunate that researchers mostly do not complement their simulation-based experiments with real ones. For the results and conclusions to be widely accepted in both the research community and industry, they need to be proven on real systems.

Several primary studies (17) rely on *simulation-based experiments*, with the most commonly used simulation tool being Gazebo[11] (P1, P7, P19, P24, S8, S22, S30). In this context, it is important to note that simulation-based experiments can involve the simulation of either (i) abstract fictitious robots, used primarily for running the simulation or (ii) existing hardware platforms, used primarily for carrying out realistic simulations. The simulated hardware platforms are reported in the context of the *Robotic platform* parameter. Despite being based on simulation, the context of such type of experiments can be complex. For instance, a robot in P7 has a task to pick up and deliver materials from several location points, while autonomously exploring a large configuration space and self-configure itself so that the mission is achieved successfully. Other experiments involve also realistic hardware platforms, such as the realistic simulation of a *Ford Hybrid Escape* autonomous car in P21.

Several evaluation studies (8) were conducted by mining software repositories, where the source code of projects developed in ROS or popular ROS packages are analysed and observations regarding state-of-the-art and best practices are extracted. Other solutions are only demonstrated on down-scaled examples as a *proof of concept* (10 primary studies), without a systematic validation. This method hinders the potential applicability and reusability of the proposed solutions, especially in the industrial context. For instance, a group of researchers proposed an initiative for safe package reuse, motivated by the usage of ROS in military applications (P4). However, they demonstrate the abilities of their tool with a comparison of only two packages,

11 http://gazebosim.org

Cartographer and Gmapping (both used for Simultaneous Localization and Mapping - SLAM), which is not generalizable to a much larger number of available ROS packages. Finally, evaluation studies are conducted via *surveys* (3) and interviews (2), with respondents being research practitioners and/or industry professionals.

3.2.4. Recurring challenges and limitations

Fig. 3(c) illustrates the distribution of primary studies according to their future challenges and limitations. The identified challenges and limitations are explicitly stated by the authors themselves. As can be inferred from the table, the set of challenges and limitations is quite diverse.

Researchers most frequently stress *multi-language or multi-platform support* as their future work (22 primary studies). For example, some researchers propose an approach that generates stressful trajectories for a broad range of robots, which lacks integration to other languages used to specify robotic environments (P14). Other primary studies also report about plans to extend their approach to cover ROS2, where the original study only focussed on ROS1 (*e.g.,* P11 and P15).

19 primary studies also lack *further validation*. This is not surprising given that proposed solutions are mostly validated on a single use case or on down-scaled examples (see Section 3.2.3). Fortunately, the acknowledgment of the need for further validation sounds promising for future research on ROS from a software engineering aspect.

15 primary studies *lack of generalization*, either due to the relatively low number of survey respondents (*e.g.,* P13, P16) or a single use-case scenario in the simulation experiment that aims to prove the validity of the proposed solution (*e.g.,* P1, P7, S10, S17). 12 primary studies need extensions for more *complex specifications* (*e.g.,* P29, S22, S23) or require future *automation* (*e.g.,* P6, S09) of a system or its parts that currently need to be performed manually. Other primary studies aim at *improving the performance* (10/63) of their proposed solutions. For instance, a study of this type aims to reduce latency in communication between software and Field Programmable Gate Arrays (FPGAs) (P9). Similarly, message passing in the bridge between ROS and Joint Architecture for Unmanned Systems (JAUS) platform also represents the performance bottleneck (P21).

The less common challenges are *evaluation of quality attributes* (5 primary studies), *enable self-adaptation* (2 primary studies), *runtime support* (2 primary studies), *support for real-time properties* (1 primary study), and *machine learning integration* (1 primary study).

Summary of results – Research aspects (RQ1.1)

▶ More than half of the studies propose methods for addressing specific concerns of ROS-based systems (*e.g.,* techniques based on formal methods, guidelines, testing frameworks), followed by studies focussing on contributions at the architectural level and on model-based approaches. Tools are also proposed as well, whereas only one study contributed with a fully-fledged ROS package.

▶ Researchers tend to focus primarily on proposing solutions in the context of software engineering aspects of ROS, with less emphasis on evaluation/validation research. Reflection papers are very rare, with only 3 occurrences of philosophical studies, and no occurrences of studies reporting opinions or personal experiences.

▶ Among the studies carrying out evaluation/validation research, researchers tend to conduct lab or simulation-based experiments, followed by proofs of concept, deployments in real environments, and mining software repositories; very few studies apply (qualitative) research methods targeting practitioners, such as surveys or interviews.

▶ Supporting multiple languages or multiple robotic platforms is the most reported future challenge/limitation, which is linked to the lack of generalization (also highly reported by researchers); further validation is also another recurrent challenge reported by researchers.

3.3. Used robots (RQ1.2)

Fig. 4 illustrates the distribution of robotic system aspects. As explained in the classification framework (see Table 2, the robots may be classified according to three different parameters: *robotic platform*, *type of robot*, and *cardinality*. Such parameters are better explained in the remainder of this section.

3.3.1. Robotic platform

To gain insight into the robotic platforms that are being used in the research community, we extracted any (possibly simulated) robotic hardware platform in use in the evaluation/validation phases of each primary study. The results of this analysis are presented in Fig. 4(c). An immediate observation is that 29 out of 63 primary studies do not consider any specific hardware configurations, aligning well with the generalized and hardware-independent nature of ROS.

Given that ROS is an open-source initiative, it is designed to be compatible with a wide range of hardware from different manufacturers. This is reflected in the hardware distribution depicted here, with many unique configurations that only find a single use among the primary studies. These range from an electric wheelchair (P3) to advanced robotic systems used in disaster scenarios (P1). The most common hardware considered in the primary studies is part of ready-made systems designed specifically for education and research purposes. This includes the Turtlebot platform[12] and Kobuki[13]. Nevertheless, those hardware enable research on realistic scenarios, such as for navigation stacks.

A total of 8 primary studies use a simulated environment as a digital representation of the physical robotic system (Grieves and Vickers, 2017), with different levels of fidelity. Examples of motivations for using simulated environments include: (i) improved repeatability of experiments (P1), (ii) no access to real hardware (P1), (iii) more efficient data capturing (P1), but also (iv) the evaluation of design alternatives without the risk of causing physical damage to the robot or its environment (P19).

3.3.2. Type of robot

Fig. 4(a) illustrates the distribution of type of robots used in the selected primary studies. The vast majority of the works do not restrict their solution or observations to a particular type of robot. Hence, we decided to extract information about concrete robots that were subjects of the performed validation or evaluation. We follow the classification of Ben-Ari and Mondada (2018), with two fundamentally different types: *fixed* and *mobile* robots. Fixed robots are attached to the stable mount, thus they can compute their position in the environment based on their internal state only. On the contrary, mobile robots must rely on their perception of the environment and they are further classified as *terrestrial* and *airborne*, according to the type of environment in which they operate.

The majority of primary studies use mobile *terrestrial* robots in their research (32/63). Ben-Ari and Mondada classified this

12 https://www.turtlebot.com/

13 https://iclebo-kobuki.readthedocs.io/en/latest/

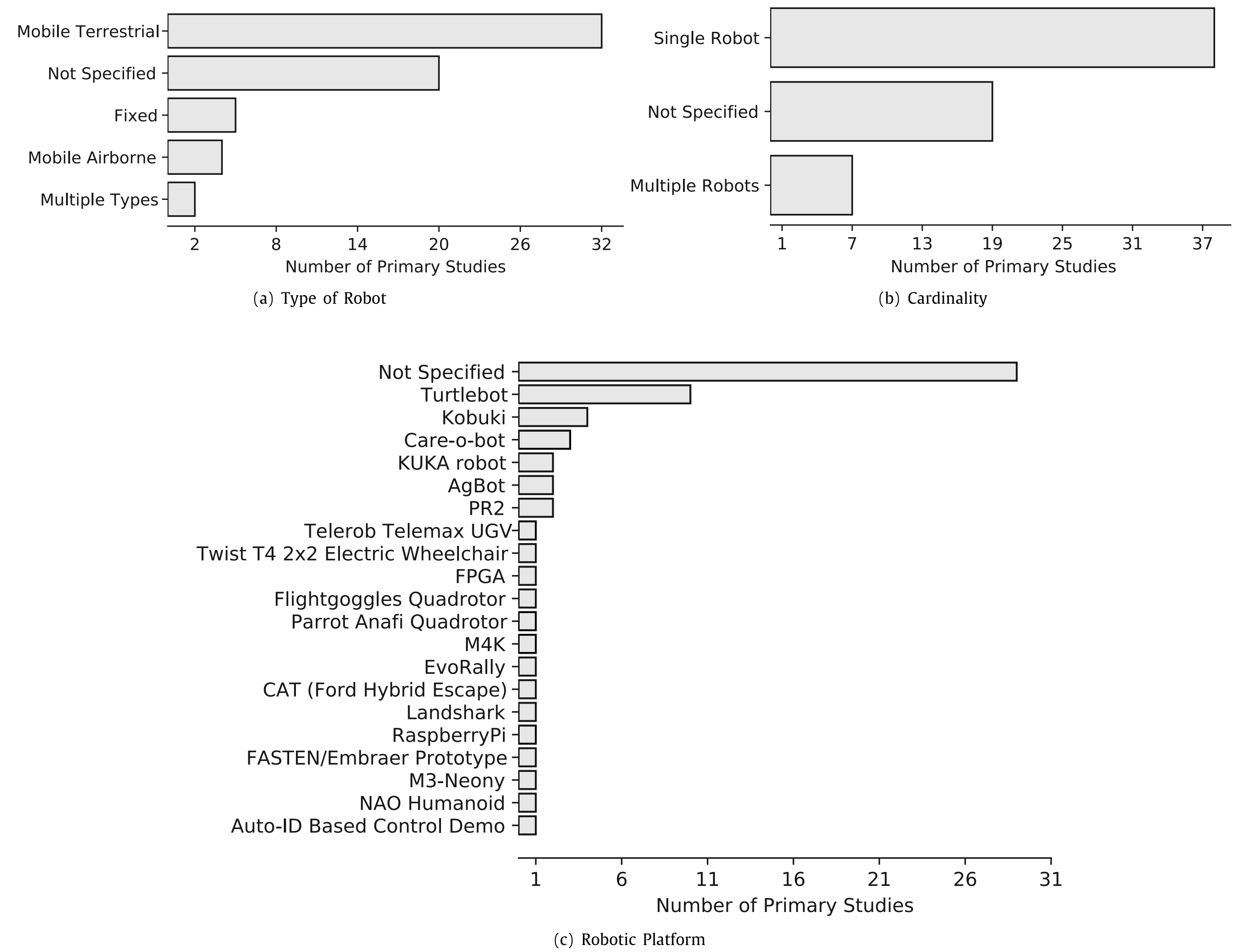


Fig. 4. Used robots mapping.

category further by the mechanism that drives the motion into wheeled and legged robots. Such subjects range from low-cost robots primarily used in education and research to more complex and commercial examples (see Section 3.3.1 for further details). A very notable example used is a search and rescue robot driven by caterpillar mechanism (P1).

Approximately a third of the primary studies do not report what type of robot was used in the research (20/63) or consider multiple types of robots (2/63). These studies mainly include evaluation studies performed via interviews, surveys or by mining software repositories (*e.g.,* P5, P12, P16, S2, S4), which tend to be generic.

Fixed (5 primary studies) and mobile airborne (4 primary studies) robots are the less studied, while none of them were performed on mobile aquatic robots. As an example, a mobile airborne robot is used in a study where a model-driven approach with Eclipse Papyrus[14] is proposed for COMP4DRONE[15] (S7). On the other hand, another primary study used a fixed robot, where the research was conducted on an industrial robotic arm (P2).

3.3.3. Cardinality

We also analyze if the primary study experiments are performed on a *single* robot or *multiple* ones, what represents a significant effort in terms of execution efforts since multiple robots tend to be challenging to coordinate (Sheng et al., 2006). Fig. 4(b) shows that researchers mostly used a single robot as a source of validation or evaluation (38/63), which goes from a toy robotic arm (S3) to an agriculture robot (S27), while multi-robot systems were not frequently explored (7 primary studies). One example of a collaborative multi-robot system consists of two rovers, one is the leader and the other one is the follower receiving instructions from the leader (P18). During our analysis, it also emerged that there are primary studies falling into both categories. For example, the authors of a primary study conducted an experiment on a single Turtlebot, and then conducted another one on unmanned aerial vehicle swarms that collaboratively explore and search a region of interest (P25). Other 19 primary studies do not mention any cardinality, which is again the case of more generic studies or studies without experiments.

Summary of results – Used robots (RQ1.2)
▶ The majority of studies does not specify the used robotic platform (this is primarily due to the general applicability of the proposed solutions). The most used robotic platform is Turtlebot (a widely-known research and education ground robot), followed by a number of various heterogeneous robotic platforms.

[14] https://www.eclipse.org/papyrus/
[15] https://www.comp4drones.eu/

▶ Mobile terrestrial robots are the most used ones, followed by fixed robots (*e.g.,* industrial arms), and flying robots. Only in two cases researchers used multiple types of robots in the same study.
▶ Researchers mostly use a single robot while evaluating/validating their approaches, while multiple robots are seldom used. Several studies do not mention the cardinality of the set of used robots since they are generally applicable to any type of robotic system.

3.4. ROS ecosystem aspects (RQ1.3)

In this section, we present our analysis of the collected data about the specificity of the ROS system considered in the primary studies.

3.4.1. ROS version

There are currently two major versions of ROS: ROS1 and ROS2. The latter is relatively new (officially released in December 2017) and written from scratch due to breaking changes such as cross-platform support (for Ubuntu, Windows, and macOS), multi-robot system support, and real-time support (Hood and Woodall, 2016). Therefore, there are considerable changes in both architectures. Here we analyze to which versions of ROS the selected primary studies restrict themselves.

Fig. 5(a) illustrates the distribution of the ROS version considered in the analyzed primary studies. We see that the majority of the primary studies explore ROS 1 (36 primary studies), while only 8 primary studies consider specifically ROS2. Moreover, 16 primary studies do not restrict the ROS version, including generic studies (*e.g.,* P10, P17, and P24) and studies that consider both versions (*e.g.,* P8, S4, and S6). This is not surprising, given that ROS1 is older and supposedly the most stable version, it is a long-term support (LTS) distribution within the publication time frame of the selected research primary studies (see Section 3.1), and counts on vast public documentation. Moreover, ROS2 seems to be a still open field of research, which may also be either a source of original researches, given its different architecture, or a way to extend the state of the art by comparing ROS1 and ROS2. For instance, S19 results show a considerable performance variation by comparing the same application over both ROS1 and ROS2.

3.4.2. ROS ecosystem level

To further categorize the contribution of current research efforts related to ROS, let us consider which parts of the ROS ecosystem are targeted among the primary studies. The ROS ecosystem is built up of three *levels*[16]:

• The **filesystem** level is related to ROS resources encountered on disk, such as: *packages* and *message types*.

• The **computation graph** level covers the whole peer-to-peer network of ROS processes which process data together. The core components of this level may include: *nodes*, *topics* and *services*.

• The **community** level refers to ROS resources that enable exchange of software and knowledge among communities that use ROS. These resources include: the *ROS Wiki*,[17] *repositories* and *distributions*

Fig. 5(c) graphically show the intersection among the considered levels of the ROS ecosystem. It can be observed that the *filesystem* is in general the most popular ecosystem level, occurring in 48 out of 63 primary studies. It is also the only considered ecosystem level in 23 primary studies. One of the causes of such popularity is the widespread reuse of packages and new methods proposals.

A total of 31 primary studies consider the *computation graph*. This is a common topic among primary studies that present an architecture, which involves inter-node communication (*e.g.,* P3, P7, P21). Alternatively, we observe considerations related to the computation graph in primary studies that describe methods for designing/validating designing such architectures (*e.g.,* P8, P11, S8) or describing the expected behavior of ROS applications (S10).

The *community* is the least common level, present in only 13 primary studies, which enforces the need for studies such as the one we conduct in this primary study. Primary studies related to the community often investigate how ROS is used in practice, with the aim of identifying challenges and best practices when working with ROS (*e.g.,* P8, P12, P16, S12, and S13). Among the community-level primary studies, there are 3 that intersect only with *filesystem*, and 4 with both, *filesystem* and *computation graph*. No primary study at the community level mentions exclusively the *computation graph*.

3.4.3. Communication paradigm

Communication between processes (nodes) in a robotic system is a core functionality of ROS middleware. The two communication paradigms that ROS offers are:

• **Topics**[18] – Nodes can *subscribe* to topics of their interest, which other nodes that generate relevant data can *publish* to. This creates a unidirectional data stream.

• **Services**[19] – A providing node can offer a named service, which can then be called by a client node. This involves sending a request, then awaiting the provider's reply.

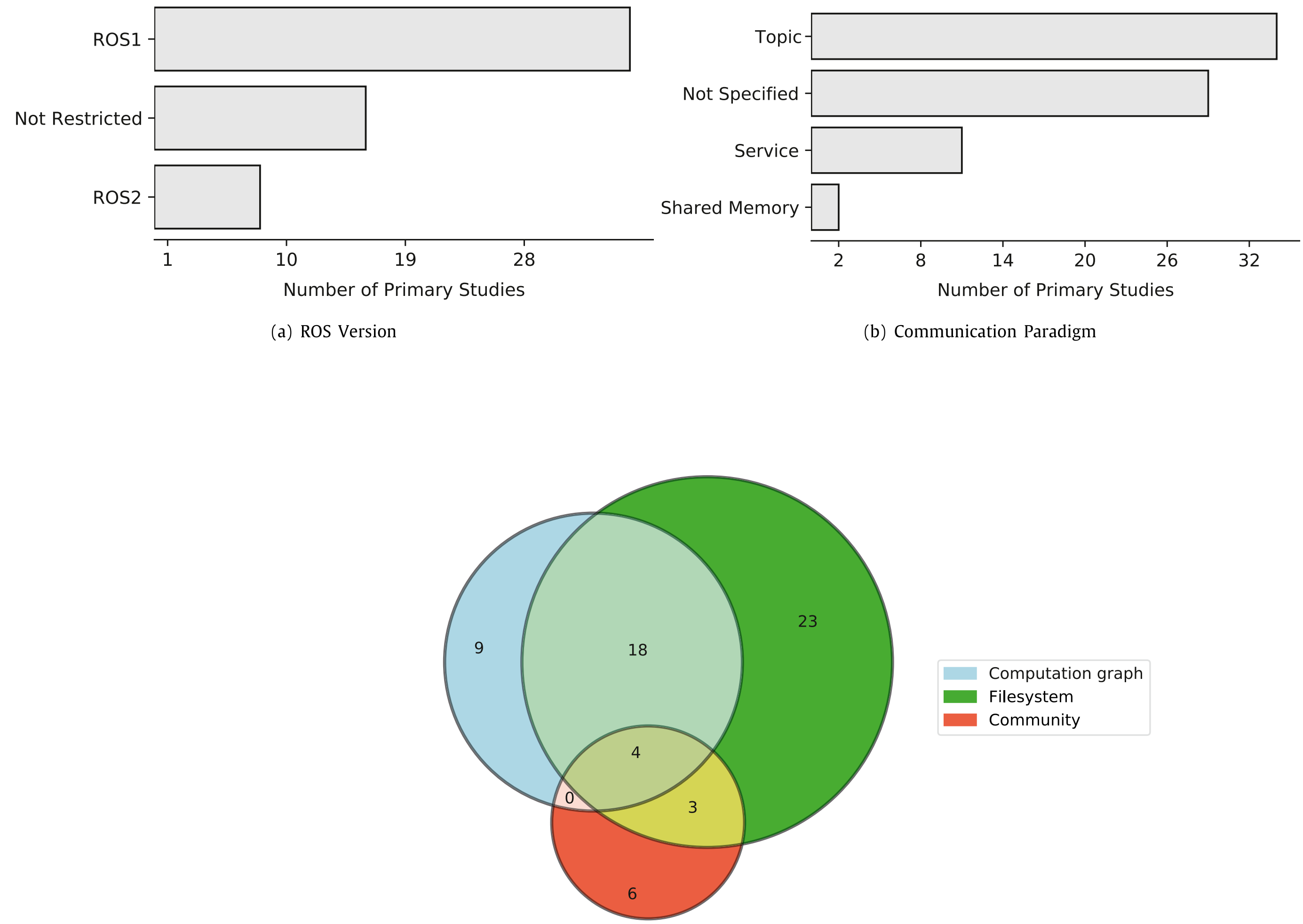
These two paradigms co-exist in the ROS ecosystem because their suitability may differ from one application to another. While topics offer flexible communication for continuous data streams (such as sensor data or state messages), some applications may rely on remote procedure calls. This is common in distributed systems, where services may be an appropriate alternative.

Fig. 5(b) illustrates the distribution of relevant communication paradigms over all the primary studies. There are 29 primary studies that do not disclose details about communication between nodes, which fall in the *not specified* category. It can be observed that *topics* are mentioned as a communication paradigm in most of the cases (34 primary studies), while *services* are only considered in 11 of them and *shared memory* in only 2.

Among the primary studies of this research, it is common to see studies that consider both topics and services as communication paradigms. For instance, a primary study proposes guidelines for architecting ROS topics and services (P8), others consider topics and services in their static analysis of ROS launch files (P11 and P20), while others consider them in their model-based code generator for ROS (*e.g.,* P3). However, while there are primary studies that consider exclusively ROS topics (*e.g.,* for formal model extraction – P6), we did not detect any primary study focussing only on ROS services.

Moreover, a primary study (*i.e.,* P3) also incorporated a novelty method of communication in the proposed architecture, namely the use of *shared memory* between nodes. This is an implementation-specific solution that imposes a smaller computational load compared to conventional communication paradigms in ROS. Along the same lines, another study investigated the performance of using shared memory in systems using ROS2 compared to that of systems using ROS1 (S19).

16 http://wiki.ros.org/ROS/Concepts
17 http://wiki.ros.org/
18 http://wiki.ros.org/Topics
19 http://wiki.ros.org/Services



(a) ROS Version

(b) Communication Paradigm

(c) Ecosystem Level

Fig. 5. ROS System mapping.

Summary of results – ROS ecosystem aspects (RQ1.3)

▶ As expected, the majority of studies on software engineering studies on ROS focusses on ROS1 and very few on ROS2. Interestingly, the results of a non-negligible number of studies are either generally applicable to both ROS1 and ROS2 or explicitly consider both versions.
▶ The majority of studies focus on ROS from at the file system level, with a strong intersection with studies considering also the ROS computation graph. Fewer studies investigate on ROS from a community point of view.
▶ From a communication point of view, software engineering studies primarily consider ROS topics, followed by generic studies that do not make any assumption on the considered ROS communication means. Fewer studies include also ROS services in their research.

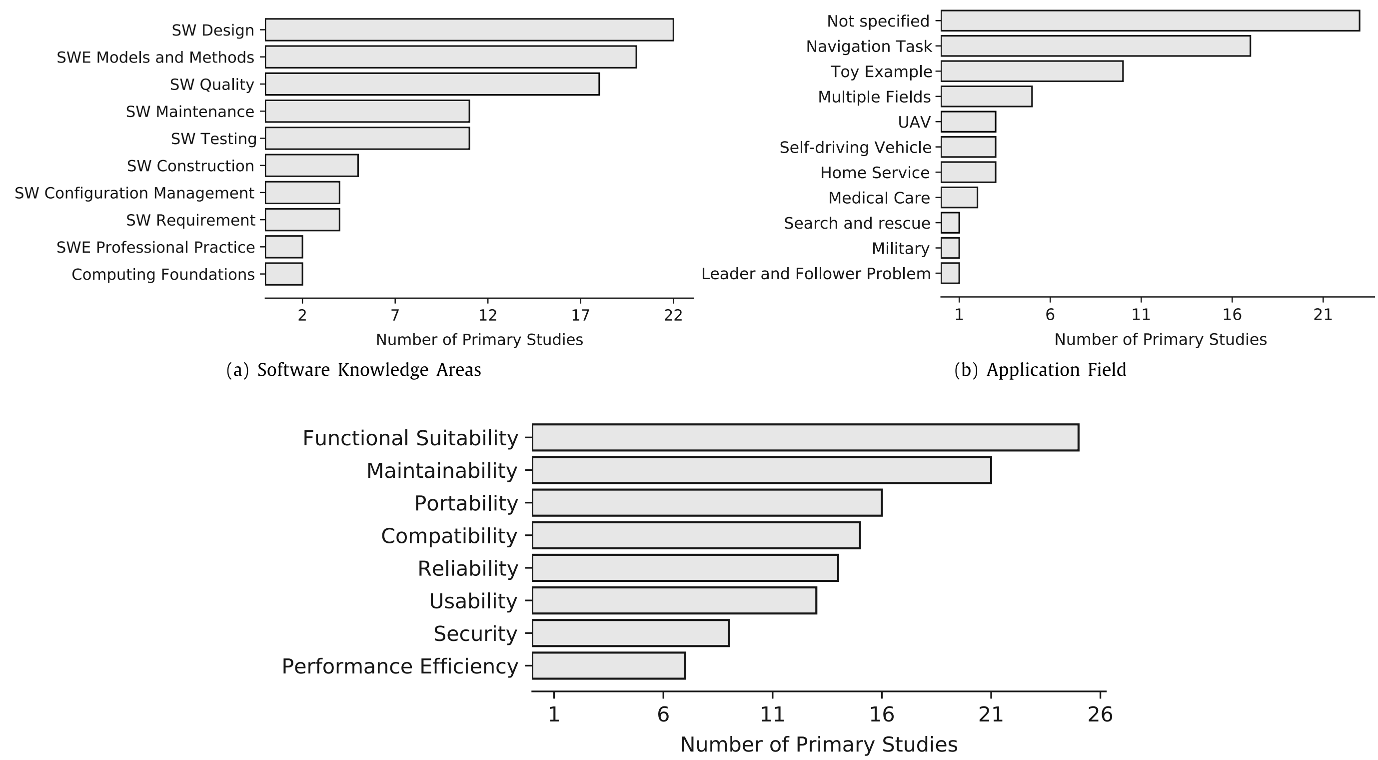
3.5. *Software engineering aspects (RQ1.4)*

Fig. 6 illustrates the distribution of software engineering aspects found in this systematic mapping, which are better discussed in the following subsections.

3.5.1. *Knowledge area*

We classified the primary studies according to the knowledge areas (KAs) defined in Software Engineering Body of Knowledge (SWEBOK) by the IEEE Computer Society (Bourque and Fairley, 2004). The defined KAs are highly correlated and each primary study consists of parts that belong to several areas. Nevertheless, we aimed to perform a classification based on the KAs that the main contributions of a primary study belong to. The complete overview of the mapping is illustrated in Fig. 6(a).

The most dominant KAs within the set of primary studies are *Software Design* (22 primary studies), *Software Engineering Models and Methods* (20 primary studies), and *Software Quality* (18 primary studies). Among others, Software Design includes



(a) Software Knowledge Areas

(b) Application Field

(c) Quality Attributes

Fig. 6. Software Engineering mapping.

designs that aim at satisfying different requirements, such as self-adaptation (P7 and P24), multi-robots integration (S29), and real-time properties (P15). In general, primary studies on Software Quality prescribe tools and techniques that assess system components and their properties against the defined criteria and ensure that the system is of desired quality. Software Engineering Models and Methods include modeling languages and frameworks that facilitate the expression of complex behavior or properties that robotic systems shall satisfy (P3, P7, S7, S26), bridge the gap between different phases of the software life-cycle with automatic code generation (P2, P3, S3) and verify the software design (P3, P6, S27).

Software Maintenance, and *Software Testing* are equally common with 11 occurrences each. *Software Maintenance* attention is directed toward community dynamics and package reuse. As an example, some researchers interviewed ROS developers to understand the issues related to package reuse (S19). Similarly, another research group also investigated Software Maintenance, where an approach that helps identify components that are impacted by changes in the robotics software is proposed (S27). *Software Testing* primary studies are less generic, among others, they propose automatic test-generation tools (P22, S21), code-aware simulations (P10, P18), rapid mission prototyping, and path generation (P14, P18, S28), but also evaluate state-of-the-art with respect to software testing in robotic systems (P5, P23).

Software Configuration Management is in 5 primary studies, while *Software Requirement* and *Software Engineering Professional Practice* KAs occur 4 times each. Software Configuration Management presents primary studies that evaluate and propose techniques that facilitate software configuration prior to deployment (P5, P11), and solutions for run-time self-configuration of ROS-based systems (P7, P24). In a primary study targeting the Software Requirements area, which intersects with Software Quality, researchers provided a tool for verifying non-functional requirements of ROS software (S17). As an example about primary studies on Software Engineering Professional Practice, we mention one that aims at understanding and documenting the practice of robotics software engineering (P23).

Only 3 primary studies fall into *Software Construction* KA, which evaluate state-of-the-art and best practices when developing robotic systems (P23) and explore productivity of development practices, such as live programming (P13). Finally, *Computing Foundations* KA appears with 2 occurrences, proposing algorithms for path generation, and only one primary study explores software engineering development processes that industry professionals and researchers typically adopt when developing new robotic systems (*e.g.,* Agile, Waterfall).

3.5.2. Application field

Fig. 6(b) illustrates a ranking of the tasks that robotic systems perform in the analyzed primary studies. In 23 primary studies the intended task is *not specified*, whereas in 5 other cases *multiple fields* are reported. This is the case, for instance, of studies that list several application fields that their survey and interview respondents are engaged in (P8 and P23). For instance, a primary study reports that industry professionals are mainly engaged in factory automation while the respondents with an academic background mainly do research on service robots (P8).

With 17 occurrences, the most cited field is *navigation task*, where robots need to move around an environment while avoiding obstacles. As shown in Section 3.3.2, mobile robots are the most dominant type of robots exploited in the primary studies hence navigation functionality represents the base for any more complex tasks that such robots may perform. Some primary studies perform experiments where robots only need to visit a set of goal locations (P14, P15, S21), while others perform additional tasks, such as the delivery of materials (P7), surveillance (P19), message delivery (P13, P24) and agricultural tasks (S28).

Proofs of concept with *toy examples* is also popular (10 primary studies). Such examples include a simple set of publisher and subscriber nodes (P9, P22) and a fixed manipulator that grasps a cup from a table (P2). However, such solutions may not be accepted within the research community, where further validation of realistic examples is required.

The other application fields are way less common, namely, *unmanned aerial vehicle* (5/63), *self-driving ground vehicle* (3/63), *medical care* (3/63), *home service* (3/63), *search and rescue* mission (1/63), *military* usage (1/63), and *leader and follower problem* (1/63). Among those, we see fields that depend directly on navigation stacks, such as self-driving ground vehicles, search and rescue, and unmanned aerial vehicles.

3.5.3. Quality attributes

To gain an understanding of which non-functional aspects are focused on in research related to ROS, we derived any considered quality attributes from the primary studies. We make the derivation based on (i) presented design considerations related to a quality requirement (for a proposed solution) or (ii) metrics related to a quality requirement (for a measurement-based experiment). To categorize the considered quality attributes, we make use of the ISO/IEC 25010 standard (ISO/IEC 25010:2011, 2011). The ranked distribution is shown in Fig. 6(c).

It can be observed that 25 primary studies consider *functional suitability* as a quality attribute. Following in prevalence are: *maintainability* (21) and *portability* (16), *reliability* (15), and *compatibility* (14). Portability is directly related with *compatibility*, which together embrace the vision of ROS itself, being an ecosystem that revolves around package reuse and community engagement to encourage collaborative robotics software development. primary studies that consider *reliability* are commonly related to other attributes, where *performance efficiency* (*e.g.,* P1, P4, P7, and P8) and *functional suitability* (*e.g.,* P6, and P9) are the most common.

A total of 13 primary studies also consider *usability*, which commonly intersects with other quality attributes such as *maintainability*, *portability* and *security*. Other 9 primary studies consider *security*, despite the occurrence is also high, and for most of the cases, it also intersects with other attributes, which include *reliability* and *usability*. *security* primary studies lay on different aspects, from military robot projects (P4) to vulnerabilities, and countermeasures in existing systems (S23). *Performance efficiency* is the less frequent aspect, with only 7 occurrences.

Summary of results – Software engineering aspects (RQ1.4)

▸ The analyzed studies cover a wide spectrum of software engineering knowledge areas, including also requirements engineering, testing, maintenance, models and methods, etc. The most considered knowledge areas are: software design and software quality.

▸ About quality attributes, the most frequently investigated is functional suitability, followed by maintainability. Other quality attributes such as portability, reliability, usability, and security are also considered. Currently, performance efficiency is the least considered quality attribute.

▸ Many of the analyzed studies is not specific to any application field, further confirming the general applicability of several research contribution in the area of software engineering for ROS. Navigation tasks emerged as the most recurrent application field, followed by a number of (non-specific) toy examples. Other application fields considered in the analyzed studies include: UAVs, self-driving vehicles, home and medical care, etc..

4. Potential for industrial adoption (RQ2)

To evaluate the *rigor* and *industrial relevance* of the analyzed primary studies, we categorized them according to the classification proposed by Ivarsson and Gorschek (2011).

Ivarsson and Gorschek (2011) define a scoring rubric for assessing the *rigor* of a technology evaluation as shown in Table 5. In essence, rigor refers to how *exact* and *precise* an evaluation is performed, as well as how it is reported. This is assessed based on the described *context* and *study design*. Furthermore, the *validity* plays a role in assessing rigor: exposing potential threats to (internal, external, and construct) validity and presenting measures of how these have been mitigated in a primary study allows the reader to establish whether the appropriate scientific rigor has been applied. While reading each primary study, we performed an assessment of the study's rigor according to the rubric and registered its score across all three components. Each can have a value of low (0), medium (0.5), or high (1), meaning that the maximum total score for a study's rigor is 3.

In the work of Ivarsson and Gorschek (2011), the industrial relevance score metric is presented as means to abstract the view of the state of the technology evaluation via a numeric score. This model aims to assess the realism of the environment in which the results were obtained, *i.e.,* the *Research method* applied, but also the realism of three different aspects of the performed validation or evaluation, namely, *Subjects*, *Context* and *Scale*. Each of the four aspects contributes to the overall industrial score between 0 and 1. The scoring rubric used in this study (an adaptation of the original) is presented in Table 6. It is important to note that in this study we look at the robotics industry as a whole (Macenski et al., 2022), without restricting ourselves to any specific area/application domain (*e.g.,* mobile robots, healthcare, home robots, autonomous vehicles, etc.). We consider the robotics industry as the ecosystem of companies, open-source contributors, and roboticists in general who are deploying and using robotics software in real/production environments (*i.e.,* no toy, demo, or didactic projects) (Malavolta et al., 2021).

For what concerns **rigor**, Figure Fig. 8(a) shows that more than half of the primary studies 35/63 has a low-medium level of rigor, with scores between 1.0 and 1.5. Nevertheless, we can also notice that 10 primary studies score very high in terms of rigor, with a maximum score of 3.0. When looking at the detailed scores in the lower part of the figure we can notice that researchers tend to describe well and in a complete manner the *Context* in which their evaluation is carried out, such as in terms of the deployment environment, problem statement, current practices in place, etc. For example, the authors of S28 first present a concrete example for motivating their study, then they give an overview of the main intuition behind their proposed approach for rate impact analysis for ROS-based systems, including snippets of source code, and preliminary data about the expected benefits of their proposed approach. The situation is quite similar when looking at the *Study Design* metric, where primary studies tend to have medium and high scores. This result does not come as a surprise since it is expected that the results of an evaluation are generally preceded by a description of its main research questions, dependent/independent variables, the details of the implementation of the measured system (or the parameters of its simulation), etc.

As a representative example, the authors of P16 present a study on the ROS ecosystem in terms of ROS packages reuse and community dynamics; here the main research questions of the study are presented right from the introduction and it is explained how those research questions emerged from a preliminary round of interviews targeting 19 ROS developers. Subsequently, the authors provide detailed information about how the participants of each part of the study (interviews, focus groups,

Table 5
Rigor scoring rubric (adaptation) (Ivarsson and Gorschek, 2011).

Component	Description
Context	Is the context described to the degree where a reader can understand and compare it to another context (e.g. in terms of deployment environment, type of robotic system and problem statement)?
Study design	Is the study design described to the degree where a reader can understand its main parts (e.g. in terms of evaluation method, implementation platform, variables, and treatments)?
Validity	Are the validity and threats of the study discussed and measured in details (e.g. in terms of internal validity, external validity, and conclusion validity)?

Table 6
Industrial relevance scoring rubric (adaptation) (Ivarsson and Gorschek, 2011).

Component	Description
Subjects	Are the subjects of the study representative (e.g. real robots, industry professionals as survey respondents or interviewees, real project repositories)?
Context	Is the study performed in a setting representative of the intended usage (e.g, robots perform tasks that are representative of industrial use-cases)?
Scale	Is the study performed using applications of realistic size (e.g. size of the operational environment, number of robots used, number of survey respondents or interviewees, number of mined repositories)
Research method	Does the research method facilitate investigation of real situations that are relevant for practitioners (e.g. real deployment, interview, descriptive/exploratory survey)?

online survey) have been selected and targeted, followed by a detailed overview of how the obtained data has been grouped and analyzed (both quantitatively and qualitatively). The main pain point in terms of rigor is *Validity*, where the majority of primary study score low (46 primary studies), meaning that those studies do not present *at all* the main limitations and threat to validity of their performed evaluations. The lack of a clear and transparent elaboration of the threats to validity of a technological evaluation is fundamental for enabling its independent verification, reproduction, and replication (Wohlin et al., 2012). It is strongly advised to the authors of evaluations in the context of software engineering for ROS to invest time in studying and clearly reporting such threats related to their evaluations in order to make our research field more robust. As a starting point, we suggest ROS researchers to study the classification of threats to validity proposed by Cook and Campbell Hyman (1982)Wohlin et al. (2012, Chapter 8.8) and to thoroughly report the external, internal, construct, and conclusion threats to validity of their performed evaluations. For the interested reader, examples of primary studies already adhering to such classification of threats to validity are: P8, P13, P23, S12, and S13.

For what concerns **industrial relevance**, as can be inferred from the cumulative score in the upper part of Figure Fig. 8(b), the industrial relevance of the majority of the primary studies is relatively low (47 primary studies), with scores lower or equal to 2. If we zoom into the lower part of the figure, which illustrates each of the four aspects that contribute to the overall industrial score, we see that around half of the studies score well on the *Subject* and *Context* criteria (with score 1). Robots used in those experiments perform tasks that are well representative (37 primary studies) and studied in real-life use-case scenarios (36 primary studies). Even though the tasks and the setting may be representative of those primary studies, they are usually performed in simulations, not on all real robots (*e.g.,* P1, P8, P18, P21, P24, P25), survey/interview respondents are mostly of researchers rather than industry professionals (*e.g.,* P8, P13, P16), or the set of mined repositories may contain student and research projects (*e.g.,* P8, P12). Moreover, experiments in 46 primary studies are not conducted on a large-enough *Scale*, and they scored 0 in terms of *Research method* in 50 cases. While deploying real robotics systems in a context representative of their intended usage may be costly (*e.g.,* unmanned aerial vehicles for military missions), or even not feasible in some cases, simulation-based experiment results cannot be considered as industrially-relevant since they generally do not take into full account the various uncertainties of real deployment. However, advances in the simulations field are emerging thus there are no obstacles to at validate the solution in the representative, but simulated environment and on a much larger scale.

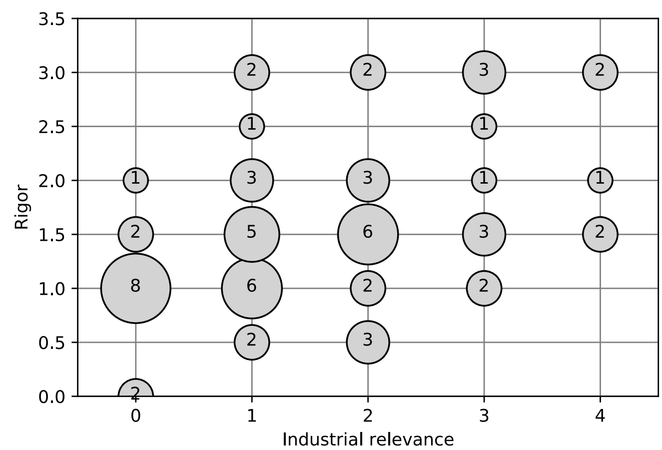
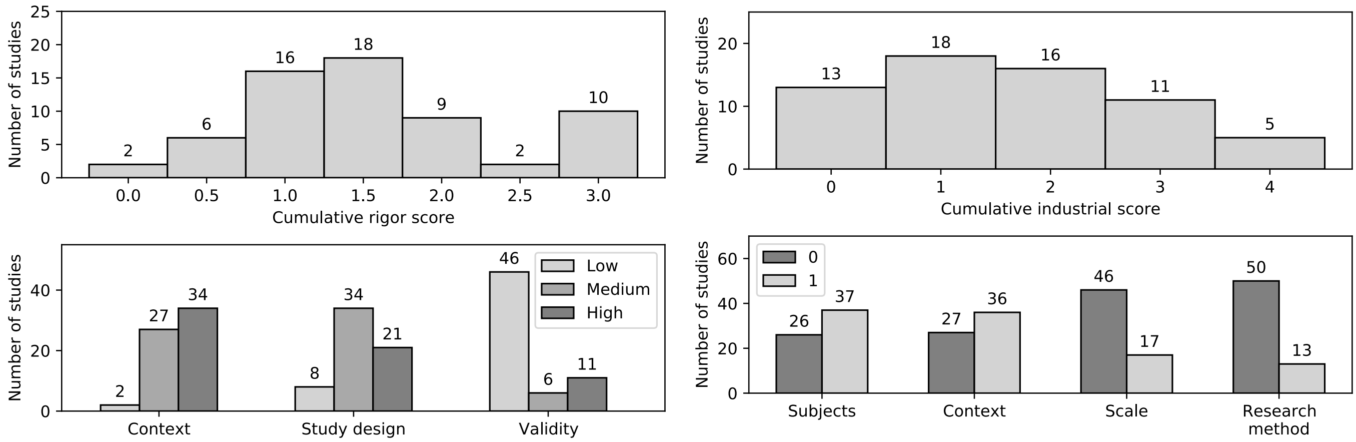


Fig. 7. Rigor and industrial relevance (evaluated as proposed by Ivarsson and Gorschek, higher is better) Ivarsson and Gorschek (2011).

Fig. 7 shows the aggregated results for the rigor and industrial relevance of the analysed primary studies, whereas Figures Fig. 8(a) and Fig. 8(b) show the detailed scores for each individual metric. A total of 36 primary studies lay in the bottom-left quadrant, with a modest rigor (from 0-1.5) and industrial relevance (from 0-2), where 2 of them are not significant neither for rigor nor industrial relevance. Other 6 primary studies lay in the top-left corner of that quadrant, representing an intermediate rigor and industrial relevance. An example of such primary study is the one where the author proposes a model-driven code generation approach for an industrial study case (S10). That primary study's research rigor score is set considering that it delivers a good experiment design description, and statistical analysis of the results; however, it does not thoroughly report its threats to validity. The industrial relevance of S10 is relatively low since its evaluation is based on a simulated-based experiment, despite the case study being representative of real-world robots. Despite being less frequent, in the top-right quadrant (rigor from 2 to



(a) Rigor.

(b) Industrial relevance.

Fig. 8. Rigor and industrial relevance of the primary studies.

3 and industrial relevance from 2 to 4) we also see 13 primary studies where the rigor and the industrial relevance are high. A good example of those primary studies is the one that has the highest rigor (equal to 3), and an industrial relevance equal to 3, only missing a good context since the number of considered robots is not representative of an operational environment (P8).

Finally, in order to support researchers willing to improve the rigor and industrial relevance of their technological evaluations, in the following we describe the main characteristics of the **top-performing primary studies in terms of both rigor and industrial relevance**. By referring to the top-right bubble in Fig. 7, our dataset contains two of such top-performing primary studies, namely P23 and S13:

P23 (García et al., 2020) – This primary study presents a large-scale empirical investigation on the state of the art and practice of robotics software engineering. Specifically, by interviewing industrial robotics experts and conducting a subsequent online survey, the authors collected data about (i) the practices related to robotics software development, in terms of typical activities, development paradigms, development processes, software languages and frameworks, quality assurance, reuse practices, and used tools, (ii) the "distinguishing characteristics" of robotics software engineering compared to other cyber-physical domains, such as avionics and automotive, and (iii) the main challenges and solutions faces and applied by robotics software practitioners.

The *rigor* of this primary study is high since (i) the context of the study is carefully described, with concrete examples, the goal and research questions are clear from the beginning, and it provides direct references to the target audience who might benefit from the answers to the identified research questions; moreover, the authors explicitly clarified that the scope of the study is on the service robotics domain only; (ii) the study design is carefully presented, including the most relevant details about the design and conduction of every single phase of the study (*i.e.*, the interviews and the survey); the structure and contents of the interviews and the online questionnaire are included in the online material as well; (iii) limitations of the study and its threats to validity are discussed explicitly, and raw data is available in the replication package of the study for independent verification.

The *industrial relevance* of this primary study is high since (i) the subjects of the interview are representative of the population of robotics engineers: interviewees were selected following a convenience sampling among the industrial collaborators of the authors, who have been involved in several robotics projects, whereas the survey has been distributed via several channels, including social media, GitHub repositories, and indirect referrals; (ii) the context/setting of the study is clearly described and it is representative of the intended setting since the interviewed and surveyed experts are both the source from which information is extracted and the intended audience of the study; (iii) with 18 interviewees and 159 survey participants distributed among 58 companies (from 20 different countries) and 16 academic institutions (from 10 different countries), which is representative; (iv) the research method facilitates the investigation of real situations since the authors obtained their results directly from the analyzed data coming from experts in the field and contributors to open-source projects; also, the authors provide a set of research themes and actions for researchers willing to have an impact in the field of software engineering for service robots (*e.g.*, to work on techniques for providing realistic test data, to improve the capabilities of simulations, to exploit and reuse reference architectures).

S13 (Malavolta et al., 2021) – This primary study presents an observational study aiming at characterizing how ROS-based systems are architected in the context of real-world open-source ROS projects. The study consists of two main parts: (i) mining 335 publicly-available repositories from GitHub, GitLab, and BitBucket to elicit which quality attributes are targeted when reasoning on the software architecture of ROS-based projects (*e.g.*, maintainability, security, performance), how ROS developers document the architecture of their systems, and a set of 47 guidelines for architecting ROS-based systems, and (ii) surveying 119 ROS developers who actively contributed to the mined repositories in order to validate the extracted architecting guideline, as well as identifying additional ones.

The *rigor* of this primary study is high since (i) the context is described in great detail in terms of targeted Git repositories, target audience (both in terms of research results and survey participants), and the characteristics of the analyzed systems (*e.g.*, their application domain, size, age, number of contributors); (ii) the study design is described precisely, with a clear indication of the goal, research questions, collected data points, and applied data extraction and analysis procedures; (iii) the threats to validity of the study are elaborated extensively according to the Cook and Campbell categorization (Wohlin et al., 2012) and a full replication package is available for independent verification of each phase of the study.

The *industrial relevance* of the study is high since (i) the analyzed Git repositories (*i.e.*, the subjects of the study) are highly heterogeneous in terms of number of contributors, number of commits, type of robot, etc. Also, the authors performed a strict selection process when building the dataset of repositories, filtering out non-representative projects (*e.g.*, demo or didactic repositories); (ii) the study is performed in a representative context since the data sources from which the guidelines are extracted correspond to the main target audience of the study (*i.e.,* "roboticists who want to apply good design principles to develop robots that meet quality requirements" (Malavolta et al., 2021)); (iii) with a qualitative analysis of 335 individual ROS-based projects, the scale of the study is in line with software engineering literature; (iv) the research method facilitates the investigation of real situations since the data and the guidelines are extracted from real ROS-based systems used in real deployments; moreover, each of the extracted guidelines underwent the scrutiny of 119 roboticists who directly contributed to the 335 repositories.

Summary of results – Potential for industrial adoption (RQ2)

▸ More than half of the analyzed studies have a low-medium level of rigor, with the most dramatic score related to the quality of reporting limitations and threats to validity of the conducted studies. The context and design of the conducted studies tend to be of good quality.
▸ The industrial relevance of the analyzed studies is relatively low, with several studies lacking in terms of scale and the representativeness of the applied research methods of real/industrial situations (*i.e.,* mostly lab or highly-controlled experiments involving small-scale robotic systems, instead of real deployments in the wild). We did not identify notable issues in terms of the number of involved subjects and the description of the context of the performed evaluation.

5. Discussion on academy-industry alignment

The goal of this systematic mapping study is to investigate the state of the art in software engineering research on ROS. In this section, we put in context the results emerging from this study and present the main implications for both researchers and practitioners, with suggested actions for a better alignment between academia and industry.

Academy-industry alignment

Action 1 – Researchers shall keep studying, applying, and evaluating their solutions in the context of *mobile robots*.

Fixed robots currently represent the majority of the market demands and they are mostly used by manufacturers for internal logistics in repetitive delivery processes and similar tasks (Anon, 2020). However, predictions state that this will change by the end of this decade and many companies already started to exploit the flexibility of mobile robots since they do not require external infrastructure to localize themselves and they can perform sophisticated tasks without manual operation or observation (Anon, 2019). As our results in Section 3.3.2 show, mobile robots are predominately used in research papers as well, so it is encouraging that advancements in both research and industry are heading in the same direction.

Academy-industry alignment

Action 2 – Researchers shall investigate more the software engineering aspects of robotic systems involving *multiple robots*.

There are many evolving paradigms and visions for the future of robotics, such as the *Fourth Industrial Revolution*, *Precision Agriculture* and *Smart Cities*, that would undoubtedly require large, collaborative multi-robot systems, also called *cobots* (Gong et al., 2019; Hrabia, 2016). Such systems are specially designed for direct interaction with humans within the specified collaboration workplace (Anon, 2020b). The market opportunities in the collaborative robots domain are very attractive and they are predicted to grow rapidly in the upcoming five years (Anon, 2020a). However, several restraints currently prevent their growth on the market, namely cyber-security threats, due to the incorporation of IoT, and lack of awareness about organizational networks (Anon, 2020b). These challenges are yet to be resolved in the software engineering/robotics research communities, since our results in Section 3.3.3 show that primary studies do not even tackle multi-robot systems to a large extent.

Academy-industry alignment

Action 3 – Researchers shall focus their attention to *ROS2* and evaluate their solutions on ROS2 systems.

As can be inferred from the results in Section 3.4.1, ROS2 is currently still under-explored in the research community. Nevertheless, and more importantly, some primary studies observe that ROS2 is also not that widely used in the industry either and there seems to be no initiative to change that (P8, P23). This is rather ironic given that one of the main goals of ROS2 is exactly to fit better into the industrial and commercial solutions (Macenski, 2020; Macenski et al., 2022). This is a good insight for researchers since there is a clear need to investigate the potential of ROS2, not just in the context of a vision for the future, yet with sound validation as well. Most importantly, there will be no long-term support for ROS1 after 2025 and all future releases will be based on ROS2, so the time to transition is running up (Anon, 2021).

Academy-industry alignment

Action 4 – Researchers shall better study how *Model-Driven Engineering* techniques can be successfully integrated in the context of real robotic projects (*e.g.,* as a complement to already-in-place practices).

We observed a large discrepancy between research and industry with respect to Model-Driven Engineering approaches. Several primary studies point out that it is very rarely used in industry (*e.g.,* P20, P23), yet the results presented in Section 3.5.1 show that most contributions in research are stemming from MDE. Only one primary study acknowledges this issue and offers a model-driven solution as a complement rather than a must (P20). A possible reason for such poor adoption by industry professionals is that practitioners need to get familiar with the syntax and semantics of the used modeling language and tools before using them. Other reasons for the current lack of adoption of

MDE approaches in the robotics industry include: lack of tools that provide functionalities on par to current IDEs (*e.g.,* IntelliJ Idea) and scalability with respect to both sizes of models and diversity of involved artifacts (Bucchiarone et al., 2020). The state-of-practice also shows that MDE is applied on key parts of the system rather than to a system as a whole (Whittle et al., 2013). While there may be several benefits of the proposed model-driven solutions, they are still another additional skill within the already-large skills set required from ROS developers. This is a good point for future investigations and valuable insight for researchers, which shall consider how the complexity of MDE techniques can be abstracted out and applied without needing to expose practitioners to all the technical details of MDE.

Academy-industry alignment

Action 5 – Researchers and practitioners shall improve the *simulation of ROS systems* and better integrate it into current development practices).

Simulated robots are commonly used to evaluate/validate new robotic software or compare different robots without additional investment or damaging the hardware. However, in a primary study, interviewees and respondents claim that the transition from simulated to the real-world robots represents a major challenge (P23). This is mainly because some phenomena are difficult to simulate in the current simulation tools (*e.g.,* physics, sensors data) or simply difficult to model (*e.g.,* experiments involving robot-human interactions). For all the aforementioned reasons, practitioners rely on experimentation as much, or even more than simulation. There is an emerging need for improvement of robotics simulation tools and their capabilities so that the transition process between the simulation and the real world can be eased. During our analysis an interesting case emerged, P14, where the authors firstly use simulated *Flight Goggles* to evaluate different versions of the software controllers, and then they use *both* simulation and real deployment to evaluate the proprietary control software of the *Parrot Anafi* drone. A possible explanation for using such a combination of configurations is that the *Parrot Anafi* was the physical hardware available for the experiments and, as the authors themselves state in the primary study, the *Flight Goggles* has rich capabilities available for simulation, allowing them to evaluate different configurations of their control software with relatively low effort. This is a clear example of the concrete advantages provided by a stable and mature simulation platform for robotics software. Overall, the fact that there are a few primary studies that propose code-aware simulation enhancements (*e.g.,* P10) and interactive visualization of robot missions (*e.g.,* P17) is promising for future advancements of both research and practice, but improvements in this area still need to be tackled to a greater extent in future research.

Academy-industry alignment

Action 6 – Researchers shall study more in depth the community of *ROS practitioners* and its dynamics (*e.g.,* by analyzing public ROS packages, their source code, their contributors, interviewing practitioners, etc.).

The primary studies considered in this review illustrate the widespread usage of ROS across various application fields and hardware types. The vast majority of primary studies propose a method (35) or an architecture (13) for working with ROS as their main contribution. This is indicative of excellent (re)usability of existing ROS packages for new projects and research efforts in robotics, despite only 7 tools being proposed among the primary studies. Also interestingly, only 1 of the primary studies resulted in a *package* as a direct contribution to the ROS ecosystem. Interestingly though, the practice of using existing functionality is representative of the ROS community as a whole, a primary study (P12) states that 80% of dependencies in ROS projects rely on packages being developed and maintained by a small number of (non-academic) *foundational working groups*: collectives organized around a shared area of expertise to provide the fundamental building blocks required in the construction of most robots (Estefo et al., 2019). Despite the results pointing out the limited focus of primary studies on the ROS *community* level, this paper reveals existing research that identifies interesting community dynamics and development bottlenecks, leading to the establishment of guidelines and best practices when working with ROS. This is a potential benefit of further research into the ROS community level. Moreover, it is interesting to note that only 5 primary studies performed surveys or interviews targeting ROS practitioners; we believe that this is a missed opportunity for researchers, who have the chance to directly talk to their target audience, to elicit their recurrent or more prominent technical problems, and, more in general, to have a better understanding of the development practices of ROS-based systems. Also, performing surveys and interviews allows researchers to establish a direct communication channel with ROS practitioners, which can be instrumental for establishing partnerships, forming consortia for research grants, and to directly informing the industry about the latest research results.

Academy-industry alignment

Action 7 – Practitioners are invited to stay up to date with respect to the latest research efforts about software engineering aspects on ROS.

While not all contributions to the ROS ecosystem originate from academic research, we believe that practitioners using (and contributing to) ROS for developing robotics software can benefit from the presented results to gain an understanding of current topics in academia. Moreover, we identify that researchers commonly publish primary studies in Software Engineering on ROS at Software Engineering venues (37 primary studies), and that the number of software-related venues is broader than Robotics ones (see Table 4). With the snowballing process, we also reveal further venues in Software Engineering (such as MSR) and Robotics (such as IRC), which were not included in the preliminary list. The list of venues we provide and the number of primary studies in each of them is a good starting point for practitioners to stay up to date with respect to the latest research efforts in the area of software engineering for ROS-based systems. The list of publication venues in Table 4 is a good starting point also for researchers who can use it for following existing interests/trends in academia as well as for considering them as publication targets.

Academy-industry alignment

Action 8 – Researchers shall focus more on *non-functional aspects of ROS systems*, especially on performance and security.

As shown in Fig. 6(c), 25 primary studies are focussing on the functional suitability of the analyzed system. Those studies focus primarily on *what* the system does (*i.e.,* its functionalities), rather than on *how* the systems operates, *e.g.,* in terms of reliability, maintainability, etc.. It strikes the eye that very few of the primary studies target its security (only 9 occurrences) and its performance (only 6 occurrences). Those are key aspects of today's robotic systems and we invite both software engineering and robotics researchers to focus more their efforts on those quality aspects of ROS systems. For example, if we think about the robotic applications that have the highest potential today, such as self-driving vehicles, precision agriculture, or healthcare, it goes without saying that those systems must have the highest level of security and their response times and scalability must be optimal and/or (at the very least) predictable.

Academy-industry alignment

Action 9 – Researchers shall take the initiative towards making the technological evaluations of their proposed solutions more easily *transferable to the industry*.

As shown in Fig. 7, according to our evaluation of rigor and industrial relevance of the primary studies, the potential for industrial adoption of solutions presented in current research on ROS is low-medium. Considering this state of the research field (as outlined in more detail in Section 4) we propose the following actions for improvement:

- Actions for improving the **rigor** of the technological evaluations of research solutions (representing an upwards trend in Fig. 7):
  - Identify threats to *validity* as part of the paper, reasoning about which approaches have been taken to mitigate these. This practice is relatively uncommon among the primary studies, applied in 18 out of the 62 primary studies, among which we find 6 that only superficially discuss the threats.
  - Establish a complete overview of the context and design of the paper by describing the specific target context (or generality thereof) in terms of application field and robot type, and setting up a clear experimental setup including hardware used, variables, and treatments.
  - Provide either a *replication package* for experimental evaluations or a detailed study design section.
- Actions for improving the **industrial relevance** of the technological evaluations of research solutions (representing a rightward trend in Fig. 7):
  - Evaluate a proposed solution on *real robots* which are also used in industry (rather than robots made for education).
  - Provide a rationale (and evidence) of the *scalability* of the proposed solution.
  - Consider the *industrial context* (including requirements for safety of the proposed solution).

In Section 4 we describe three primary studies with high rigor and industrial relevance; these can serve as concrete examples of how the actions listed above can be put in place by researchers.

6. Threats to validity

In this section we identify and discuss potential threats to validity of this study, as well as the actions taken to mitigate these.

6.1. Internal validity

To mitigate the influence of potential external influences when conducting the research, we followed a predefined research protocol based on established guidelines for systematic mapping studies (Petersen et al., 2015; Kitchenham and Brereton, 2013; Wohlin et al., 2012). Potential external influences that caused changes to the research protocol were discussed by all researchers and mitigated accordingly. More specifically, given our focus on *software engineering* studies on ROS, initially our search string for the DBLP query was including the `software` keyword connected to the `ROS` and `robot` via a logical *AND* operator; however, the presence of the `software` keyword in the query was reducing the number of potentially-relevant studies to just a few tens, so we decided to remove it from the query in order to have a wider (and potentially more representative) set of primary studies. Moreover, initially we considered only software engineering venues when mining Google Scholar, but in a subsequent iteration of this research we included a search in robotic venues as well in order to include additional studies published in robotic venues. Finally, another potential source of external influences is due to the fact that some publication venues might be more attractive for specific sub-populations of researchers in software engineering for ROS-based systems, potentially leading to skewed results during our data analysis phase; we mitigated this potential source of bias by (i) complementing our automatic search on selected venues (stage 1 in our search and selection process) with backward/forward snowballing (stage 4) and (ii) analyzing each primary study independently of the scientific venue where it is published.

6.2. External validity

A prevalent external threat to validity for this study is that the selected set of studies may not be representative of the state-of-the-art in software engineering research on ROS. To mitigate this threat, we established a list of top venues in software engineering and robotics based on rankings from both Google Scholar and CORE as described in Section 2.2. Our search and selection process starts from studies published at venues in the list, hence we are confident that the obtained results do reflect the state-of-the-art. Since Google Scholar and CORE partially overlap and complement each other, combining both sources mitigates the risks associated with the usage of a single source of truth. Furthermore, we used an automated script to obtain the initial list of studies to mitigate potential bias or human error.

If the list of top venues was expanded to a more general context rather than only software engineering, our resulting findings may have turned out different. However, the decision for limiting this scope is well founded, since software engineering research of ROS is the particular topic of this systematic mapping study.

The set of inclusion and exclusion criteria (see Section 2.2, Stage 3), was rigorously defined and agreed upon by all researchers involved prior to the analysis, such that no potentially relevant study is excluded. Most notably, we excluded studies which simply state that ROS was used for implementation, without disclosing further implementation details. Such studies are excluded since they usually focus more on robotics in general rather than related software engineering aspects and, as such, they are not relevant for our study. While there may be some potentially relevant studies that are not written in English, this is still the most widely used language in publications.

6.3. Construct validity

The initial search string (see Section 2.2, Stage 2) was deliberately defined as general as possible so to mitigate the risk of not including potentially relevant studies.

Furthermore, the assessment against the defined inclusion and exclusion criteria (see Section 2.2, Stage 3) was performed independently and in parallel by the two researchers. When comparing the assessments, disagreements were resolved by the third researcher acting as arbiter. This procedure mitigated personal bias.

While assessing the rigor and industrial relevance scores of the selected studies we used the rubric proposed by Ivarsson and Gorschek (2011). We are aware that this rubric is restricted to the information reported by the authors in their primary studies; the score could have been different if additional information about each primary study was considered, *e.g.,* the presence of detailed documentation, the presence of a test suite with sufficient coverage, the involvement of an active community. Overall, while rigor and industrial relevance score metrics are widely used and thus represent a good indicator of potential for industrial adoption, there are a number of other parameters that could complement them and, together with research strategies and methods, strengthen the obtained results. Additional parameters (*e.g.,* industry involvement of the authors) were not included due to time limitations, but they present a solid basis for future improvements of this work.

Finally, we are also aware that the ROS industrial ecosystem is highly heterogeneous and involves very different systems in terms of, *e.g.,* safety, security, and performance requirements, application domain, size, longevity, and required technical backgrounds. In line with our study design, our analysis of the industrial relevance of the 62 primary studies aims at (i) providing a *general overview* of the current landscape in terms of transferability of the performed evaluations from research to practice and (ii) giving concrete examples and widely-applicable insights for researchers for making their evaluations more industrially relevant (see Section 5).

6.4. Conclusion validity

The classification framework (see Section 2.3) was defined by two researchers together, upon a thorough analysis of all studies performed by both researchers independently. This ensures that the parameters defined in the framework are representative with respect to both the information presented in the studies and to the research questions defined prior to this stage. The resulting classification framework was reviewed by the third researcher to verify suitability to answer the research questions and mitigate any potential bias.

In the data extraction stage, we equally divided the columns of the classification framework between two researchers. This could potentially impose threats to the conclusions drawn from the extracted data. The final results are reviewed by the third researcher such that this threat is mitigated to an extent. Additionally, rigor and industrial relevance score data extraction (see Section 4) was performed by the two researchers separately so that potential conclusion threats with respect to this two parameters can be mitigated. For instance, it should not be the case that the study with relatively low rigor scores well on the industrial relevance since the latter is inferred based on the information presented in the research study itself.

7. Related work

In this section, we briefly present the studies that are the most similar to this one, either in terms of context or goals.

Santos and Petrillo present the work that resembles this study the most in terms of goals (Santos and Petrillo, 2021). They identify, classify and evaluate the current state-of-the-art software engineering for robotic systems. While they classified the primary studies by the SWEBOK knowledge areas mentioned, we focused only on the areas where the main contributions of a study belong to. Nevertheless, the four most recurring areas in our study are exactly the same as those identified in their work. While SWEBOK areas are their only focus, they are just one of the many parameters analyzed in our study.

Santos et al. performed a systematic mapping on software engineering for robotic systems, from a software quality perspective (dos Santos et al., 2020). They systematically analyzed software quality attributes considered in the robotic systems. As their results show, security is one of the less investigated software quality aspects, hence they present an overview of the state-of-art on blockchain applied in the context of robotic systems. The results of our study also show that security is the least considered quality attribute. On the other hand, quality attributes are only one among several parameters analyzed in our study, but not our main focus.

Bozhinoski et al. performed an extensive systematic mapping study of the state-of-the-art in software engineering research on solutions for managing safety for mobile robotic systems (Bozhinoski et al., 2019). To that end, they have defined 50 different parameters to perform the comparison, providing a very clear picture of both state-of-the-art in safety for mobile robots and its potential for industrial adoption.

In their work, Anjomshoae et al. clarify, map, and analyze the studies on explainable agents and robots published within the last ten years (Anjomshoae et al., 2019). To that end, they defined a set of nine parameters quite similar to the ones we defined to answer our research question RQ1. However, they do not focus on the potential for industrial adoption in their work.

Swanborn and Malavolta reviewed the existing research studies that focus on energy efficiency in robotics software (Swanborn and Malavolta, 2020). The context of this study is thus narrowed down to a very specific topic. Moreover, energy efficiency does not appear to be a quality attribute considered in our selected primary studies at all.

To the best of our knowledge, this is the first work that is not restricted to any specific field of robotics, neither in the type of robot nor application context, while focusing on many different classification parameters to analyze state-of-the-art in software engineering research in the context of robotic systems. Moreover, this study is restricted to ROS-based robotic systems only.

8. Conclusions

In this systematic mapping study, we evaluated the state-of-the-art and potential for industrial adoption of software engineering research on ROS. Specifically, we consider the **state-of-the-art** in software engineering research on ROS and its **potential of industrial adoption**.

The results from our analysis can aid researchers who are willing to contribute to this research field. Furthermore, our results can be used to gain a better understanding of the research field as a whole, serving as a starting point for further reading about software engineering research on ROS for both researchers and practitioners.

Such results also indicate that there is room for improving the alignment between research and industry in software engineering research on ROS. To facilitate this, we considered market demands and developments and proposed the following concrete routes of action:

- Researchers shall keep studying, applying, and evaluating their solutions in the context of *mobile robots*.
- Researchers shall investigate more the software engineering aspects of robotic systems involving *multiple robots*.
- Researchers shall focus their attention to *ROS2* and evaluate their solutions on ROS2 systems.
- Researchers shall better study how *Model-Driven Engineering* techniques can be successfully integrated in the context of real robotic projects (e.g., as a complement to already-in-place practices).
- Researchers and practitioners shall improve the *simulation of ROS systems* and better integrate it into current development practices.
- Researchers shall study more in depth the *community of ROS practitioners and its dynamics* (e.g., by analyzing public ROS packages, their source code, their contributors, interviewing practitioners, etc.).
- Practitioners are invited to stay up to date with respect to the latest research efforts about software engineering aspects on ROS.
- Researchers shall focus more on *non-functional aspects of ROS systems*, especially on performance and security.
- Researchers shall take the initiative towards making the technological evaluations of their proposed solutions more easily *transferable to the industry*.

As future work, our proposed classification framework can be extended to include additional parameters for addressing RQ2 (*e.g.,* the level of industrial background of the authors of the primary studies). Moreover, a qualitative study targeting practitioners for complementing our obtained results will give a more complete overview of the state of software engineering practices for ROS-based systems. Moreover, researchers are invited to replicate this study with a focus on specific software engineering knowledge areas (*e.g.,* software design) or quality attributes (*e.g.,* maintainability, security).

9. Acknowledgements

This research was partially funded by the Rijksdienst voor Ondernemend Nederland (RVO) through the ITEA3 BUMBLE project (18006), and the CNPQ/FA through the PPP-CP-20/2018 call.

CRediT authorship contribution statement

Michel Albonico: Investigation, Data curation, Visualization, Writing – review & editing. **Milica Đorđević:** Investigation, Data curation, Writing – original draft. **Engel Hamer:** Investigation, Data curation, Writing – original draft. **Ivano Malavolta:** Conceptualization, Methodology, Investigation, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Michel Albonico reports a relationship with Federal Technological University of Paraná - Campus Francisco Beltrão that includes: employment. Ivano Malavolta reports a relationship with Vrije Universiteit Amsterdam that includes: employment.

Data availability

The data is available in the replication package of the study (available on GitHub and linked from the paper).

Appendix A. List of top venues in software engineering and robotics

Software engineering venues

See Table 7.

Robotic venues

See Table 8.

Appendix B. Primary studies added via automatic searches

See Table 9.

Appendix C. Primary studies added via snowballing

See Table 10.

Table 7
Software engineering venues.

Name	Acronym	Source	Type	CORE rank	DBLP key
ACM Computing Surveys	CSUR	CORE journal	JOURNAL	A*	journals/csur
ACM Transactions on Computer Systems	TOCS	CORE journal	JOURNAL	A*	journals/tocs
ACM Transactions on Programming Languages and Systems	TOPLAS	CORE journal	JOURNAL	A*	journals/toplas
ACM Transactions on Software Engineering and Methodology	TOSEM	CORE journal	JOURNAL	A*	journals/tosem
IEEE Transactions on Computers	TC	CORE journal	JOURNAL	A*	journals/tc
IEEE Transactions on Multimedia	TMM	CORE journal	JOURNAL	A*	journals/tmm
IEEE Transactions on Services Computing	TSC	CORE journal	JOURNAL	A*	journals/tsc
IEEE Transactions on Software Engineering	TSE	CORE journal	JOURNAL	A*	journals/tse

(*continued on next page*)

Table 7 (*continued*).

Name	Acronym	Source	Type	CORE rank	DBLP key
Journal of Functional Programming	JFP	CORE journal	JOURNAL	A	journals/jfp
Constraints	CONSTRAINTS	CORE journal	JOURNAL	A	journals/constraints
IEEE Transactions on Information Forensics and Security	TIFS	CORE journal	JOURNAL	A	journals/tifs
Theoretical Computer Science	TCS	CORE journal	JOURNAL	A	journals/tcs
ACM Transactions on Privacy and Security (was ACM Transactions on Information and System Security, TISSEC pre 2018)	TISSEC	CORE journal	JOURNAL	A	journals/tissec
Empirical Software Engineering: an international journal	ESE	CORE journal	JOURNAL	A	journals/ese
IEEE Transactions on Dependable and Secure Computing	TDSC	CORE journal	JOURNAL	A	journals/tdsc
IEEE Transactions on Reliability	TR	CORE journal	JOURNAL	A	journals/tr
Journal of Systems and Software	JSS	CORE journal	JOURNAL	A	journals/jss
Science of Computer Programming	SCP	CORE journal	JOURNAL	A	journals/scp
Theory and Practice of Logic Programming	TPLP	CORE journal	JOURNAL	A	journals/tplp
Information and Software Technology	INFSOF	CORE journal	JOURNAL	A	journals/infsof
IEEE Software	SOFTWARE	Google Scholar	JOURNAL	B	journals/software
Software and Systems Modeling	SOSYM	Google Scholar	JOURNAL	B	journals/sosym
Software: Practice and Experience	SPE	Google Scholar	JOURNAL	B	journals/spe
Proceedings of the ACM on Programming Languages	PACMPL	Google Scholar	JOURNAL	-	journals/pacmpl
Architectural Support for Programming Languages and Operating Systems	ASPLOS	CORE conf	CONF	A*	conf/asplos
Computer Aided Verification	CAV	CORE conf	CONF	A*	conf/cav
European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering	ESEC	CORE conf	CONF	A*	conf/esec
ACM SIGSOFT International Symposium on Foundations of Software Engineering	SIGSOFT	Google Scholar	CONF	-	conf/sigsoft
International Conference on Functional Programming	ICFP	CORE conf	CONF	A*	conf/icfp
International Conference on Software Engineering	ICSE	CORE conf	CONF	A*	conf/icse
ACM International Symposium on Computer Architecture	ISCA	CORE conf	CONF	A*	conf/isca
ACM Conference on Object Oriented Programming Systems Languages and Applications	OOPSLA	CORE conf	CONF	A*	conf/oopsla
ACM-SIGPLAN Conference on Programming Language Design and Implementation	PLDI	CORE conf	CONF	A*	conf/pldi
ACM-SIGACT Symposium on Principles of Programming Languages	POPL	CORE conf	CONF	A*	conf/popl
Measurement and Modeling of Computer Systems	SIGMETRICS	CORE conf	CONF	A*	conf/sigmetrics
International Conference on Software Testing, Verification and Validation	ICST	CORE conf	CONF	A	conf/icst
International Conference on Model Driven Engineering Languages and Systems (Previously UML, changed in 2005)	MODELS	CORE conf	CONF	A	conf/models
IEEE International Working Conference on Mining Software Repositories	MSR	CORE conf	CONF	A	conf/msr
International Symposium on Software Testing and Analysis	ISSTA	CORE conf	CONF	A	conf/issta
IEEE International Conference on Software Maintenance and Evolution (prior to 2014 was ICSM, IEEE International Conference on Software Maintenance)	ICSM	CORE conf	CONF	A	conf/icsm

(*continued on next page*)

Table 7 (*continued*).

Name	Acronym	Source	Type	CORE rank	DBLP key
European Conference on Software Architecture	ECSA	CORE conf	CONF	A	conf/ecsa
Automated Software Engineering Conference	KBSE	CORE conf	CONF	A	conf/kbse
International Symposium on Software Reliability Engineering	ISSRE	CORE conf	CONF	A	conf/issre
International Symposium on Empirical Software Engineering and Measurement	ESEM	CORE conf	CONF	A	conf/esem
International Conference on Evaluation and Assessment in Software Engineering	EASE	CORE conf	CONF	A	conf/ease
International Symposium on Automated Technology for Verification and Analysis	ATVA	CORE conf	CONF	A	conf/atva
International Conference in Business Process Management	BPM	CORE conf	CONF	A	conf/bpm
International Symposium on Code Generation and Optimization	CGO	CORE conf	CONF	A	conf/cgo
European Conference on Object-Oriented Programming	ECOOP	CORE conf	CONF	A	conf/ecoop
European Symposium on Programming	ESOP	CORE conf	CONF	A	conf/esop
International Symposium on Formal Methods (was Formal Methods Europe FME)	FM	CORE conf	CONF	A	conf/fm
Foundations of Software Science and Computational Structures	FOSSACS	CORE conf	CONF	A	conf/fossacs
International Colloquium on Automata Languages and Programming	ICALP	CORE conf	CONF	A	conf/icalp
IEEE International Conference on Engineering of Complex Computer Systems	ICECCS	CORE conf	CONF	A	conf/iceccs
International Conference on Software and System Processes (was ICSP prior to 2011)	ISPW	CORE conf	CONF	A	conf/ispw
International Symposium on Memory Management	IWMM	CORE conf	CONF	A	conf/iwmm
IFIP International Symposium on Computing Performance, Modelling, Measurement and Evaluation	PERFOR-MANCE	CORE conf	CONF	A	conf/performance
IEEE International Requirements Engineering Conference	RE	CORE conf	CONF	A	conf/re
Static Analysis Symposium	SAS	CORE conf	CONF	A	conf/sas
ACM Symposium on Software Reusability	SSR	CORE conf	CONF	A	conf/ssr
Empirical Software Engineering	ESE	Google Scholar	CONF	-	conf/ese
Symposium on Operating Systems Principles (SOSP)	SOSP	Google Scholar	CONF	-	conf/sosp
IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER)	WCRE	Google Scholar	CONF	-	conf/wcre
International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)	TACAS	Google Scholar	CONF	-	conf/tacas

Table 8
Robotic venues.

Name	Acronym	Source	Type	CORE rank	DBLP key
IEEE International Conference on Robotics and Automation	ICRA	Google Scholar	CONF	B	conf/icra
IEEE Robotics and Automation Letters	RAL	Google Scholar	JOURNAL	–	journals/ral
Science Robotics	SCIROBOTICS	Google Scholar	JOURNAL	–	journals/scirobotics
IEEE/RSJ International Conference on Intelligent Robots and Systems	IROS	Google Scholar	CONF	A	conf/iros
IEEE Transactions on Robotics	TROB	Google Scholar	JOURNAL	A*	journals/trob
The International Journal of Robotics Research	IJRR	Google Scholar	JOURNAL	A*	journals/ijrr
Robotics and Computer-Integrated Manufacturing	RCIM	Google Scholar	JOURNAL	–	journals/rcim
Robotics: Science and Systems	RSS	Google Scholar	CONF	A*	conf/rss
Robotics and Autonomous Systems	RAS	Google Scholar	JOURNAL	B	journals/ras
ACM/IEEE International Conference on Human Robot Interaction	HRI	Google Scholar	CONF	–	conf/hri
Autonomous Robots	AROBOTS	Google Scholar	JOURNAL	B	journals/arobots
Journal of Field Robotics	JFR	Google Scholar	JOURNAL	A	journals/jfr
Journal of Intelligent & Robotic Systems	JIRS	Google Scholar	JOURNAL	C	journals/jirs
Frontiers in Robotics and AI	FIRAI	Google Scholar	JOURNAL	–	journals/firai
International Journal of Social Robotics	IJSR	Google Scholar	JOURNAL	–	journals/ijsr
IEEE Robotics & Automation Magazine	RAM	Google Scholar	JOURNAL	C	journals/ram
International Conference on Control, Automation, Robotics and Vision	ICARCV	CORE Conf	CONF	A	conf/icarcv
International Symposium on Robotics Research	ISRR	CORE Conf	CONF	A	conf/isrr

Table 9
Primary studies added via automatic searches.

Study	Title	Authors	Venue	Year
P1	IoT-Based Activity Recognition for Process Assistance in Human-Robot Disaster Response	Adrian Rebmann, Jana-Rebecca Rehse, Mira Pinter, Marius Schnaubelt, Kevin Daun, Peter Fettke	International Conference in Business Process Management (BPM)	2020
P2	Formal Modeling and Automatic Code Synthesis for Robot System	Xinxin Li, Rui Wang, Yu Jiang, Yong Guan, Xiaojuan Li, Xiaoyu Song	IEEE International Conference on Engineering of Complex Computer Systems (ICECCS)	2017
P3	A use case in model-based robot development using AADL and ROS	Gianluca Bardaro, Andrea Semprebon, Matteo Matteucci	International Workshop on Robotics Software Engineering (RoSE)@ICSE	2018
P4	Towards rapid composition with confidence in robotics software	Neil A. Ernst, Rick Kazman, Philip Bianco	International Workshop on Robotics Software Engineering (RoSE)@ICSE	2018
P5	The forgotten case of the dependency bugs: on the example of the robot operating system	Anders Fischer-Nielsen, Zhoulai Fu, Ting Su, Andrzej Wasowski	International Conference on Software Engineering (ICSE)	2020
P6	Formal Verification of ROS-Based Robotic Applications Using Timed-Automata	Raju Halder, José Proença, Nuno Macedo, André Santos	International FME Workshop on Formal Methods in Software Engineering (FME)@ICSE	2017
P7	Machine learning meets quantitative planning: enabling self-adaptation in autonomous robots	Pooyan Jamshidi, Javier Cámara, Bradley R. Schmerl, Christian Kästner, David Garlan	International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)@ICSE	2019
P8	How do you architect your robots?: state of the practice and guidelines for ROS-based systems	Ivano Malavolta, Grace A. Lewis, Bradley R. Schmerl, Patricia Lago, David Garlan	International Conference on Software Engineering (ICSE)	2020
P9	High level synthesis of ROS protocol interpretation and communication circuit for FPGA	Takeshi Ohkawa, Yuhei Sugata, Harumi Watanabe, Nobuhiko Ogura, Kanemitsu Ootsu, Takashi Yokota	International Workshop on Robotics Software Engineering (RoSE)@ICSE	2019
P10	Towards code-aware robotic simulation: vision paper	John-Paul Ore, Carrick Detweiler, Sebastian G. Elbaum	International Workshop on Robotics Software Engineering (RoSE)@ICSE	2018
P11	Checking consistency of robot software architectures in ROS	Thomas Witte, Matthias Tichy	International Workshop on Robotics Software Engineering (RoSE)@ICSE	2018
P12	It Takes a Village to Build a Robot: An Empirical Study of The ROS Ecosystem	Sophia Kolak, Afsoon Afzal, Claire Le Goues, Michael Hilton, Christopher Steven Timperley	IEEE International Conference on Software Maintenance and Evolution (ICSME)	2020
P13	Live programming in practice: A controlled experiment on state machines for robotic behaviors	Miguel Campusano, Johan Fabry, Alexandre Bergel	Information and Software Technology (IST)	2019
P14	Feasible and stressful trajectory generation for mobile robots	Carl Hildebrandt, Sebastian G. Elbaum, Nicola Bezzo, Matthew B. Dwyer	International Symposium on Software Testing and Analysis (ISSTA)	2020
P15	Real-time control architecture based on Xenomai using ROS packages for a service robot	Raimarius Delgado, Bum-Jae You, Byoung-Wook Choi	Journal of Systems and Software (JSS)	2019
P16	The Robot Operating System: Package reuse and community dynamics	Pablo Estefo, Jocelyn Simmonds, Romain Robbes, Johan Fabry	Journal of Systems and Software (JSS)	2019
P17	A Hybrid Editor for Fast Robot Mission Prototyping	Thomas Witte, Matthias Tichy	International Workshop on Explainable Software (EXPLAIN)	2019
P18	Robotic system testing with AMSA framework	Hamza El Baccouri, Goulven Guillou, Jean-Philippe Babau	International Conference Workshop on Model-Driven Engineering Tools (MDETools)	2018
P19	AC-ROS: assurance case driven adaptation for the robot operating system	Betty H. C. Cheng, Robert Jared Clark, Jonathon Emil Fleck, Michael Austin Langford, Philip K. McKinley	International Conference on Model Driven Engineering Languages and Systems (MODELS)	2020
P20	Bootstrapping MDE Development from ROS Manual Code - Part 2: Model Generation	Nadia Hammoudeh Garcia, Ludovic Delval, Mathias Lüdtke, André Santos, Björn Kahl, Mirko Bordignon	International Conference on Model Driven Engineering Languages and Systems (MODELS)	2019

(*continued on next page*)

Table 9 (*continued*).

Study	Title	Authors	Venue	Year
P21	Generating a ROS/JAUS bridge for an autonomous ground vehicle	Patrick Morley, Alex Warren, Ethan Rabb, Sean Whitsitt, Matt Bunting, Jonathan Sprinkle	ACM Workshop on Domain-specific Modeling (DSM)@OOPSLA	2013
P22	Property-based testing for the robot operating system	André Santos, Alcino Cunha, Nuno Macedo	ACM SIGSOFT International Workshop on Automating TEST Case Design (A-TEST)@SIGSOFT	2018
P23	Robotics software engineering: a perspective from the service robotics domain	Sergio García, Daniel Strüber, Davide Brugali, Thorsten Berger, Patrizio Pelliccione	ACM SIGSOFT International Symposium on Foundations of Software Engineering (ESEC/FSE)	2020
P24	Model-Based Adaptation for Robotics Software	Jonathan Aldrich, David Garlan, Christian Kästner, Claire Le Goues, Anahita Mohseni-Kabir, Ivan Ruchkin, Selva Samuel, Bradley R. Schmerl, Christopher Steven Timperley, Manuela Veloso, Ian Voysey, Joydeep Biswas, Arjun Guha, Jarrett Holtz, Javier Cámara, Pooyan Jamshidi	IEEE Software	2019
P25	Runtime Verification on Hierarchical Properties of ROS-Based Robot Swarms	Chi Hu, Wei Dong, Yonghui Yang, Hao Shi, Ge Zhou	IEEE Transactions on Reliability	2020
P26	Automatic configuration of ROS applications for near-optimal performance	José Cano, Alejandro Bordallo, Vijay Nagarajan, Subramanian Ramamoorthy, Sethu Vijayakumar	International Conference on Intelligent Robots and Systems (IROS)	2016
P27	Application-level security for ROS-based applications	Bernhard Diebe, Severin Kacianka, Stefan Rass, Peter Schartner	International Conference on Intelligent Robots and Systems (IROS)	2016
P28	Scanning the Internet for ROS: A View of Security in Robotics Research	Nicholas DeMarinis, Stefanie Tellex, Vasileios P. Kemerlis, George Dimitri Konidaris, Rodrigo Fonseca	International Conference on Robotics and Automation (ICRA)	2013
P29	ROS commander (ROSCo): Behavior creation for home robots	Hai Nguyen, Matei T. Ciocarlie, Kaijen Hsiao, Charles C. Kemp	International Conference on Robotics and Automation (ICRA)	2013
P30	ROSlink: Interfacing legacy systems with ROS	Fabio Dalla Libera, Hiroshi Ishiguro	International Conference on Robotics and Automation (ICRA)	2013
P31	ROS-based online robot programming for remote education and training	Gustavo A. Casañ, Enric Cervera, Amine Abou Moughlbay, Jaime Alemany, Philippe Martinet	International Conference on Robotics and Automation (ICRA)	2015
P32	Quantitative and Qualitative Evaluation of ROS-Enabled Local and Global Planners in 2D Static Environments	Alexandros Filotheou, Emmanouil G. Tsardoulias, Antonis G. Dimitriou, Andreas L. Symeonidis, Loukas Petrou	Journal of Intelligent and Robotic Systems (JIRS)	2020

Table 10
Primary studies added via snowballing.

Study	Title	Authors	Venue	Year
S1	Towards an Actor-based Approach to Design Verified ROS-based Robotic Programs using Rebeca	Dehnavi, Saeid; Sedaghatbaf, Ali; Salmani, Bahar; Sirjani, Marjan; Kargahi, Mehdi; Khamespanah, Ehsan	Procedia Computer Science	2019
S2	From Models to Software Through Automatic Transformations: An AADL to ROS End-to-End Toolchain	Bardaro, Gianluca; Semprebon, Andrea; Chiatti, Agnese; Matteucci, Matteo	IEEE International Conference on Robotic Computing (IRC)	2019
S3	Generating ROS-based Software for Industrial Cyber-Physical Systems from UML/MARTE	Wehrmeister, Marco Aurelio	IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)	2020
S4	Mining Energy-Related Practices in Robotics Software	Albonico, Michel; Malavolta, Ivano; Pinto, Gustavo; Guzman, Emitza; Chinnappan, Katerina; Lago, Patricia	Mining Software Repositories (MSR)	2021
S5	Verification of system-wide safety properties of ROS applications	Carvalho, Renato; Cunha, Alcino; Macedo, Nuno; Santos, André	IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)	2020
S6	Robot Runner: A Tool for Automatically Executing Experiments on Robotics Software	Swanborn, Stan; Malavolta, Ivano	IEEE/ACM International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)	2021
S7	Designing Drone Systems with Papyrus for Robotics	Radermacher, Ansgar; Morelli, Matteo; Hussein, Mahmoud; Nouacer, Reda	Drone Systems Engineering and Rapid Simulation and Performance Evaluation: Methods and Tools Proceedings	2021
S8	Documentation and Modeling of ROS Systems	Drumheller, William R.; Conner, David C.	SoutheastCon	2021
S9	Bootstrapping MDE Development from ROS Manual Code - Part 1: Metamodeling	Hammoudeh Garcia, Nadia; Lüdtke, Mathias; Kortik, Sitar; Kahl, Björn; Bordignon, Mirko	IEEE International Conference on Robotic Computing (IRC)	2019
S10	Towards application level testing of ROS networks	Breitenhuber, Guido	IEEE International Conference on Robotic Computing (IRC)	2020
S11	Testing, Verification and Improvements of Timeliness in ROS Processes	Okumuş, Fatih; Kocamaz, Adnan Fatih	International Artificial Intelligence and Data Processing Symposium (IDAP)	2019
S12	Mining the ROS ecosystem for Green Architectural Tactics in Robotics and an Empirical Evaluation	Malavolta, Ivano; Chinnappan, Katerina; Swanborn, Stan; Lewis, Grace; Lago, Patricia	Mining Software Repositories (MSR)	2021
S13	Mining guidelines for architecting robotics software	Malavolta, Ivano; Lewis, Grace A.; Schmerl, Bradley; Lago, Patricia; Garlan, David	Journal of Systems and Software	2021
S14	A Visual Modeling Language for RDIS and ROS Nodes Using AToM3	Kilgo, Paul; Syriani, Eugene; Anderson, Monica	Simulation, Modeling, and Programming for Autonomous Robots	2012
S15	A framework for quality assessment of ROS repositories	Santos, André; Cunha, Alcino; Macedo, Nuno; Lourenço, Cláudio	IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)	2016
S16	Static-Time Extraction and Analysis of the ROS Computation Graph	Santos, André; Cunha, Alcino; Macedo, Nuno	Third IEEE International Conference on Robotic Computing (IRC)	2019
S17	Mining the usage patterns of ROS primitives	Santos, André; Cunha, Alcino; Macedo, Nuno; Arrais, Rafael; dos Santos, Filipe Neves	IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)	2017
S18	ROSRV: Runtime Verification for Robots	Huang, Jeff; Erdogan, Cansu; Zhang, Yi; Moore, Brandon; Luo, Qingzhou; Sundaresan, Aravind; Rosu, Grigore	Runtime Verification	2014

(*continued on next page*)

Table 10 (*continued*).

Study	Title	Authors	Venue	Year
S19	Evaluating impact in the ROS ecosystem	Curran, William; Thornton, Thomas; Arvey, Benjamin; Smart, William D.	IEEE International Conference on Robotics and Automation (ICRA)	2015
S20	Exploring the performance of ROS2	Maruyama, Yuya; Kato, Shinpei; Azumi, Takuya	International Conference on Embedded Software	2016
S21	Code duplication in ROS launchfiles	Estefo, Pablo; Robbes, Romain; Fabry, Johan	International Conference of the Chilean Computer Science Society (SCCC)	2015
S22	Model-based integration testing of ROS packages: A mobile robot case study	Ernits, Juhan; Halling, Evelin; Kanter, Gert; Vain, Jüri	European Conference on Mobile Robots (ECMR)	2015
S23	Flexible Navigation: Finite state machine-based integrated navigation and control for ROS enabled robots	Conner, David C.; Willis, Justin	SoutheastCon	2017
S24	A Study on ROS Vulnerabilities and Countermeasure	Jeong, Se-Yeon; Choi, I-Ju; Kim, Yeong-Jin; Shin, Yong-Min; Han, Jeong-Hun; Jung, Goo-Hong; Kim, Kyoung-Gon	ACM/IEEE International Conference on Human-Robot Interaction	2017
S25	Applying Software Static Analysis to ROS: The Case Study of the FASTEN European Project	Neto, Tiago; Arrais, Rafael; Sousa, Armando; Santos, André ; Veiga, Germano	Iberian Robotics Conference	2020
S26	From High-Level Task Specification to Robot Operating System (ROS) Implementation	Wong, Kai Weng; Kress-Gazit, Hadas	IEEE International Conference on Robotic Computing (IRC)	2017
S27	Using AADL to Model and Develop ROS-Based Robotic Application	Bardaro, Gianluca; Matteucci, Matteo	IEEE International Conference on Robotic Computing (IRC)	2017
S28	Rate impact analysis in robotic systems	Sharma, Nishant; Elbaum, Sebastian; Detweiler, Carrick	IEEE International Conference on Robotics and Automation (ICRA)	2017
S29	A Case Study on Improving the Software Dependability of a ROS Path Planner for Steep Slope Vineyards	Santos, Luıs Carlos; Santos, André; Santos, Filipe Neves; Valente, António	Robotics	2021
S30	Comprehensive Simulation of Quadrotor UAVs Using ROS and Gazebo	Meyer, Johannes; Sendobry, Alexander; Kohlbrecher, Stefan; Klingauf, Uwe; von Stryk, Oskar	International Conference on Simulation, Modeling, and Programming for Autonomous	2012
S31	Compositional Design of Multi-Robot Systems Control Software on ROS	Spellini, Stefano; Lora, Michele; Fummi, Franco; Chattopadhyay, Sudipta	ACM Transactions on Embedded Computing Systems	2019

References

Alshuqayran, Nuha, Ali, Nour, Evans, Roger, 2016. A systematic mapping study in microservice architecture. In: 2016 IEEE 9th International Conference on Service-Oriented Computing and Applications. SOCA, IEEE, pp. 44–51.
Anjomshoae, Sule, Najjar, Amro, Calvaresi, Davide, Främling, Kary, 2019. Explainable agents and robots: Results from a systematic literature review. In: 18th International Conference on Autonomous Agents and Multiagent Systems. AAMAS 2019, Montreal, Canada, May 13–17, 2019, International Foundation for Autonomous Agents and Multiagent Systems, pp. 1078–1088.
Anon, 2019. Mobile robotic systems: Market update. ABI Research: For Visionaries.
Anon, 2020. Fixed vs mobile – Let the robot wars begin!
Anon, 2020a. Collaborative Robot (Cobot) Market. Market Research Firm.
Anon, 2020b. Collaborative Robot (Cobot) Market: Global opportunity analysis and industry forecast, 2019–2026. Market Research Firm.
Anon, 2021. Wiki. Ros.Org.
Basili, V., Caldiera, G., Rombach, H., 1994. Goal question metric paradigm. In: Encyclopedia of Software Engineering, Vol. 1. pp. 528–532.
Ben-Ari, Mordechai, Mondada, Francesco, 2018. Robots and their applications. In: Elements of Robotics. Springer, pp. 1–20.
Bourque, Pierre, Fairley, R., 2004. Swebok. Nd: IEEE Computer Society.
Bozhinoski, Darko, Di Ruscio, Davide, Malavolta, Ivano, Pelliccione, Patrizio, Crnkovic, Ivica, 2019. Safety for mobile robotic systems: A systematic mapping study from a software engineering perspective. J. Syst. Softw. 151, 150–179.
Bucchiarone, Antonio, Cabot, Jordi, Paige, Richard F, Pierantonio, Alfonso, 2020. Grand challenges in model-driven engineering: an analysis of the state of the research. Softw. Syst. Model. 19 (1), 5–13.
dos Santos, Marcela G, Napoleão, Bianca M, Petrillo, Fabio, Ameyed, Darine, Jaafar, Fehmi, 2020. A preliminary systematic mapping on software engineering for robotic systems: A software quality perspective. In: Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops. pp. 647–654.
Estefo, Pablo, Simmonds, Jocelyn, Robbes, Romain, Fabry, Johan, 2019. The robot operating system: Package reuse and community dynamics. J. Syst. Softw. 151, 226–242.
García, Sergio, Strüber, Daniel, Brugali, Davide, Berger, Thorsten, Pelliccione, Patrizio, 2020. Robotics software engineering: A perspective from the service robotics domain. In: Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. pp. 593–604.
Gong, Zheng, Xue, Wuyang, Liu, Ziang, Zhao, Yimo, Miao, Ruihang, Ying, Rendong, Liu, Peilin, 2019. Design of a reconfigurable multi-sensor testbed for autonomous vehicles and ground robots. In: 2019 IEEE International Symposium on Circuits and Systems. ISCAS, IEEE, pp. 1–5.
Grieves, Michael, Vickers, John, 2017. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In: Transdisciplinary Perspectives on Complex Systems. Springer, pp. 85–113.
Hood, Deanna, Woodall, William, 2016. ROS 2 Update. ROSCon.
Hrabia, Christopher-Eyk, 2016. A framework for adaptive and goal-driven behaviour control of multi-robot systems. In: 2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems. FAS* W, IEEE, pp. 275–280.
Hyman, Ray, 1982. Quasi-experimentation: Design and analysis issues for field settings (book). J. Personal. Assess. 46 (1), 96–97.
ISO, ISO, 2011. IEC/IEEE systems and software engineering: Architecture description. ISO/IEC/IEEE 42010: 2011 (E)(Revision of ISO/IEC 42010: 2007 and IEEE Std 1471-2000), IEEE New York, NY.
ISO/IEC 25010:2011, 2011. Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models. https://www.iso.org/standard/35733.html,
Ivarsson, Martin, Gorschek, Tony, 2011. A method for evaluating rigor and industrial relevance of technology evaluations. Empir. Softw. Eng. 16 (3), 365–395.
Jamshidi, Pooyan, Ahmad, Aakash, Pahl, Claus, 2013. Cloud migration research: A systematic review. IEEE Trans. Cloud Comput. 1 (2), 142–157.
Jung, Jin Hwa, Lim, Dong-Geon, 2020. Industrial robots, employment growth, and labor cost: A simultaneous equation analysis. Technol. Forecast. Soc. Change 159, 120202.
Kitchenham, Barbara, Brereton, Pearl, 2013. A systematic review of systematic review process research in software engineering. Inf. Softw. Technol. 55 (12), 2049–2075.
Macenski, Steve, 2020. The life and times of ROS2: A look at the future of robotics infrastructure. RoboBusiness.
Macenski, Steven, Foote, Tully, Gerkey, Brian, Lalancette, Chris, Woodall, William, 2022. Robot operating system 2: Design, architecture, and uses in the wild. Science Robotics 7 (66), eabm6074.
Malavolta, Ivano, Lewis, Grace A, Schmerl, Bradley, Lago, Patricia, Garlan, David, 2021. Mining guidelines for architecting robotics software. J. Syst. Softw. 178, 110969.
McHugh, Mary L., 2012. Interrater reliability: The kappa statistic. Biochem. Med. 22 (3), 276–282.
Petersen, Kai, Feldt, Robert, Mujtaba, Shahid, Mattsson, Michael, 2008. Systematic mapping studies in software engineering. In: 12th International Conference on Evaluation and Assessment in Software Engineering. EASE 12, pp. 1–10.
Petersen, Kai, Vakkalanka, Sairam, Kuzniarz, Ludwik, 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. Inf. Softw. Technol. 64, 1–18.
Popay, Jennie, Roberts, Helen, Sowden, Amanda, Petticrew, Mark, Arai, Lisa, Rodgers, Mark, Britten, Nicky, Roen, Katrina, Duffy, Steven, et al., 2006. Guidance on the conduct of narrative synthesis in systematic reviews. p. b92, A Product from the ESRC Methods Programme Version, 1.
Quigley, Morgan, Conley, Ken, Gerkey, Brian, Faust, Josh, Foote, Tully, Leibs, Jeremy, Wheeler, Rob, Ng, Andrew Y, 2009. ROS: An open-source robot operating system. In: ICRA Workshop on Open Source Software, Vol. 3. Kobe, Japan, p. 5.
Santos, Marcela G. dos, Petrillo, Fabio, 2021. Software engineering for robotic systems: A systematic mapping study. arXiv preprint arXiv:2102.12520.
Sardi, Lamyae, Idri, Ali, Fernández-Alemán, José Luis, 2017. A systematic review of gamification in e-Health. J. Biomed. Inform. 71, 31–48.
Sheng, Weihua, Yang, Qingyan, Tan, Jindong, Xi, Ning, 2006. Distributed multi-robot coordination in area exploration. Robot. Auton. Syst. 54 (12), 945–955.
Su, Lian Jye, 2019. ROS on the rise. ABI Research: For Visionaries.
Swanborn, Stan, Malavolta, Ivano, 2020. Energy efficiency in robotics software: A systematic literature review. In: Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering Workshops pp. 144–151.
The dblp team, 2021. dblp computer science bibliography: Monthly snapshot release of February 2021.
Whittle, Jon, Hutchinson, John, Rouncefield, Mark, 2013. The state of practice in model-driven engineering. IEEE Softw. 31 (3), 79–85.
Wieringa, Roel, Maiden, Neil, Mead, Nancy, Rolland, Colette, 2006. Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. Requir. Eng. 11 (1), 102–107.
Wohlin, Claes, 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering. pp. 1–10.
Wohlin, Claes, Runeson, Per, Höst, Martin, Ohlsson, Magnus C, Regnell, Björn, Wesslén, Anders, 2012. Experimentation in Software Engineering. Springer Science & Business Media.

Michel Albonico Adjunct professor (tenured) at the Federal University of Technology - Paraná (UTFPR), Brazil, where he also leads students in innovation and software projects. He also leads the research group (IntelAgir), as well as technological and research projects. He holds a Ph.D. in Software Engineering from the IMT Atlantique Bretagne Pays de la Loire, France. He worked for one year as a postdoc researcher at the S2 Group at the Vrije Universiteit Amsterdam, The Netherlands, where he is collaborating as an external researcher. His current research focuses on empirical software engineering for cyber–physical systems, more specifically IoT and Robotics.

Milica Đorđević Junior Full-Stack Software Engineer at Advidi, Amsterdam, The Netherlands In 2021 she received a Master degree in Computer Science (with specialization on Software Engineering and Green IT) from the Vrije Universiteit Amsterdam, The Netherlands Her research interests include software engineering, robotics, software development, software design, and software architecture.

Engel Hamer Founder and business developer at eJournal, Amsterdam, The, Netherlands In 2021 he received a Master degree in Computer Science (with specialization on Software Engineering and Green IT) from the Vrije Universiteit Amsterdam, The Netherlands His research interests include software engineering and robotics.

Ivano Malavolta Associate professor and Director of the Network Institute at the Vrije Universiteit Amsterdam (The Netherlands). His research focuses on data-driven software engineering, with a special emphasis on software architecture, mobile software development, robotics software. He applies empirical methods to assess practices and trends in the field of software engineering. He authored several scientific articles in international journals and peer-reviewed international conferences proceedings. He is program committee member and reviewer of international conferences and journals in the software engineering field. He received a Ph.D. in computer science from the University of L'Aquila in 2012. He is a member of ACM, IEEE, VERSEN, and Amsterdam Data Science.