Requirements Engineering for Embedded Systems: A Systematic Literature Review

Tarcísio Pereira[1], Quelita Ribeiro[2], Mozart Melo[3], Sthefanie Magro[2], Fernanda Alencar[2], and Jaelson Castro[2]

[1] Instituto Federal de Pernambuco, Caruaru, PE - Brazil
tarcisio.pereira@caruaru.ifpe.edu.br
[2] Universidade Federal de Pernambuco, Recife, PE - Brazil
{qadsr, sdm2}@cin.ufpe.br, fernandaalenc@gmail.com, jbc@cin.ufpe.br
[3] Universidade de Pernambuco, Recife, PE - Brazil
mmaj@ecomp.poli.br

Abstract. Context: Most problems that occur at system delivery of Embedded Systems (ES) are related to requirements misconceptions in capturing requirements. Hence, a Requirements Engineering (RE) process is crucial to meet time, cost, and quality goals. **Objective:** We investigate the studies to analyze and understand how the RE approaches are used for embedded systems development. **Method:** We conducted a Systematic Literature Review (SLR) as the basis for our work. **Results:** We uncovered 92 studies and found out the RE phases that are attracting more attention, the main requirements modeling styles used, the types of requirements modelled, and the existing challenges/problems. We also proposed a research agenda. **Conclusions:** This paper is a step towards developing a body of knowledge in RE for ES that is derived from a large-scale SLR. We believe the results will benefit both researchers and practitioners.

Keywords: Embedded Systems · Requirements Engineering · Systematic Literature Review.

1 Introduction

An Embedded System (ES) can be defined as a system that regulates a physical device by sending control signals to actuators in reaction to input signals provided by its users and by sensors capturing the relevant state parameters of the system [4]. In everyday life, people are dependent on several services supported by embedded software, many of them transparent to the final user. The majority of ES are less visible, and for example they run in engines, brakes, seat belts, airbag, and audio system in your car. Additionally, they command robots on a factory floor, power generation in a power plant, processes in a chemical plant, and traffic lights in a city [14].

Embedded systems are known for their high complexity, caused by the increasing number of functions and the growing number of interactions among different functions. Due to the complexity, the risk of undetected errors and deficiencies increases considerably. According to Broy and Stauner [4], in the embedded system domain, most problems are detected when the system is delivered. However, the problems are not related to the correctness of implementation but with misconceptions in requirements capture. Improper Requirements Engineering (RE) practices may result in incomplete requirements, incorrect elicitation and specification, high complexity, and economic or human loss. Inappropriate Requirements Engineering (RE) practices may result in incomplete requirements, incorrect elicitation and specification, high complexity, and economic or human loss. Hence, an adequate requirements engineering process is crucial to meet time, cost, and to develop quality embedded systems [21].

Requirements engineering problems continue to occur despite the efforts and advances in their understanding [18]. Due to their unique properties, ES require different approaches, methods, and tools to improve their quality. Some studies provide insights into the state of art and practice of RE for embedded systems [8,21,22,24]. However, there is no recent, systematic attempt to perform an extensive identification, mapping, and constraints of requirements approaches for ES.

Hence, the goal of this work is to conduct a Systematic Literature Review (SLR) to identify and analyze the Requirements Engineering (RE) approaches for Embedded Systems (ES). In doing so, we investigate the RE research problems addressed by them. We want to find out which RE phases (eg. elicitation, analysis, specification, validation and management) are attracting more attention to the ES community and which ones deserve a special care. It is also important to investigate the main requirements modeling styles (scenario-based, goal-oriented, textual requirements, etc) used. It is critical to discover whether the RE approaches for ES have been used to manage both functional and non-functional requirements.

In this paper, we perform an SLR [10] to evaluate and synthesize the evidence available in the literature to answer research questions (see Table 1) related to the use of approaches, methods, techniques, and processes to support the RE in the ES domain. We also summarize the existing evidence concerning RE for embedded systems to highlight empirical evidence of the challenges/problems of the published studies [10]. Last but not least, we propose a research agenda to RE community.

The results presented in this systematic review can be beneficial to the RE community, since it gathers evidence from the primary studies included in the review, forming a body of knowledge regarding requirements engineering for embedded systems.

The remainder of this paper is organized as follows: Section 2 presents related work. The research methodology adopted to conduct the SLR is outlined in Section 3. The results and the analysis of our research questions are described in Section 4. Finally, conclusions and future works are shown in Section 5.

2 Related work

Similar to our work, another secondary study that synthesizes RE in embedded systems domain is discussed in [22]. The authors present an SLR on requirements elicitation and specification for embedded systems. Their work differs from ours by means of a time interval, RE activities, the number of databases, and research questions. Our SLR considered studies from a 48 year period. It took into account all activities of the RE process and included seven databases in the studies selection, while [22] considered studies from 2000 to 2014 and only included those related to elicitation and specification activities. Regarding the databases, the authors in [22] used six, while we considered seven. It is important to highlight that our research questions are different from those proposed in [22].

Ossada et al. [19] presented a Requirements Elicitation Guide for Embedded Systems (GERSE). The authors propose a set of organized and practical requirements elicitation activities for the domain of embedded systems. However, their work focuses only on elicitation activity of the RE process. Although the authors stated that their study contributed to decrease the gap between software engineering and embedded systems engineering, they suggested that more work is required to include activities related to production and control quality of the product. In our SLR we considered studies that address at least one RE phase, and we also provide a set of embedded systems characteristics to support elicitation and specification of these systems.

In 2012, Sikora et al. published the results [21] of an industrial survey in the embedded systems domain to identify RE needs for ES. In their survey, the authors focused on five aspects of RE: use of natural language versus requirements models; support for high system complexity; quality assurance of requirements; the transition between RE and design; and the interrelation of requirements engineering and safety engineering. Our work differs from [21] in two main aspects. First, we performed a retrospective on RE for embedded systems. This, in turn, contributes to the identification of RE approaches that solve some particular RE problem. Second, we considered studies from academy and industry. This has helped us to identify a more significant number of concepts and challenges on RE for ES compared to the results depicted in [21].

Although the above works cover several aspects related to requirements engineering for ES, none of these works perform an extensive identification and mapping of requirements approaches for embedded systems.

In our previous work [20], we presented some limited results. It differs from this new work in terms of time interval, method description, presentation and discussion of the results, research questions, and research agenda. It is worth noting that this SLR was updated and considered studies from 1970 to June 2019, while [20] recognized studies from 1970 to October 2016. As a result, seventeen new studies were considered. The quality assessment of the studies, as well as their overview, were not described in [20]. In this work, we carefully detailed the quality assessment criteria, and we provided an overview of the studies by publication year, application context, research method, and type of source. Finally, the findings presented in this work are more representative. Moreover, we also provide a set of questions for further investigation.

3 Research methodology

A systematic literature review, as well as other kinds of review studies, is an exploratory study to investigate evidence in the literature about a specific theme [12]. To perform this SLR, we used guidelines and the protocol template proposed by Kitchenham and Charters [10], whose process involves several activities grouped into three main phases: planning, conducting, and reporting of the review. It consists of the following steps: (1) identification of the need for a systematic review, (2) development of a review protocol, (3) a comprehensive, exhaustive search for primary studies, (4) quality assessment of included studies, (5) data extraction and monitoring, (6) data analysis and synthesis, and (7) report-writing.

The purpose of this systematic literature review is to analyze and understand how the RE approaches are used for embedded systems development as well as to understand which information regarding embedded systems should be specified by requirements engineers to reduce the gap about what should be considered when performing RE for ES. Thus, we intend to answer the research questions described in Table 1.

Table 1: Research questions and motivations

Research Question	Description and Motivation
RQ1. What phases of the requirements engineering process have been supported?	This question provides a starting point to understand what are the main phases (elicitation, analysis, specification, validation and management) of the requirements engineering process supported by the approaches. It may help to identify which RE phases are attracting more attention to community and which ones deserve a special care.
RQ2. What style of requirements modeling have been supported by the approaches?	The answer to this question allows the identification of main styles of requirements modeling (scenario-based, goal-oriented, textual requirements and so on) that have been supported by the approaches. It may help to identify which requirements styles are attracting more attention to embedded systems community and the strengths and weakness of each style in the embedded systems domain.
RQ3. What type of requirements have been supported by the approaches?	This question intends to identify what is the distribution of the studies with respect to the type of requirements (functional and/or non-functional) addressed. It is important to investigate whether requirements engineering approaches for embedded systems have been used to manage both functional and non-functional requirements.
RQ4. What challenges/problems are identified in research literature relating to RE for ES?	The purpose of this question is to identify the open problems reported by the studies. This information is useful to identify gaps in current research and suggest areas for further investigation.

The SLR was motivated by the need of specific RE processes for embedded systems desired by academia and industry as reported in many studies [3], [16],

[17], [19], [21], [22].The gap that exists between the traditional requirements engineering processes, methodologies, tools, and notations and the ones used for embedded systems also contributes to the need of this SLR.

An automatic search was conducted in the electronic databases ACM Digital Library, El Compendex, IEEE Xplore, ISI Web of Science, Science Direct, Scoups, Springer Link, and validated by requirements experts.We selected these libraries because they include high-quality software engineering journals and proceedings of conferences.

The search period starts in 1970 when embedded system was an emergent area. The search period finishes in June 2019. The search was executed in title, keywords, and abstract based on terms presented in Table 2.

Table 2: Terms of the search.

#	Related words
(T1)	"requirements engineering" OR "requirements elicitation" OR "requirements specification" OR "requirements management" OR "requirements validation" OR "requirements verification" OR "requirements eduction"
(T2)	"requirements modeling" OR "requirements modelling"
(T3)	"embedded systems" OR "safety critical systems" OR "real time systems" OR "embedded software" OR "embedded product"
(T4)	"approach" OR "technique" OR "framework" OR "processes" OR "methods" OR "tool"

In this paper, we want to collect information about requirements engineering for embedded systems. Therefore, such information can be used by academics and practitioners to improve the requirements process to reduce the risk of undetected errors and deficiencies. Thus, we had focused on terms in RE area, embedded systems, and kind of contribution.

Our procedure for selecting studies consisted of six main steps, as shown in Fig. 1.

Inclusion and exclusion criteria We are interested only in primary studies, that present some contribution to requirements engineering for embedded systems, and that satisfies a minimum quality threshold. The results presented here are important since they take into account the several decades of research about RE for embedded systems.

Secondary studies, short papers ($\leq$ 5 pages), studies that are not related to research questions, non-peer-reviewed studies, duplicated studies (only one copy of each study was included), non-English written papers, studies that do not discuss requirements engineering in embedded systems development, grey literature, redundant paper of same authorship, and ongoing work were considered exclusion criteria.

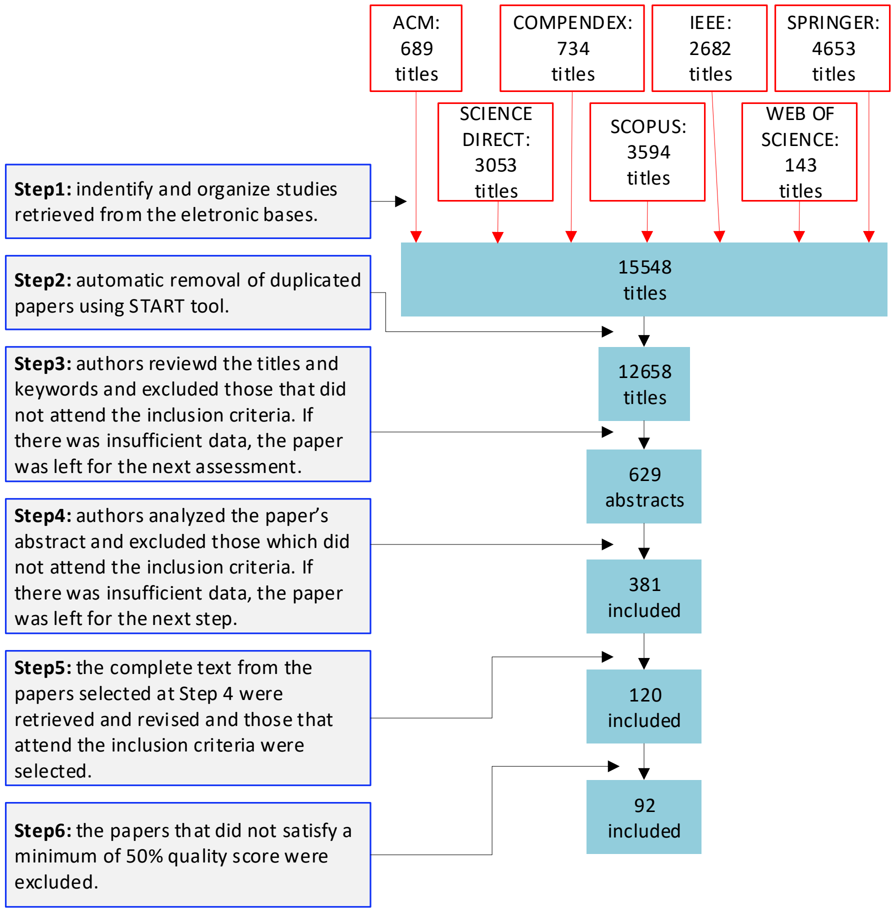

Fig. 1: Systematic literature review steps. Adapted from [12].

Threats to validity We used the four categories of threats presented by [26], which includes threats such as construct, internal, external and conclusion validity threats.

Construct validity: We followed the guidelines provided by [10] to develop a reliable and auditable research protocol. The protocol was validated employing inspection and comparison between already published SLR protocols. The search string of an SLR was evaluated several times to avoid the risk of omitting relevant studies.

Internal validity: During data extraction, subjective decisions may have occurred since some papers did not provide a clear description or proper objectives and results. We conducted the SLR process iteratively to try to mitigate the probabilities due to personal bias on study understanding.

External validity: To mitigate external threats, the search was defined after several trial searches and validated with the consensus of all authors.

Conclusion validity: The methodology chosen in [11] already considers that not all relevant primary studies that exist can be identified. It is possible that some studies excluded in this review could have been included. To mitigate this threat, the selection process and the inclusion and exclusion criteria were carefully designed and discussed by authors to minimize the risk of exclusion of relevant studies.

In order to avoid bias and maximize internal and external validity we performed a quality assessment (QA) of the selected studies. All studies were assessed against a set of 12 quality criteria. The quality score in each criterion of the selected studies are available at the following [link]. We considered 50% as minimal score for a study to accepted. The overall quality of the selected studies is acceptable since the mean of quality is 66.32%.

4 Results and analysis

In this section, we present the results and analysis of the selected studies.

Publication year The analyzed studies in this review were published between 1980 and June 2019. Figure 2 shows that, from 1980 until 2004, the number of publications was almost continuous. We highlight the year 2015, with a total of eleven publications. From 2004 until 2014, we can observe a gradual increase in the number of publications.

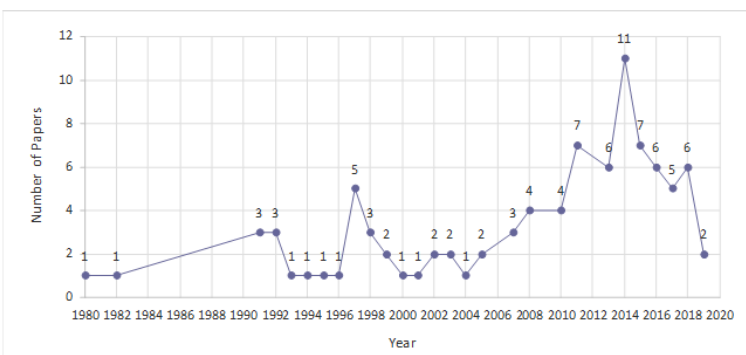

Fig. 2: Temporal view of the studies.

Application context The academic and industrial context were considered in this classification. Forty-nine papers (53.26%) are in the academic category and 43 papers (46.73%) in the industry category. Hence, the industrial community of embedded systems is also very interested in investigating methods, processes, and approaches for requirements engineering.

Research method In order to classify the research method of the publications, we rely on [7].

The research methods of the majority of the papers are illustrative scenarios (49 papers, 53.26%), followed by case study (37 papers, 40.21%), controlled experiment (3 papers, 3.26%), quasi-experiment (1 paper, 1.08%), survey (1 paper, 1.08%), and not applicable (1 paper, 1.08%). Ethnography and action research were not found in our classification. We can observe that research in the field of RE for embedded systems is focused on small examples and case studies to evaluate contributions while the remaining methods are put aside.

4.1 *RQ1. What phases of the requirements engineering process have been supported?*

The phases taken into account to answer this question were defined according to the RE process established by [13]: elicitation, analysis and negotiation, specification, validation, and management. The predominant phase that we identified was *Specification* (76.09 %), followed by *Validation* (42.39%), *Analysis and Negotiation* (36.96%), *Elicitation* (27.17%), and *Management* (17.39%). It is important to note that a study could have addressed more than one phase of the RE process.

Requirements style	Count	%
Scenario-based	25	27.2%
Formal language	22	23.9%
UML	21	22.8%
Textual requirements	18	19.6%
Non-specific	12	13.0%
Description logic	8	8.7%
Feature models	1	1.1%
Problem frames	1	1.1%

Table 3: Requirements modeling style

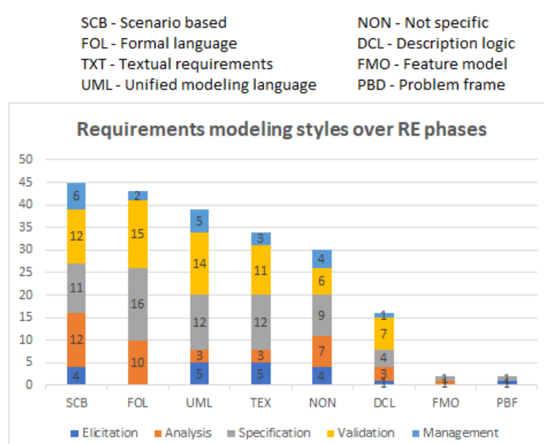

Fig. 3: Requirements modeling styles over RE phases.

An interesting finding is that 28 studies (30.4% of papers included) met both *Specification* and *Validation* phases in the same paper, indicating the interest to specify and validate the ES requirements.

4.2 *RQ2. What style of requirements modeling have been supported by the approaches?*

In order to guide our classification, we used seven requirements style presented in the work of [7], except for *Description logic* since it was discovered during the classification. The results presented in Table 3 were defined according to the distribution of the studies.

Scenario-based category is the most frequent type of RE modeling addressed by the studies. This category included studies on all RE phases, but the majority of studies are concerned with requirements specification, analysis, and validation of requirements, as can be seen in Fig. 3.

4.3 *RQ3. What type of requirements have been supported by the approaches?*

The purpose of this research question was to identify what is the distribution of the studies with respect to the type of requirements (functional and/or non-functional) addressed. The categories we used were: (1) functional requirements, (2) non-functional requirements, and (3) both types. Table 4 presents the distribution of the studies within these categories. As expected, the majority of studies (80.4%) addressed functional requirements. Interestingly, (39.1%) of studies considered both functional and non-functional requirements in the same study, indicating their concern with a more complete requirements specification.

Table 4: Functional and non-functional requirements distribution

Type of requirements	Count	%
Functional req. only	38	41.3
Non-functional req. only	18	19.6
Both	36	39.1

4.4 *RQ4. What challenges / problems are identified in research literature relating RE for ES?*

The purpose of this question is to identify further research needed in this area. These challenges/problems were extracted from the selected studies, and they are presented in the [link]. We focused on specific RE problems for ES. The problems of the studies that fit in *general purpose problems* (32.6%, 30 studies) were left aside. Besides, many studies presented just their proposals, but did not discuss challenges/problems on RE for ES. This corresponds to 31.5% of the studies (29 studies).

One important finding is that eight challenges (33.2%) are related to non-functional requirements for ES (O3, O5, O6, O8, O9, O13, O18, and O21). There are 11 (11.9%) studies highlighting the need to handle quality requirements such as timing, safety, reliability, performance, user interface, and others, indicating that NFRs need further investigation. It should be noticed that safety is related to five challenges and timing to 3 problems.

Seven studies are concerned with the specification phase of the RE process. The challenges O3, O4, O9, O10, and O18 point a concern to investigate the specification of quality requirements, environmental assumptions, and scenarios for embedded systems. Considering that 58 of the studies did not cite any tool support, the finding of *O5 and O14* might indicate the need of development of tools capable of verifying timing requirements, and dealing with different domain tools. Another important aspect is the need to investigate requirements standards in different domains. Moreover, one study argue the need to enhance the IEEE Std 830 [9] for automotive systems.

The most cited challenge/problem is *Apply the proposed study in industry embedded systems project (O1).* This challenge was referenced in 6 studies (6.5%), and it is the consequence of the low number of proposals evaluated in the industrial context (46.7%). These results show the need of applying the proposal in practice with real users to assess the extension of the contributions. *O2 - Apply the proposed approach in different domains* is the second most cited challenge. The goal is to test the proposed solution in different domains, expanding the examples and checking the effects [25,6,23].

In sequel, we have *O3 - Specification of timing requirements*, *O4 - Refining requirements into specifications taking the environmental assumptions into account*, *O5 - Timing requirements verification and tool support development*, and *O6 - Analysis of hazard and threats, timing, performance, and safety.* Timing requirements and environmental assumptions are necessary for the correct operation of embedded real-time systems [1].

ISO 26262 is an international standard for functional safety of road vehicles. This standard can be used to *Improve the development process for ensuring functional safety requirements (O7).* It is possible to deal with performance analysis for specification of NFRs in the *Handling of non-functional requirements such as QoS, safety, reliability, resource and scheduling properties (O8).*

Investigations are also necessary to propose mechanisms to perform the *Specification of safety requirements (O9).* A possible approach is to derive safety requirements from the results of a systematic analysis of the system. This derivation can be done by formalizing the results of fault tree analysis. Multiprocessor based environments and case maps notation can be used in the *Specification and analysis of scenarios for embedded systems (O10)*

Difficulties are also faced to *Enhance the IEEE Std 830 and establish a reference Software Requirements Specification (SRS) for automotive system (O11).* Considering that there are many different SRS for the automotive system, the development of a reference template is a substantial challenge. Introducing hardware at the beginning of the verification infrastructure and considering the use of different interaction paradigms can support the *Hardware verification and modeling of synchronous and asynchronous components (O12)* respectively.

Measuring requirements stability and reusability in embedded systems domain (O13) is a challenge. The goal is to define a set of criteria for measuring requirements stability and its reusability. A possible solution may be the use of the Requirements Maturity Model (RMM) to evaluate the requirement reuse process.

The contributions presented (see the table with the complete list of contributions in the [link]) can be useful in different contexts. For example, a newcomer (e.g., new student) will be able to identify new research opportunities, and they can become the subject of new research projects.

5 Conclusions

In this paper, we presented a systematic literature review that investigates studies reporting approaches proposed to elicit, analyze, specify, validate and manage requirements for embedded systems. Our SLR draws on 92 studies, selected out of 12658 over 49 years, through a multi-stage process. An essential aspect of this review is that it does not restrict itself to a particular domain. Thus, the search gives us deeper insights on the state-of-the-art about the content of requirements engineering for embedded systems.

The most relevant findings from this review and their implications for further research are as follows:

Need to include all requirements engineering phases. The majority of studies only partially address the RE process. In fact, only two studies considered all five phases and only one addressed four phases of the RE process. This indicates that only part of the RE process is met by the studies included in this review, demonstrating that there is a lack of approaches that consider the whole RE process.

Lack of a specific requirements engineering process for embedded systems. The embedded systems community does not present a well-defined, standardized, and known requirements engineering process to guide companies. Hence, there is a need to investigate and develop a specific requirements engineering process by providing inputs, outputs, and work products for correct requirements development and management for the scope of embedded systems.

Need to improve the specification and analysis of timing requirements. Timing requirements are necessary for the correct operation of embedded real-time systems [2]. Hence, the requirements engineering for ES should provide guidelines for the specification, analysis, and verification of timing requirements [5,27].

Need to improve safety requirements. It should be noticed that specification of *safety* requirements have been treated by some approaches, as depicted in question RQ4. However, *safety* is related to five challenges (20.8%), which involves specification, analysis, and verification activities. For example, the requirements specification must be suitable to derive safety requirements Therefore, RE approaches for ES need to provide guidance of safety-related concerns.

Need to improve the specification and analysis of the NFRs. Despite the current contributions to ameliorate the non-functional requirements problem, there are several open issues to be solved. Handling of multiple NFRs such as QoS, safety, and reliability is pointed out by [15] as future work. The difficulty is to select the best possible system configuration based on a set of non-functional requirements. Therefore, RE approaches for embedded systems need to provide a significantly improved account of NFRs concerns.

Need of integration tools. The development of embedded systems requires stakeholders from different expertise. Thus, a tool should be able to organize the specification in a structured manner into several abstraction layers (additional views/expertise). Additionally, it is necessary to develop interfaces to support migrating from well-known requirements management tools such as Doors and RequisitePro. Moreover, the tools should be capable of dealing with elicitation, analysis, specification, validation, and management of embedded systems requirements.

Need to consider RE standards. There are different requirements engineering standards such as IEEE Std 29148:2011 and IEEE Std 830:1998. Nevertheless, RE approaches for ES are not using what has already been done to develop specific RE processes for ES domain. The RE standards provide tasks, activities, practices, goals, and work products that can be tailored to accomplish the RE needs for a specific ES domain.

Need for industry validation. Despite the fact that the industrial community of embedded systems is investigating requirements engineering (46.7% of the studies were applied in an industrial context), there is still a need for implementing the studies in real industry projects with real users in order to improve the relevance of the research and to assess to what extent the approach contributes to the RE for embedded systems.

Need to apply the proposed solution in different domains. Fourteen studies (15.2%) applied their solutions in more than one domain. This fact may indicate the the need to test with the appropriate indicators, the proposed solution in different domains, expanding the examples and checking the effects [25,6,23]. Hence, it may be necessary to evaluate in a second domain, a solution specially developed for a specific area.

Motivated by the results of this SLR we propose a research agenda for the RE community for embedded systems:

1. How can we develop a RE process to address elicitation, analysis, specification, validation, and management phases?
2. What is the core set of information that should be considered by requirements engineers in the development of embedded systems?
3. What are the main requirements engineering patterns and how they can be used in a RE process for the domain of embedded systems?

4. What are the requirements and constraints need to develop a tool capable of dealing with integration with other tools and that supports all five RE phases?
5. How to measure the feasibility of using requirements engineering approaches in embedded systems?
6. What are the primary non-functional requirements and how they are related to embedded systems specificities?
7. How to improve the maturity level of requirements engineering processes for embedded systems?

6 Acknowledgments

We would like to acknowledge that this work was partially supported by CNPq, FACEPE, and CAPES.

References

1. Blouin, D., Senn, E., Turki, S.: Defining an annex language to the architecture analysis and design language for requirements engineering activities support. In: Model-Driven Requirements Engineering Workshop (MoDRE), 2011. pp. 11–20. IEEE (2011)
2. Blouin, D., Senn, E., Turki, S.: Defining an annex language to the architecture analysis and design language for requirements engineering activities support. In: Model-Driven Requirements Engineering Workshop (MoDRE), 2011. pp. 11–20. IEEE (2011)
3. Braun, P., Broy, M., Houdek, F., Kirchmayr, M., M¨uller, M., Penzenstadler, B., Pohl, Weyer, T.: Guiding requirements engineering for software-intensive embedded systems in the automotive industry. Computer Science-Research and Development **29**(1), 21–43 (2014)
4. Broy, M., Stauner, T.: Requirements engineering for embedded systems. Informationstechnik und Technische Informatik **41**, 7–11 (1999)
5. Chen, X., Liu, J., Mallet, F., Jin, Z.: Modeling timing requirements in problem frames using ccsl. In: Software Engineering Conference (APSEC), 2011 18th Asia Pacific. pp. 381–388. IEEE (2011)
6. Damak, Y., Jankovic, M., Leroy, Y., Yannou, B.: Analysis of safety requirements evolution in the transition of land transportation systems toward autonomy. In: 15th International Design Conference-DESIGN 2018 (2018)
7. Dermeval, D., Vilela, J., Bittencourt, I.I., Castro, J., Isotani, S., Brito, P., Silva, A.: Applications of ontologies in requirements engineering: a systematic review of the literature. Requirements Engineering pp. 1–33 (2015)
8. Umm-e Habiba, S.A.K., Javed, M.Y.: Gap analysis in software engineering process adoption in implementing high end embedded system design. J. Appl. Environ. Biol. Sci **4**(7S), 495–503 (2014)
9. IEEE: Ieee recommended practice for software requirements specifications. IEEE Std 830-1998 pp. 1–40 (Oct 1998). https://doi.org/10.1109/IEEESTD.1998.88286
10. Keele, S.: Guidelines for performing systematic literature reviews in software engineering. In: Technical report, Ver. 2.3 EBSE Technical Report. EBSE (2007)

11. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering (2007)
12. Kitchenham, B., Brereton, O.P., Budgen, D., Turner, M., Bailey, J., Linkman, S.: Systematic literature reviews in software engineering–a systematic literature review. Information and software technology **51**(1), 7–15 (2009)
13. Kotonya, G., Sommerville, I.: Requirements engineering: processes and techniques. Wiley Publishing (1998)
14. Lee, E.A., Seshia, S.A.: Introduction to embedded systems: A cyber-physical systems approach. Lee & Seshia (2011)
15. Liu, X., Wang, Z.: Extending east-adl2 to support aspectual requirement specification and analysis for automotive software. In: Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on. pp. 1255–1263. IEEE (2011)
16. Martins, L.E.G., de Oliveira, T.: A case study using a protocol to derive safety functional requirements from fault tree analysis. In: 2014 IEEE 22nd International Requirements Engineering Conference (RE). pp. 412–419. IEEE (2014)
17. Martins, L.E.G., Ossada, J.C., Belgamo, A., Ranieri, B.S.: Requirements elicitation guide for embedded systems: An industry challenge. In: ICSEA 2013 : The Eighth International Conference on Software Engineering Advances (2013)
18. Naumchev, A., Meyer, B., Mazzara, M., Galinier, F., Bruel, J.M., Ebersold, S.: Autoreq: Expressing and verifying requirements for control systems. Journal of Computer Languages **51**, 131–142 (2019)
19. Ossada, J.: Gerse: Guia de elicitação de requisitos para sistemas embarcados de pequeno e médio porte. Dissertação de Mestrado do Programa de Pós-Graduação em Ciência da Computação–Universidade Metodista de Piracicaba-Piracicaba (2010)
20. Pereira, T., Albuquerque, D., Sousa, A., Alencar, F.M.R., Castro, J.: Retrospective and trends in requirements engineering for embedded systems: A systematic literature review. In: Proceedings of the XX Iberoamerican Conference on Software Engineering, XX Workshop on Requirements Engineering (WER 2017). Buenos Aires, Argentina, May 22-23. pp. 427–440 (2017)
21. Sikora, E., Tenbergen, B., Pohl, K.: Industry needs and research directions in requirements engineering for embedded systems. Requirements Engineering **17**(1), 57–78 (2012)
22. Sousa, A., Agra, C., Melo, J., Alencar, F.: Elicitação e especificação de requisitos em sistemas embarcados: Uma revisão sistemática (in portuguese). requirements elicitation and specification for embedded systems: a systematic literature review. Workshop on Requirements Engineering: Lima, Peru (2015)
23. Span, M., Mailloux, L.O., Mills, R.F., Young, W.: Conceptual systems security requirements analysis: Aerial refueling case study. IEEE Access **6**, 46668–46682 (2018)
24. Teufl, S., Khalil, M., Mou, D.: Requirements for a model-based requirements engineering tool for embedded systems: Systematic literature review and survey. White paper. fortiss GmbH (2013)
25. Vilela, J.F.F., Castro, J., Martins, L.E.G., Gorschek, T.: Safety practices in requirements engineering: The uni-repm safety module. IEEE Transactions on Software Engineering pp. 1–1 (2018). https://doi.org/10.1109/TSE.2018.2846576
26. Wohlin, C., Runeson, P., Host, M., Ohlsson, M., Regnell, B., Wesslen, A.: Experimentation in software engineering: an introduction. 2000 (2000)
27. Zafar, S., Dromey, R.G.: Integrating safety and security requirements into design of an embedded system. In: Software Engineering Conference, 2005. APSEC'05. 12th Asia-Pacific. pp. 8–pp. IEEE (2005)