Retrospective and Trends in Requirements Engineering for Embedded Systems: A Systematic Literature Review

Tarcísio Pereira[12], Deivson Albuquerque[12], Aêda Sousa[1], Fernanda Alencar[1], and Jaelson Castro[1]

Federal University of Pernambuco - UFPE[1]
{tcp,mdac,amcsb,jbc}@cin.ufpe.br,fernanda.ralencar@ufpe.br
Federal Institute of Sertão Pernambucano - IF-Sertão[2]

Abstract. In the embedded systems (ES) area, more than 50% of problems occur at system delivery and are related to misconceptions in capturing requirements. Also, requirements engineering (RE) is crucial to meet time, cost, and quality goals. An important step to improve the RE approaches for ES is to gain a detailed understanding of the retrospective and trends presented by the literature. We have conducted a systematic literature review to gain an in-depth understanding of trends and needs concerning RE research. We report on the main results of our study related to three research questions: what requirements should be considered during ES development? what are the RE contributions for ES? and what challenges/problems are identified in the research literature to RE for ES? Based on the results of the study, we draw conclusions for future RE research.

Keywords: requirements engineering · embedded systems · trends.

1 Introduction

Embedded Systems (ES) are present in different domains such as automation technology, automotive, avionics, energy technology or medical technology. The majority of ES are less visible, and they run in engines, brakes, seat belts, airbag, and audio system in your car. In addition, they command robots on a factory floor, power generation in a power plant, processes in a chemical plant, and traffic lights in a city [21]. Broy [7] defines Embedded System 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.

The requirements for software products and systems are gathered, analyzed, documented, and managed through the Requirements Engineering (RE) process. It focuses on the development of processes, methods, models, techniques, and tools to help in the conception of software and systems, covering the activities of elicitation, analysis, specification, validation, and management of requirements [32].

Embedded systems are known for their high complexity, caused by the increasing number of functions and the growing number of interactions among different functions. According to [7], in the embedded system domain, more than 50% of the problems occur when the system is delivered. However, many problems are not related to the correctness of implementation but with misconceptions in capturing requirements. Besides that, companies face the challenge of market pressure. They have to produce high-quality and innovative products in short time. Hence, requirements engineering process is crucial to meet time, cost, and quality goals [30].

This paper aims to provide insights into trends and needs in RE for ES research. To this end, we have analyzed the results of a Systematic Literature Review (SLR) which included studies from 1970 to September 2016, resulting initially in 11.403 papers, from which we classified and analyzed evidence from 75 studies. The SLR addressed the following questions: (1) What requirements should be considered during embedded systems development? (2) What are the requirements engineering contributions for embedded systems? And (3) What challenges/problems are identified in research literature relating to RE for ES? The results of this paper are a step towards developing a body of knowledge in RE for ES necessary to be handled to requirements engineering researchers in the development of a precise RE for ES.

The remainder of this paper is organized as follows: Section 2 discusses related works. The SLR method used in this research is presented in Section 3. Section 4 presents the results of the review with a discussion and analysis for each research question. Threats to validity are discussed in Section 5. Finally, Section 6 summarizes our conclusions.

2 Related Work

Requirements engineering for embedded systems is a research area addressed by many studies. However, we identified only one secondary study [33] that synthesize RE in embedded systems domain.

A previous SLR on requirements elicitation and specification for embedded systems proposed by Sousa [33] differs from ours by means of a time interval, RE activities, the number of databases, and research questions. Our SLR considered studies from the last 46 years. It took into account all activities of the RE process and included seven databases in the studies selection, while [33] considered studies from 2000 to 2014 and included those related to elicitation and specification activities. Regarding the databases, six were used. It is important to highlight that our research questions are different from those proposed in [33].

In order to gain an in-depth understanding of practitioners needs concerning RE research and method development, Sikora [30] conducted an industrial survey to investigate what were the current industry needs concerning method support for requirements engineering in the ES domain. They conducted the study with seven large companies from five different ES domain. They also highlighted the need for a more intensive use of models in RE, a method support for abstraction layers, and the need to consider safety engineering concerns in RE approaches.

3 Systematic Literature Review

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 [16]. To perform this SLR, we used guidelines and the protocol template proposed by Kitchenham and Charters [15], 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. We used a tool called StArt (State of the Art through Systematic Review) [19] to support the application of the SLR.

3.1 Source Selection and Search

The search string was defined according to the research question and relevant papers about the domain to be studied. Table 1 presents the search string used. The string was applied according to the format required by the following electronic databases searched: ScienceDirect, ISI Web of Science, Scopus, SpringerLink, ACM Digital Library, IEEE Xplore, and Compendex. Figure 1 depicts the systematic review process and the number of papers identified at each step.

Table 1. Search String

Search string
(”requirements engineering” OR ”requirements elicitation” OR ”requirements specification” OR ”requirements management” OR ”requirements validation” OR ”requirements verification” OR ”requirements eduction” OR (”requirements modeling” OR ”requirements modelling”)) AND (”embedded systems” OR ”safety critical systems” OR ”real time systems” OR ”embedded software” OR ”embedded product”) AND (”approach” OR ”technique” OR ”framework” OR ”processes” OR ”methods” OR ”tool”)

The papers we selected to be part of the review are primary studies, peer-reviewed studies, and studies that discuss requirements engineering in embedded systems development, published since 1970 until September 2016 and satisfy the minimum quality threshold (50%). Due to the lack of space, the quality assessment (QA) process and results are omitted in this paper[1]. It is important to note that the SLR references with its respective IDs are listed in `http://bit.ly/2kAYp6F`. Thus, the reader can consult the list to identify the studies represented in Tables 2 and 3.

4 Key Results of the Study

This section reports on the findings of the investigation for each research question as defined in Section 1.

[1] The QA results can be found at `http://bit.ly/2otluKz`

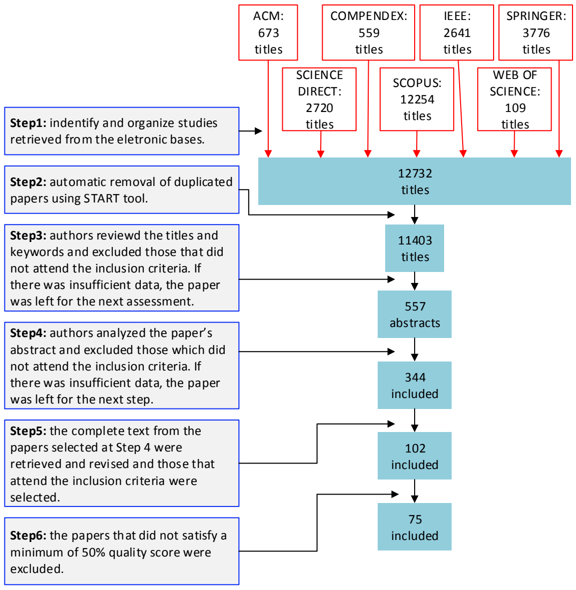

Fig. 1. Systematic literature review process. Adapted from [9].

4.1 Requirements for Embedded Systems Development

This section aims to answer the following research question *"What requirements should be considered during embedded systems development?"*. The identification we performed to answer this question was based on the reading of the studies.

After the requirements identification, we classified them into the four levels of our Requirements Engineering Process for Embedded Systems (REPES). The REPES process is under development [25], and we created the levels to divide the requirements engineering tasks in four views. The description of the levels is depicted below.

Our research on the requirements in systematic literature review studies has resulted on the identification of 56 requirements. These 56 requirements are presented in Figure 2. According to our classification, the majority of requirements (39, 70%) belongs to the requirements level, followed by enterprise level (9 requirements, 16%), system level (5 requirements, 9%), and context level (3 requirements, 5%).

Requirements on Enterprise Level The requirements classified at this level are related to the stakeholders involved in the embedded systems development, the standards, policies, and laws that should guide the development process, the execution environment of the ES, and relationships with suppliers and manufacturers.

The handling of multiple stakeholders is a problematic requirement. Requirements engineering for embedded systems is a multi-disciplinary approach. It

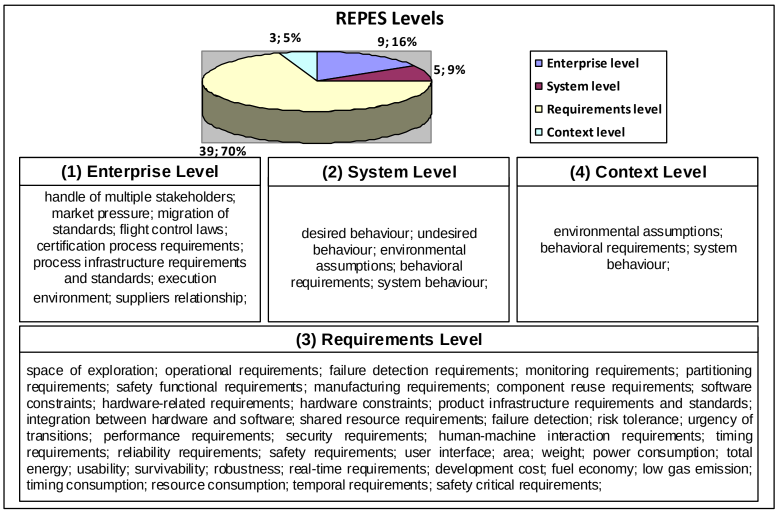

Fig. 2. Requirements classification into the REPES levels.

requires domain experts from several areas, such as mechanical engineers for physical context, electrical engineers for the hardware context, as well as Human-Machine-Interface (HMI) experts for the usability aspects [17]. The market pressure involves the consumer changes, environmental, and regulatory needs. All of these can impact the requirements engineering for ES since it is necessary to move fast and change requirements to facilitate the agility and flexibility needs of its container [17].

Certification process requirements aim the application of a particular development process, guidelines, documentation, and steps for the product under development. These issues will depend on the embedded system domain, such as avionics, automotive, industrial automation, and medical. Process infrastructure requirements and standards are related to the way the results of RE activities and tasks will be documented, managed and which development tools must be used [6].

Execution environment requirements should capture the essential properties of the environment such as temperature, humidity, the area of operation, and electric current in which the embedded system under development will operate [11]. Migration of standards relies on requirements that depict the migration towards more complicated/sophisticated standards based on their availability [20] [10]. It is important to note that migration of standards is closely related to the certification process requirements. Finally, suppliers relationship requirements are based on existing commercial off-the-shelf (COTS) [14].

Requirements on System Level The requirements classified at this level are related to the system behavior, the environment required by the system and the functionalities that the system is supposed to perform.

Desired behavior and system behavior relies on customer wishes to the final embedded product [13]. In this way, customer expectations should be managed as well as the undesired behavior, the system reactions that should be avoided. Thus, it is possible to find different ways to treat them. Environmental assumptions (expectations) are conditions that the system environment must accomplish for correct system operation [4].

Incorrect environmental assumptions can be the cause of several system failures, such as the number of architectural elements (e.g., control variables). Correct assumptions can benefit system reuse and integration [4]. Behavioral requirements should determine the behavior of a software system and the services offered by it [6].

Requirements on Requirements Level The requirements classified at this level are related to functional and non-functional requirements, hardware requirements, software and hardware constraints, integration between hardware and software, manufacturer requirements, and product infrastructure requirements and standards.

Failure detection and monitoring requirements aim the identification and definition of which kinds of failures must be detected (failure detection mechanisms/functions) or to what extent correct functionality must be monitored [6]. Sophisticated techniques have to be used to find good design solutions using the space of exploration [31]. Space of exploration is a space of possible solution, and the aim is to identify the best functional requirements configuration considering the non-functional requirements. A big challenge in ES domain is to manage conflicting NFRs (e.g. cost and safety).

Partitioning is related to fault containment requirements. The goal of such requirements is to ensure that a failure in one partition must not propagate to cause failure in another partition [29]. In order to maintain the system in a safe state, it is necessary to describe what actions and/or constraints should or should not be performed, the so-called safety functional requirements [23]. Manufacturing requirements aim the production of relevant information/documentation for fabricating the product [27]. Another challenge is the requirements specification for component reuse. Maintenance and integration issues may cause compatibility problems, mainly with legacy code developed for the original core component model [18].

Embedded systems encompass a software/hardware-constrained computing. These constraints are related to cost, size, CPU, memory, energy, etc. The system needs to be able to deal with its computation resources efficiently [12]. Hardware-related requirements should define the physical devices (input and output devices, e.g. interaction displays), interface characteristics (e.g. external communication interface), voltage component requirements (e.g. ac/dc adapter), etc [24]. Product infrastructure requirements and standards aim the definition of which implementation technology (programming language, code pattern) must be used or which standard components have to be re-used [6].

Integration between hardware and software relies on the system integration, which verifies the interaction among software components and the communication interfaces between the software and hardware [27]. Shared resource requirements should define the resources that are shared by several components (e.g. electric motor), they are commonly used to detect collisions among sharing components in the system [2].

Our investigation resulted in the identification of 22 non-functional requirements. Generally these NFRs consist of quality attributes (e.g. performance, reliability, survivability, robustness, safety, safety critical, security, real-time, temporal); development constraints (e.g. timing, area, weight, power consumption, total energy, cost, fuel economy, low gas emission, timing consumption, resource consumption), and external interfaces requirements (e.g. user interface, usability, human-machine interaction).

Requirements on Context Level Requirements at this level are related to environmental assumptions, behavioral requirements, and system behavior (discussed in section 4.1). According to Ali (2010) [1], a context is a partial state of the world that is relevant to an actor's goal. Therefore, we should analyze the requirements cited before to identify and specify contexts that could affect the system execution. Requirements at this level must describe the system execution, the place in which the context is localized and the available resources to use [26]. The description of these items can be used to model and specify contexts.

The above contributions may be useful in different contexts. For example, (i) during the definition of a requirements engineering process for embedded systems, (ii) the contributions can provide insights for discussions to decide the issues that will be considered during the ES development, (iii) to understand the requirements involved in ES domain, and (iv) it can provide an attempt to separate the requirements concerns at the levels of the REPES process.

4.2 Requirements Engineering Contributions for Embedded Systems

This section aims to answer the following research question *"What are the requirements engineering contributions for embedded systems?"*. Thus, we can perform a retrospective on the RE problems considered by the community. The classification we made to answer this question was based on the analysis of RE problems addressed by the studies. The results of this question are depicted in Table 2. Due to the lack of space, we will not provide descriptions for all RE contributions illustrated in Table 2.

We focused on specific RE contributions for ES. The contributions of the studies that fit in *general purpose contributions* (32%, 24 studies) were left aside. Besides, many studies did not explicitly discussed the goal of their proposals (13.33%, 10 studies).

The most cited contribution are *Integration between requirements and architecture (P1)* and *Specification of safety requirements (P2)*. They were referenced

Table 2. Requirements engineering contributions.

#	Contributions	Studies
General open issue.		24 studies
It does not cite/Not clear.		10 studies
P1	Integration between requirements and architecture	S07, S38, S60, S74
P2	Specification of safety requirements	S12, S25, S57, S61
P3	Specification of timing requirements	S34, S52, S63
P4	Specification of real-time capabilities	S58, S62, S75
P5	Lack of a well-defined requirements engineering process for embedded systems domain	S09
P6	Handling of multiple stakeholders	S27
P7	Integration between hardware and software	S11
P8	Improved use case for the requirements elicitation and specification of embedded systems	S42
P9	Specification of timing behaviour and operational system properties	S41
P10	Elicitation and analysis of security requirements	S40
P11	Specification of security requirements	S48
P12	Requirements reuse for embedded systems	S36
P13	Verification of timing requirements	S34
P14	Specification of resource requirements	S05
P15	Physical and non-functional requirements in SPL for embedded systems	S68
P16	Modeling of functional and non-functional requirements for the domain of embedded systems	S47
P17	Handling of multiple non-functional requirements on the entire distributed system	S03
P18	Specification of electronic control unit (ECU)	S37
P19	Prototyping embedded systems for requirements validation	S01
P20	System and controller specification	S35
P21	Control of the physical processes of embedded systems	S45
P22	Space of system specification	S06
P23	Detection and correction of behavioral requirements	S44
P24	Integration of safety and security requirements to the overall system	S64
P25	Specification of system behavior and software architectures	S67
P26	Integration of tools used by different stakeholders	S04
P27	High number of elements available for modeling due to different domain experts	S74
P28	Elicitation of trustworthiness requirements	S70
P29	Requirement definition and their verification in the context of distributed embedded system	S22
P30	Lack of a unified framework for requirements engineering of safety critical systems	S69
P31	Avoid manual intervention for analysis of embedded systems	S10

in 4 studies (5.33%) each. The studies of P1 aim at the representation and alignment of functional and/or non-functional requirements to the system architecture. Among the solutions, we can highlight the definition of a domain specific language (S07 and S38), requirements formalization (S60), and SysML extension (S74). These results show the importance to keep requirements engineering activities aligned with the architecture development.

The studies of P2 discuss that the capability to use formal methods (S12), fault tree analysis (S25), observation (S57), and boilerplate notation (S61) might be used to improve safety requirements specification of embedded systems. *Specification of timing requirements (P3)* is a concern mentioned in 3 studies (4%). It is necessary for the correct execution of embedded real-time systems. Component-based approach (S34) and scenarios (S52) can be used to generate time specification, or it is possible to integrate timing constraints in other approaches such as problem frames (S63).

The *Specification of real-time capabilities (P4)* is discussed in 3 studies (4%). The solutions include a design methodology for constraints of embedded real-time system specification (S58), an approach to check the specification of ERTS (S62), and a combination of UML, SysML, and MARTE stereotypes to improve activities of requirements specification (S75). In the automotive industry, requirements engineering processes are not well-defined in most cases. The current practices of RE in the automotive industry are often inadequate to deal with the increasing size and complexity of software-intensive systems [5]. Thus, there is a *Lack of a well-defined requirements engineering process for embedded systems (P5)*.

One of the major problems in the development of embedded systems is the *Integration between hardware and software (P7)*. The goal is to determine whether, on a given hardware platform, the implementation of the requirements specification is feasible. *Specification of electronic control unit (ECU) (P18)*. It is necessary to have an approach to manage the large amount of requirements information to specify the number of functionalities that work in an ECU, to know what is the boundary through which the ECU interacts with its environment, to define how a functionality reacts to the inputs by producing the outputs, and to describe how an ECU exchanges data and control with other ECUs via communication media.

The above contributions may be useful in different contexts. For example, a newcomer (e.g. new student) will be able to identify, study, and understand the main contributions related to requirements engineering for embedded systems.

4.3 Research needs on Requirements Engineering for Embedded Systems

This section aims to answer the following research question *"What challenges/problems are identified in research literature relating to RE for ES?"*. The purpose of this question is to identify further work needed on RE for ES. It was based on the analysis of the future works addressed by the studies, and they are presented in Table 3. Due to the lack of space, we will not discuss all RE challenges depicted in Table 3.

We focused on specific RE problems for ES. The problems of the studies that fit in *general purpose problems* (36%, 27 studies) were left aside. Besides, many studies presented their proposals, but they did not discuss challenges/problems on RE for ES. This corresponds to 33.33% of the studies (25 studies).

The most cited challenge/problem is *Apply the proposed study in industry embedded systems project (O1)*. This challenge was referenced in 4 studies (5.33%), and it is the consequence of the low number of proposals evaluated in the industrial context (46.6%). These results show the need of applying the proposal in practice with real users to assess the extension of the contributions.

Specification of timing requirements (O2) and *Refining requirements into specifications taking the environmental assumptions into account (O3)* are the second most cited challenges/problems (2.66%). Timing requirements and environmental assumptions are necessary for the correct operation of embedded

Table 3. Challenges/Problems on requirements engineering for ES.

#	Research directions	Studies
General open issue.		27 studies
It does not cite/Not clear.		25 studies
O1	Apply the proposed approach in industry embedded systems project	S04, S09, S31, S59
O2	Specification of timing requirements	S01, S20
O3	Refining requirements into specifications taking the environmental assumptions into account	S07, S21
O4	Improve the development process for ensure functional safety requirements	S41
O5	Handling of non-functional requirements such as QoS, safety, reliability, resource and scheduling properties	S37
O6	Specification of safety requirements	S45
O7	Specification and analysis of scenarios for embedded systems	S52
O8	Enhance the IEEE Std 830 and establish a reference Software Requirements Specification for automotive system	S72
O9	Hardware verification and modeling of synchronous and asynchronous components	S22
O10	Measuring requirements stability and reusability in embedded systems domain	S36
O11	Analysis of hazard and threats, timing, performance, and safety	S64
O12	Integration of requirements management tools	S07
O13	Simulation execution of practical real-time software in a visual way	S62
O14	Modeling of urgency	S10
O15	Evolution of behavioral requirements and functional design	S44
O16	Timing requirements verification and tool support development	S63
O17	Specification of user interface for embedded systems	S20
O18	Variability in embedded systems development	S47
O19	Complete the language implementation; validate the language considering 244 FR; transformation of ReSA requirements to formal requirements	S71

real-time systems [4]. S21 expects to look at how environment entities could be modeled using Colored Petri Nets(CPN) representations.

ISO 26262 is an international standard for functional safety of road vehicles. This ISO can be used to *Improve the development process for ensuring functional safety requirements (O4)*. *Handling of non-functional requirements such as QoS, safety, reliability, resource and scheduling properties O5* is pointed out by Liu et al. [22]. With NFRs parameters, it is possible to deal with performance analysis for specification.

Investigations are also necessary to propose mechanisms to perform the *Specification of safety requirements O6*. A possible solution may be an investigation 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. Difficulties are also faced up to *Enhance the IEEE Std 830 and establish a reference Software Requirements Specification (SRS) for automotive system O8*. Considering that there are many different SRS for the automotive system, the development of a reference template is a substantial challenge.

S10 argue that a model checking tool would allow the direct description of the *Modeling of urgency O16*. When the functional design of an ES is changed, a potential disorder of the behavioral requirements can be detected and displayed. The *Evolution of behavioral requirements and functional design O17* aims to support the model evolution of them on the fly. *Specification of user interface for embedded systems O19* is a concern that should be considered for embedded systems in an industrial context. Depending on the case, many users interactions can happen. Thus, a good user interface can increase production and reduce costs. *Variability in embedded systems development O22* is a crucial issue to consider. Most consumer electronics contain embedded software, and this dynamic market requires approaches to support fast development and deliver entire families of products.

The above contributions may 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 Threats to Validity

According to Bittencourt et al. [3], bias in the selection of studies and inaccuracy in data extraction are the main threats to validity of systematic reviews. With the aim to avoid these threats, we followed the guidelines provided by Keele [15] to develop a reliable and auditable research protocol. The protocol was validated by means of inspection and comparison between already published SLR protocols.

The search string of a SLR must be evaluated several times to avoid the risk of omitting relevant studies. The risk of this threat highly depends on how well the keyword list is built. To decrease this kind of risk, we took into account the research questions and a well-known set of papers to build the initial taxonomy, which evolved over time. While the protocol was being built, (during a period of 15 days) we executed the search string several times in the digital libraries. In doing so, we identified the weaknesses and provided some refinements in order to obtain better results. We also used seven different search engines to omit the limitations implied by employing a particular one.

During data extraction, subjective decisions may have occurred since some papers did not provide a clear description or proper objectives and results. Therefore, it is possible that we wrongly classified the studies when we applied the inclusion/exclusion criteria or that impartial data extraction may have occurred.

6 Conclusions and Future Work

This paper presents trends and needs in RE for ES research based on the results of a SLR. We pointed out specific requirements that RE researchers should consider (section 4.1), the contributions of the extensive research on RE for embedded systems (section 4.2), and we highlight the challenges/problems regarding RE for embedded systems (section 4.3).

This work may help RE researchers to focus their work on more relevant needs on RE for embedded systems. In addition, the results and conclusions of this work are expected to help practitioners to identify specific requirements that must be considered in embedded systems development and gain awareness of the relevant research results identified in our SLR. The most relevant findings from this review and their implications for further research are as follows.

Need to improve the specification and analysis of timing requirements. Timing requirements is necessary for the correct operation of embedded real-time systems [4]. Hence, the requirements engineering for ES should provide guidelines for the specification, analysis, and verification of timing requirements [8] [34].

Need to improve the specification and analysis of the NFRs. Despite the current contributions to ameliorate the non-functional requirements problem, there is a lot of open issues to be solved. Handling of multiple NFRs such as QoS, safety, and reliability is pointed out by Liu et al. S37 as future work. The difficulty is to select the best possible system configuration on the selection of one or a set of non-functional requirements. In addition, the requirements specification must simply be suitable to derive safety requirements from the results of a systematic analysis of the overall system [28]. 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). In addition, it is necessary to develop interfaces for migrating from well-known requirements management tools such as Doors and RequisitePro.

Need to include RE standards. There are different requirements engineering standards such as IEEE Std 1233:1998 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 necessities for a particular ES domain.

Need to apply the studies on real industry projects. Despite the fact that the industrial community of embedded systems is investigating requirements engineering (46.66% 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 contribute to the RE for embedded systems.

In our future work, we will exploit the study results to enhance a RE process for embedded systems that is being developed in our group (see [25]). The aim is to tailor our research in order to address the challenges/needs identified in this study. We also intend to continue this SLR to answer the following research questions: i) what phases of the requirements engineering process have been supported by the approaches?; ii) what are the domains that the approaches support?; and iii) what style (scenario-based, goal-oriented, etc.) of software requirements modeling have been supported by the studies?.

Acknowledgments. The following Brazilian institutions have supported this work: CNPQ and FACEPE.

References

1 Ali, R., Dalpiaz, F., Giorgini, P.: A goal-based framework for contextual requirements modeling and analysis. Requirements Engineering 15(4), 439–458 (2010)

2 Ben-Abdallah, H., Lee, I., Kim, Y.S.: The integrated specification and analysis of functional, temporal, and resource requirements. In: Proceedings of the Third IEEE International Symposium on Requirements Engineering. pp. 198–209. IEEE (1997)

3 Bittencourt, I.I., Baranauskas, M.C., Pereira, R., Dermeval, D., Isotani, S., Jaques, P.: A systematic review on multi-device inclusive environments. Universal Access in the Information Society pp. 1–36 (2015)

4 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)

5 Braun, P., Broy, M., Houdek, F., Kirchmayr, M., Müller, M., Penzenstadler, B., Pohl, K., Weyer, T.: Guiding requirements engineering for software-intensive embedded systems in the automotive industry. Computer Science-Research and Development 29(1), 21–43 (2014)

6 Broy, M., Slotosch, O.: From requirements to validated embedded systems. In: International Workshop on Embedded Software. pp. 51–65. Springer (2001)

7 Broy, M., Stauner, T.: Requirements engineering for embedded systems. Informationstechnik und Technische Informatik 41, 7–11 (1999)

8 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)

9 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)

10 Doering, D., Pereira, C.E., Denes, P., Joseph, J.: A model driven engineering approach based on aspects for high speed scientific x-rays cameras. In: 16th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC 2013). pp. 1–8. IEEE (2013)

11 Fidge, C.J., Lister, A.M.: Disciplined approach to real-time systems design. Information and Software Technology 34(9), 603–610 (1992)

12 Friedrich, L.: A survey of operating systems infrastructure for embedded systems. In: VI Workshop de Sistemas Operacionais (2009)

13 Gomes, L., Fernandes, J.M., Gomes, L., Fernandes, J.M.: Behavioral Modeling for Embedded Systems and Technologies: Applications for Design and Implementation. Information Science Reference - Imprint of: IGI Publishing, Hershey, PA (2009)

14 Hyysalo, J., Parviainen, P., Tihinen, M.: Collaborative embedded systems development: survey of state of the practice. In: 13th Annual IEEE International Symposium and Workshop on Engineering of Computer-Based Systems (ECBS'06). pp. 9–pp. IEEE (2006)

15 Keele, S.: Guidelines for performing systematic literature reviews in software engineering. In: Technical report, Ver. 2.3 EBSE Technical Report. EBSE (2007)

16 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)
17 Krüger, I., Farcas, C., Farcas, E., Menarini, M.: 7 requirements modeling for embedded realtime systems. In: Model-Based Engineering of Embedded Real-Time Systems, pp. 155–199. Springer (2010)
18 Kuz, I., Liu, Y.: Extending the capabilities of component models for embedded systems. In: International Conference on the Quality of Software Architectures. pp. 182–196. Springer (2007)
19 LAPES: Start of the art thought systematic review tool (2015), `http://lapes.dc.ufscar.br/tools/start_tool`
20 Lecomte, S., Guillouard, S., Moy, C., Leray, P., Soulard, P.: A co-design methodology based on model driven architecture for real time embedded systems. Mathematical and Computer Modelling 53(3), 471–484 (2011)
21 Lee, E.A., Seshia, S.A.: Introduction to embedded systems, a cyber-physical systems approach. UR L http://LeeSeshia. org 6, 18 (2011)
22 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)
23 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)
24 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)
25 Pereira, T., Albuquerque, D., Sousa, A., Alencar, F., Castro, J.: Towards a metamodel for a requirements engineering process of embedded system. In: Brazilian Symposium on Computing Systems Engineering. IEEE (2016)
26 Pereira, T.C.: BVCCoN-Tool: Uma Ferramenta para Apoiar uma Abordagem de Configurao de Processos de Negcio Dinmicos. Master's thesis, Universidade Federal de Pernambuco, Recife, Pernambuco, Brasil (2014)
27 Pries, K.H., Quigley, J.M.: Project Management of Complex and Embedded Systems: Ensuring Product Integrity and Program Quality. CRC Press (2008)
28 Ravn, A.P., Rischel, H., Hansen, K.M.: Specifying and verifying requirements of real-time systems. IEEE Transactions on Software Engineering 19(1), 41–55 (1993)
29 Rushby, J.: Partitioning in avionics architectures: Requirements, mechanisms, and assurance. Tech. rep., DTIC Document (2000)
30 Sikora, E., Tenbergen, B., Pohl, K.: Industry needs and research directions in requirements engineering for embedded systems. Requirements Engineering 17(1), 57–78 (2012)
31 Sokolsky, O., Lee, I., Ben-Abdallah, H.: Specification and analysis of real-time systems with paragon. Annals of Software Engineering 7(1-4), 211–234 (1999)
32 Sommerville, I.: Software Engineering. Addison-Wesley, 9th edition edn. (2011)
33 Sousa, A., Agra, C., Melo, J., Alencar, F.: Elicitação e especificação de requisitos em sistemas embarcados: Uma revisão sistemática. Workshop em Engenharia de Requisitos: Lima, Peru (2015)
34 Zafar, S., Dromey, R.G.: Integrating safety and security requirements into design of an embedded system. In: Proceedings of the 12th Asia-Pacific Software Engineering Conference. pp. 629–636. APSEC '05, IEEE Computer Society, Washington, DC, USA (2005), `http://dx.doi.org/10.1109/APSEC.2005.75`