

A Requirements Definition Framework for the Robotic Systems Domain - An Exploratory Study

Danyllo Albuquerque, Jaelson Castro, Aeda Sousa

Universidade Federal de Pernambuco (UFPE)

danyllo.albuquerque@ufcg.edu.br, jbc@cin.ufpe.br,
aedasousa@gmail.com

Abstract. A robotic system is a physical device composed by special-purpose hardware and software that must be integrated to enable a robotic system to function as expected. The first step towards the modeling robotic systems is to determine its requirements, which is a description of how the system should behave together with constraints on its operation. Much research effort in the last two decades has been invested to support requirements engineering (RE) for general systems. However, to our knowledge, there is no framework to properly support RE for robotic systems. Hence, we conducted an exploratory study aiming to answer the question "What technical features affect the RE for the robotic systems domain?" Based on what we learned in the study, we point out some challenges and issues in RE for the robotic systems domain. Additionally, we provide a framework to identify and manage several technical features which affect RE for this domain.

Keywords: Robotic systems, Requirement Definition, Empirical Study.

1 Introduction

A robotic system is a combination of some parts – *hardware* for system assembling and *software* for system operations – that must be integrated to enable a robotic system to function as expected. Robotic systems are being increasingly integrated in several aspects of everyday life. These systems range from critical mission [1] to infotainment and home service tasks [2]. Robotic systems are expected to assist or replace their human counterparts for efficient and effective performance of all sorts of tasks such as industrial operations [3] or even surgical procedures [4]. In order to support the vision of a robotic-driven world, academic research [1][5], industrial [6], and open source solutions [7] are striving to provide cost-effective and efficient solutions for robotic systems. Researchers [12] and practitioners [6] are increasingly focusing on exploiting software engineering methodologies to abstract complexities and enhance efficiency for modeling, developing, maintaining and evolving robotic systems cost-effectively [8].

The first step towards modeling robotic systems is to define its requirements, which are descriptions of how the system should behave, and the constraints associated with its operation [11]. Normally, the requirement definition occurs through a process called Requirements Engineering (RE). This process refers to the activities to defining, documenting and maintaining the system requirements [11]. Requirement definition for the robotic system has always been a severe challenge mainly due to technological heterogeneity issues involved in this task [13]. Since robotic systems often rely on special-purpose *hardware* and operating *software*, the RE for these systems usually involves both

hardware and *software* requirements. Additionally, the process of bringing intelligence requires tighten capabilities of sensing, processing and acting by robotic systems. In this scenario, requirements definition plays a key role as it is the medium to embody these robotic system functionalities aiming at properly modeling the system.

Since the early 2000's, there has been a continuous stream of reported research on requirement definition techniques for robotic systems. From the complexity of such systems, we infer it requires systematic collection of requirements on levels not considered in general systems [11]. However, we realize most of the requirements definition techniques for robotic systems come from ones which are normally used in general systems [19]. Thus, we believe several technical features (e.g. collaboration with environment, autonomy and decision-making process) related to robotic systems are not properly addressed by the requirement definition techniques in the state-of-the-art. The main question in this research is "*What technical features affect the requirements definition for the robotic systems domain?*"

Aiming at answering the aforementioned question, we used an exploratory study to identify and analyze the existing techniques and research progress that influence requirements definition on the robotic systems. Due to particularities in the robotic systems domain, we have grouped the main technical features in a framework. We consider the framework as an extensible set of cooperating technical features which robotic system engineers should take into consideration aiming at performing properly the requirements definition task. The key contributions of this study are: (i) identifying and exploring the challenges and issues associated with requirements definition considering the robotic systems domain and (ii) providing robotic system engineers with a framework to identify and manage several technical features which affect the requirements definition task. Finally, we believe the results of this study can benefit *Researchers* who are interested in knowing the state-of-the-art of requirements definition for the robotics systems; and *Practitioners* who may be interested in understanding the techniques that are suitable for the requirements definition for this domain.

We organized the remainder of this paper as follows: *Section 2* introduces robotic systems and RE process. *Section 3* explains the research methodology used. *Section 4* describes existing requirement definition techniques for robotic domain. *Section 5* reports the main challenges and issues for requirement definition considering robotic systems. *Section 6* presents the proposed framework for requirements definition for the robotic systems. Finally, *Section 7* show the main validity threats and *Section 8* presents key conclusions and future works from this research.

2 Background

In the following, we describe the main concepts related to this research. We briefly introduce robotic systems as well as their main features in *Section 2.1* whereas requirements engineering is minimally outlined in *Section 2.2*.

2.1 Robotic Systems

Robotic system is a combination of *hardware* and *software* components that can be integrated to build a robotic system [6]. The ISO 8373:2012 standard defines terms used in relation with robots and various robotic devices that operate in industrial and non-industrial environments. The *hardware* components - such as the sensors and robotic arms - enable the assembly of a robot. Hardware components are controlled and manipulated by *Control Layer* that is essentially a collection of drivers (as system specific code) to interact with the hardware as depicted in Fig 1.

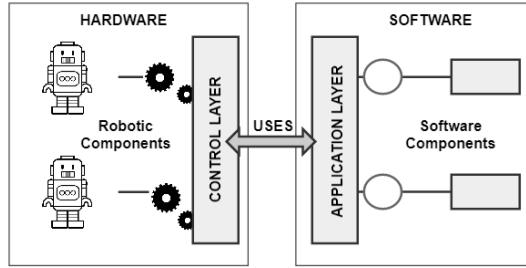


Fig. 1. A reference model for robotic system.

For more complex robotic functions, specialized software is provided for integration and coordination of hardware components to manipulate the robotic behavior. In Fig. 1, this refers to as *Application Layer* that utilizes the *Control Layer* to support robotic operations. For example, considering a home service robot [10], the control layer provides a driver that enables access to a robotic arm. Depending on specific requirements, a software system at application layer must be provided. Such software system is expected to utilize drivers from control layer to enable the movement of arm for home service robot at certain degrees of precision and/or avoiding any obstacles.

Since robotic systems often rely on special-purpose hardware using software components, the first step towards modeling the robotic systems is to define their *software* and *hardware* requirements. The requirements are descriptions of how the system should behave and the constraints associated with its operation [11]. Normally, the requirements definition occurs through a process called Requirements Engineering (RE) minimally outlined in the next section.

2.2 Requirement Engineering

The Requirements Engineering (RE) is a process composed by several phases [11]. The *elicitation* phase involves understanding the application domain, the specific problem to be solved and the specific facilities required by the system stakeholders. The *analysis and negotiation* phase is concerned with the establishment of an agreement related to high-level statement of requirements elicited from stakeholders. The *specification* phase provides a list of agreed set of requirements, which should be complete and consistent. The *validation* phase is the final stage of RE aiming at checking the final draft of a requirements document to certify if it represents an acceptable system description. In addition to these phases, the requirements *management* is the process of managing changes in requirements. The Fig. 2 describes the various phases of the RE process.

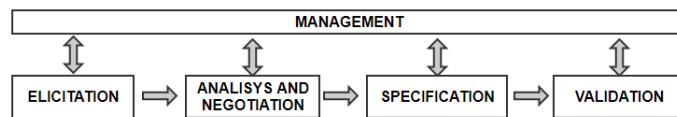


Fig. 2. Requirement engineering phases.

RE is a recognized importance area for the general systems and lots of research effort has been expended in the last decades [11]. There have been significant advances aiming to provide techniques to adequately modeling the general systems [11]. Some studies pointed out requirements definition process for the robotic systems is challenging [18][13]. Several particularities inherent to the domain make this activity even more difficult compared to general systems [13]. There are an increasing number of studies focusing on analyzing techniques for address some RE phases on this domain. However, to extend of our

knowledge, there is no framework in the literature describing the main technical features which affect the requirements definition for the robotic systems.

3 Study Settings

The main question in this research is “*What technical features affect the requirements definition for the robotic systems domain?*”. The quality focus is to facilitate the accomplishment of this task in the context of robotic systems, and thus help robotic system engineers to properly modeling these systems. The context of this study consists of providing benefits to *Researchers* who are interested in knowing the state-of-the-art of requirements definition for the robotics systems; and *Practitioners* who may be interested in understanding the techniques that are suitable for the requirements definition for this domain.

We conducted an exploratory study aiming at address the aforementioned research question. The procedure undertaken involved a set of phases outlined following. The *initial phase* was composed by collection of research papers discussing the idea of software engineering (SE) for robotic systems domain and requirements engineering (RE) for embedded system domain:

SE for robotics: We chose some studies in this area aiming at identifying ES techniques that could be applied - even indirectly - in the requirement definition for robotic systems. First, Oliveira et al [14] have reported a systematic literature review (SLR) of Service-Oriented Development of robotic systems. Their review presents the solutions that support design, development and operation of robotic systems based on software services using service-oriented approaches. Second, Pons et al. [15] have reported another SLR of SE approaches for robotics systems. They highlight some trends of SE techniques for robotic software. The review highlights the application of component based, service oriented as well as model driven development (MDD) for robotics. Finally, Heineck et al [16] investigated how MDD techniques have helped robotic systems domain. They conducted a SLR seeking to identify approaches and their main technical features, as well as the types of specific requirements, behavioral and social issues.

RE for Embedded Systems: We chose studies on this domain because most of the characteristics of Embedded Systems can also be applied to robotic systems [20]. First, Pereira et al [17] conducted a SLR which provides insights into trends and needs in RE for embedded systems. The authors classified and analyzed evidence from 75 studies, developing a body of knowledge in RE for embedded systems. Second, Sousa et al [18] also conducted a literature review focusing on elicitation and specification phases of RE for embedded systems. The authors identified 15 studies, providing a mapping and classification of techniques which provide support to the two considered RE phases. Finally, Marwedel [20] conducted an exploratory study to investigate the main characteristics that affect a modeling and development of embedded systems. In the study was pointed out some components and techniques as well as the main issues and challenges existing for the development of embedded systems.

As a *second phase* a selection of papers in direct relation to RE for robotic systems were chosen, and the provided information was analyzed to derive an understanding. To the extent of our knowledge, the only exploratory study directly related to RE for robotic systems was recently proposed by Albuquerque et al [18]. The authors conducted a SLR providing a classification of techniques, which support the RE phases considering robotic systems. The principles used in managing and evolving requirements in robotic system were identified.

As a *third phase*, we searched for studies describing certain applications with autonomic capabilities and some reports describing autonomic systems [21], [22], [23], [25], [26],

[30], [32], [33]. This phase aimed to identify the technical features related to intelligence in the robotic systems. We believe the description and modeling of these technical features are challenging in the context of robotic systems. It is worthy to mention that in order to collect primary studies in the present research we follow minimally the guidelines provided by 35.

As a *final phase* (a) we pointed out a set of challenges and issues that robotic system engineers must face aiming at defining requirements consistently as well as (b) we proposed a framework that explains to robotic system engineers the main technical features to consider for accommodating requirements definition.

4 Requirement Definition Techniques for Robotic Systems

As previously mentioned, the first step towards modeling the robotic systems is to define its requirements. Requirement definition for the robotic systems has always been a severe challenge compared to general systems [13]. First, since robotic systems often rely on special-purpose hardware and operating software, the requirement definition usually involves both *hardware* and *software* requirements. Additionally, the process of bringing intelligence requires tighten capabilities of sensing, processing and acting by robotic systems.

From the complexity and heterogeneity of such systems, we infer it requires systematic collection of requirements on levels not considered in general systems [11]. However, we realize most of the techniques used to requirements definition for robotic systems come from ones commonly used in general systems [19]. Thus, we believe several technical features directly related to robotic systems domain (e.g. autonomy, collaboration with environment, human and other robots) are not properly addressed by the techniques for requirement definition in the state-of-the-art.

More recently, a study gathered several techniques for requirement definition employed to robotic systems domain [18]. The authors conducted an exploratory study for identifying and analyzing 38 peer-reviewed studies which describe techniques for requirement definition for the robotic systems. To illustrate, a small part of these techniques are showed in Table 1 which describes (i) the RE phase addressed by the study, (ii) a short study description, tools and techniques used in the study as well as (iii) the main findings. We can easily notice there are a wide variety of requirements definition techniques that can be applied for the robotic systems domain.

Table 1. Requirement definition techniques for the robotic systems.

Study	RE phase	Study description	Tools and techniques	Findings
[37]	Elicitation and Specification	The proposed Autonomy Requirement Engineering (ARE) uses special Generic Autonomy Requirements (GAR) model to specify requirements with a proper formal notation.	Goal Oriented RE (GORE) and (ARE)	The approach helps designers to identify and record the autonomy requirements of a system in the form of special self-i* objectives and other assistive requirements.
[38]	Elicitation	This study proposes a visual tool for systematically mapping the customers' requirements and the functional requirements.	Social Modeling and Questionnaire	The contribution of this paper is the construction of the standard Quality Function Deployment (QFD) chart to guide the design revisions of the robot.
[39]	Specification and Validation	The authors presented the concept of applying Domain Specific Modeling (DSM) to the robotics domain to handle	DSM and Feature Model	The use of DSM to support the control of industrial robots using models that are at a higher level of abstraction than traditional

[40]	Specification	<p>the challenges of industrial robotics development.</p> <p>This paper proposes a set of requirements that will aid engineers in creating robots and interfaces that will be effective and practical for use in autism therapy.</p>	Interview, questionnaire, textual requirements	robot programming languages.
[41]	Elicitation	<p>The goal of this paper is to present the first steps of a systematic exoskeleton design lifecycle and evaluate the method from a qualitative point of view</p>	Goal Oriented RE (GORE), KAOS and formal methods	<p>This paper enumerates a set of requirements for a robot and the associated human interface.</p> <p>Therapists in the treatment of children with autism properly used the proposed design.</p> <p>Although requirement modeling requires an initial effort from the designer regarding goals formulation, the proposed approach provides a more comprehensive system overview and documentation.</p>
[44]	Specification	<p>In this project the authors followed the “engaging personas and narrative scenarios” process [2] for developing robot systems</p>	Agile (narrative scenarios and engaging personas)	<p>The benefit from using personas and scenarios was their role as communication catalyst and how the descriptions facilitate building a common vision within the project team.</p>

According to the study [18], all RE phases are addressed by at least one technique. However, we also noticed the most techniques used for requirements definition for the robotic systems come from general systems. That is, they were created to define general system requirements and do not take into account the particularities inherent to robotic system domain. Accordingly, we realize there is a clear need for research in order to (i) identify and analyze some challenges and issues to perform requirement definition for the robotic systems; (ii) identify the main technical features which affect the requirement definition for this domain and (iii) group them as an extensible set of cooperating technical features that help robotic system engineers to perform the requirements definition task.

5 Challenges for Requirement Definition for the robotic Systems

This section describes how difficult is requirements definition for robotic systems (*Section 5.1*) and the main challenges and issues to perform this task (*Section 5.2*).

5.1 Problem Formulation

Simply stated, a robot is a complex hardware-software system made up able to plan and carry out tasks autonomously. During the tasks execution, the robot interacts with the *environment* (inherently affected by errors, uncertainty and noise), with other *robots* (requires ability to exchange data, to coordinate task execution and to synchronize the activities), or even with the human *operator* (requires appropriate GUI and speech and gesture recognition tools) [13]. Robotic systems are made up of components that communicate through a variety of media (e.g. shared memory, radio links or internet). These systems must deal with uncertainty. The constantly changing environment imposes a great challenge that must be well managed. Intelligence levels are required because in many situations robotic systems must act in an autonomous way, that is, based on the perception of the system context, it must take some decision to act automatically.

In order to properly modeling these systems, detailed descriptions of robot functionalities should be described by requirements. These requirements must be detailed as

the robotic scheme should operate and solve the conditions. Common requirements definition problems that every robot system engineer has to solve are described following:

- How can the environment, in which the robotic system is immersed, affect the requirements definition task?
- How to identify and analyze the various interaction possibilities of the robotic system within the external world?
- What are the common ways used to express robotic systems actions (physical or behavioral)?
- How can the choice of technologies to build hardware and software components affect the requirements definition task?
- How robotic systems can decide to do some action? And how to select plausible actions to cope a particular event?

It is worthy to report these questions are not an exhaustive list of requirements definition problems. They were used aiming at illustrating the difficulty that robotic systems engineers must face in order to properly define the requirements. In the next subsection, we will present the main challenges which have intimate connection with the specific characteristics of the robotic system domain.

5.2 Main Challenges

We do not intend to display an exhaustive set of challenges and issues; instead, we intend only to point out those ones identified so far in the considered studies. It is worthy to mention this set has not been adequately validated by researchers and practitioners. We intend to accomplish this research effort in a future work. We gathered the following challenges and issues:

Dealing with uncertainty: Robotic systems should be context aware allowing for monitoring the environment and adapting to the changes in it. The constantly changing environment imposes a great challenge that must be well managed during early stages of requirement definition [21]. Several techniques and approaches have been developed to deal with technical features of intelligence associated with robotic systems. The table 2 describes some related studies related to these techniques, as well as the tools, languages and approaches adopted and the main findings of each study.

Table 2. Studies related to address intelligence levels on robotic systems.

Study	Study description	Tools, languages Approach	Findings
[25]	Attempted to formalize a new class of requirements to address adaptive requirements.	Goal Oriented RE (GORE). Event engineering & Analysis Toolkit (EEAT). Object Constraint Language (OCL)	Proposed a new type of requirements “awareness requirements”. Described elicitation and formalization of such requirements.
[21]	Authors studied the change of requirement during runtime	Requirement Reflection	Defined a new concept aiming at managing the change of requirement during runtime.
[30]	The authors studied the uncertainty in adaptive systems	Continuous adaptive requirements engineering approach (CARE)	Proposed to use analogous mechanisms to achieve requirements reflection
[32]	The authors introduce a goal-based modeling to develop requirements for dynamically adaptive systems Authors presented a Travel Companion system scenario to explain their research as an approach to enable requirements aware systems	RELAX specification language	Uncertainty must be handled when developing for dynamically adaptive systems
[26]	CARE and RE language “Techne”		Proposed a goal-and user-oriented framework for building Self-Adaptive systems

Defining appropriate models: Define models to specify, understand, and implement robotic system is challenging. Engineers must understand the role of high level goals and how they will be decided upon and achieved dynamically; they should also decide on the level of dynamism to incorporate into the software. High dynamism, in which systems evolve and change throughout their lifetime, may lead to unexpected behavior. A number of research papers addressed the problem of autonomy requirements representation. Two main concepts were most intriguing: Awareness Requirements [25] and Requirements Reflection [21]; with the first defining a new type of requirements and the second proposing a way to analyze requirements at runtime. Robotic systems may require the dynamic selection of optimal solutions from several alternatives at runtime. This selection is guided by the occurring changes in the environment, an unattainable capability with static requirements [26]. Goal-oriented modeling languages such as KAOS and i* are thought to be promising in this field since they integrate certain technical features [26] allowing for automated reasoning about goals.

Integration of RE Models: The modeling conventions and methodologies simplifies the RE techniques. To the best of our knowledge, there is no definition of appropriate models for the robotic systems domain [37]. Modeling theory which incorporates RE modeling elements is described in [29] as being a great challenge in the field of robotic systems. Most research projects focus on a RE phase singly (e.g. elicitation or validation) and there has been little work on interconnection of requirement models and combining RE phases [24] mainly in the robotic domain [19]. Well defined approaches are required to interrelate RE goals, scenarios, data, functions, state-based behavior and constraints. This problem affects almost all RE phases and subsequent effect will be on architecture and design of the robotic system.

Technique Selection: Regarding the five RE phases, each one is supported by one or more techniques [19] considering robotic systems domain. In general, the elicitation and specification phases received greater attention from the academic community whereas validation, analysis & negotiation and management phases still require more attention [18] [22]. There is a great diversity of requirement definition techniques for each RE phase, being models, textual requirements, scenario-based and goal-oriented more frequent, in this order. Finally, we can highlight there are a few empirical pieces of evidence to state benefit on some techniques to perform RE phases in the modeling of robotic systems.

Requirements Reuse: The reuse of existing requirement artifacts makes the RE task more prescriptive and systematic [28]. The reusability of requirements facilitates the advantages at design level as well as in the development of related domain system or applications. Several robotics projects, such as those documented in ORCA and MARIE projects [7], aim at creating software environments for the development of reusable objects which are interoperable across robotic platform and control applications. What is still missing to achieve stability of middleware service for the robotics domain (those that sit between physical devices and control applications to enable interoperation) is a clear understanding of the functional and communication requirements of robotic Industrial Objects. The challenge is to let emerge this knowledge from common practice [13].

Synchronization between requirements and architecture: This challenge was pointed by the authors of [21]. Changing requirements at runtime might affect the architecture of the software. That effect must be managed carefully to ensure that the architectural components are changed smoothly so that no requirements or goals are broken during the process. Consequently, Autonomic specific architectures have been formed. IBM introduced what is now considered a prevalent architecture for autonomic systems [23], the MAPE

architecture. This architecture is composed of four key components: Monitor, Analyze, Plan, and Execute. Although it is widespread, some researchers think that this architecture does not reflect autonomic systems completely [23]. Other architectures are proposed, however with no solid achievement, such as the Intelligent machine Design (IMD) Architecture [23].

Addressing NFR: The Non Functional Requirement (NFR) includes the system attributes like security, scalability, operability and much more [11]. The general system development method generally gives less attention to NFRs compared to robotic system development. The problem is that a robotic system has to satisfy a considerable number of functional and non-functional requirements even for implementing basic robot skills [2][9]. This presents an even greater challenge compared to general systems. For example, the ability of navigating in an indoor environment autonomously requires the mobile robot to express several basic behaviors, such as obstacle avoidance, self-localization, path planning, and place recognition. The robot has to execute most of these behaviors simultaneously, thus the control application must enforce NFR such as real-time performance, fault tolerance, concurrence, and distribution.

Conflicting and Ambiguous Requirements: The different stakeholders' opinions, objectives, needs may have different meanings and may conflicts with vague words. When eliciting the requirements; the terminology, keywords and domain knowledge should be properly notified [24]. Simply stated, the business of robotics research consists in "requirements that allow robots to function autonomously in unstructured, dynamic, partially observable, and uncertain environments" [27]. This is an ambitious goal that describes the "user requirements" of every robotic system. These requirements are commonly expressed in fuzzy and ambiguous terms that make it hard to understand what functionality the robotic system being built must express [13].

6 Framework for Identifying Robotic Requirements

Aiming to cope with the challenges and issues previously described, we propose a requirement abstraction outlined in Fig. 3. During the task of requirement definition, several *internal* and *external* technical features specific for robotic systems must be considered. These technical features will be described in detail in the following:

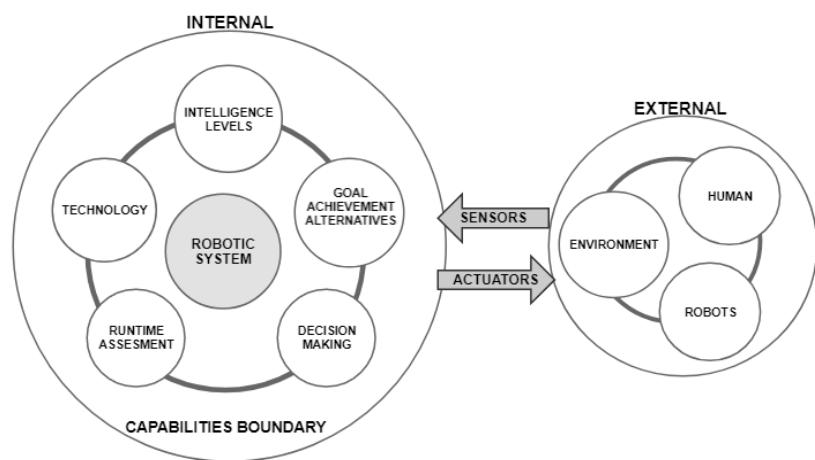


Fig. 2. Requirement abstraction for robotic system.

Level of Intelligence: The concept of intelligence (i.e. the ability of expressing useful behavior) is quite elusive [22]. It is usually associated to other concepts, such as: autonomy - the robot's ability to control its own activities and to carry on tasks without the intervention of the human operator; deliberativeness - the ability of planning and revising future actions in order to achieve a given goal while taking into account the mutable conditions of the external environment; or adaptability - the ability of changing its behavior in response to external stimuli according to past interactions with the real world [23]. Deciding the level of intelligence shall enable the engineers to make correct choices on how the system in hand will be developed.

Choice of Technology: There is a wide variety of hardware and software to compose robotic systems. A development technology might be dictated by the system requestors, or the domain in which the software will operate. The choice between these components represents important design decisions. In addition, development of robotic system requires a different environment that enables runtime adaptation. The XML-Based Autonomic Computing Expression Language (ACEL) [22] and Software Component Ensemble Language (SCEL) [23] operating on a Java Runtime Environment are two examples of technology supporting autonomic computing.

Runtime Requirement Assessment: Robotic systems usually require a mechanism to monitor and evaluate the success or failure of achieving the system goals at runtime. This mechanism could be based on the chosen technology or developed separately. Prior research has suggested a number of ways to this such as the definition of Awareness Requirements [25], and the concept of requirements reflection [21].

Decision-Making: Understanding the runtime behavior of the requirements helps in deciding about the action to be taken by the robotic system. Robot environment behavior is governed by three major components: (i) the robot itself, its sensors and actuators, (ii) the environment's perceptual properties and (iii) the task, usually the control program being executed by the robot. The robot's behavior emerges through the interaction of these three aforementioned components.

Goal Achievement Alternatives: Clarifying alternative ways to achieve goals is an essential part for developing requirements for the robotic systems. The uncertainty associated with the environment in which robotic systems exist, may hinder certain actions. However, that should not prevent the system from achieving the main goal.

System Interactions: By capabilities we mean what the system can do to interact with the surrounding world by robotic system in terms of physical connections and data interaction. The physical interactions occur through *sensors* and *actuators*. Sensors allow the robot to perceive the environment's changes and to react to them by changing its own behavior. Actuators are the tools with which the robot changes the environment.

External technical features also affect the requirements definition task. First, the collaboration of the robotic system with the *environment* will assist in predicting the possible changes and required responses to be attained by the system. It refers to existing in a complex, dynamic, and unstructured environment that strongly affects the robot behavior. Second, the collaboration with other *robots* requires the ability to exchange data (e.g. sensory information acquired from different viewpoints), to coordinate task execution (e.g. assigning roles to robot team-mates), and to synchronize activities (e.g. two robots assembling a work piece). Finally, the collaboration with the *human operator* is often desirable for robotics tasks. Effective Human-Robot interaction [13] requires the exploitation of advanced technologies, such as appropriate graphical user interfaces, speech and gesture recognition tools, and haptic mechanical interfaces.

The table 3 briefly groups all these technical features, presenting their descriptions, objectives and the main existing available approaches. The objective of this study was only to identify some technical features which directly affect the requirement definition task. Further research is required in order for fully understand the relationship between these features.

Table 3. Concerns to be considered for requirement definition for the robotic systems.

Concerns	Description	Objective	Available approaches
External Environment	Define the environment in which system will exist as well as the collaboration with human and other robots.	Capture environment and information related to collaboration	- Natural language representation of environment elements
System Interactions	Define the systems connections and interaction with the environment.	Identify all connections, sensors and actuators	- Analyze the base requirements of the system - Five levels of Autonomy [22]: Basic, Managed, Predictive, Adaptive, Autonomic
Intelligence Level	Decide on the desired level of autonomy, to determine functionality.	Minimize cost, Increase efficiency and dependability	- Eight levels of Autonomy Assessment Scale [33]
Technology Choice	Describes the development environment and programming language.	Familiarity with technology or available training	- Software Component Ensemble Language (SCEL) - Agent-Oriented Programming Languages [34]
Runtime Requirement Assessment	Monitor and evaluate the success or failure of achieving the system goals at runtime.	Provide an accurate measure of requirements success	- Awareness Requirements [25] - The concept of requirements reflection [21].
Decision Making	Describe a mechanism for task planning and decision making to achieve goals	Optimize decision making process	- Rule-Based approach - Control Theory Approach - Biology inspired processes
Alternatives to Goal Achievement	Describe possible alternatives to achieve goals	Provide more than one path to goals	- Further Analysis of the goals

7 Threats to Validity

We followed the guidelines for conducting exploratory studies described in [36][35]. Like any other empirical study, the present study can also have limitations that must be considered for analyzing the potential impact of the validity threats to its findings. We discuss three types of validity threats related to:

Identification of studies: In the literature search strategy, we aimed to retrieve as many relevant studies as possible to avoid any possible literature selection bias. We faced a challenge in determining the scope of our study as the notion of “requirements definition” means different things to different research communities including software engineering, robotic systems and others. Therefore, to cover them all and avoid any bias, we searched the literature based on relevant terms and combined them in our literature search.

Quality of studies and data extraction consistency: The results and quality of this study are based on the quality of the studies that have been retrieved. This means that if the quality of the primary studies is low, the claims and their supporting evidence are unlikely to be strong and reliable. Therefore, it is vital to (i) minimize the threats regarding the quality of selected studies and to ensure (ii) a consistent representation of data extracted from these studies.

Data synthesis and results reporting: The final type of threat corresponds to the bias or a lack of systematic approach to synthesize and report the results. We tried to mitigate this threat by conducting a pilot study. A limited number of researchers and their expertise (software and requirements engineering) may have an internal bias on the style and reporting of results. Although we followed the guidelines from [36] [35] to conduct the study, we had deviations from the ideal approaches based on the requirements of this research. We believe that the validity of the study is high, given the use of a systematic and recommended procedure and a pilot study to refine the scope of review.

8 Concluding and Future Works

The Robotic systems are increasingly being integrated into various aspects of everyday life. They are made up of hardware and software components. The requirement definition for this type of system is more complex than general systems. It required the achievement of high level goals taking into consideration the changing environment. Defining the requirements for the robotic systems requires identifying certain technical features surrounding and contributing to the system. It is not sufficient to consider only the basic software requirements, but also hardware requirements.

Our strategy to address our research question was an exploratory study to identify and analyze the existing techniques and research progress that influence requirements definition for the robotic systems. Therefore, the key contributions of this study are: (i) identifying and exploring the challenges and issues associated with requirements definition for robotic systems and (ii) providing robotic system engineers with a framework to identify and manage several technical features which affect requirements definition. We expected the proposed framework shall support the robotic system engineer to further understand the robotic functionalities and make ease the requirements definition task.

This work is part of a broader project which aims at analyzing important issues and challenges involved in requirements definition task for the robotic system domain. The presented set of challenges and issues are based on prior research and more research efforts can be required to properly validate those ones described in this research. We intend to provide a modeling language to support the requirement abstraction described in the proposed framework.

Acknowledgments.

The following Brazilian institutions have supported this work: UFPE, CNPq, FACEPE.

References

1. Parker, L.E.: ALLIANCE: An architecture for fault tolerant multirobot cooperation. In IEEE transactions on robotics and automation 14, no. 2 (1998), 220-240.
2. Maenpaa, T., Tikanmaki, A., Riekki, J. and Roning, J.: A distributed architecture for executing complex tasks with multiple robots. In Robotics and Automation, 2004. Proceedings. ICRA'04. IEEE International Conference on Robotics and Automation, vol. 4, (2004), 3449-3455.

3. Angerer, A., Hoffmann, A., Ortmeier, F., Vistein, M. and Reif, W.: Object-Centric Programming: A New Modeling Paradigm for Robotic Applications. In International Conference on Automation and Logistics, (2009), 18-23.
4. Buzurovic, I., Podder, T.K., Fu, L. and Yu, Y.: Modular software design for brachytherapy image-guided robotic systems. In International Conference on Bioinformatics and Bioengineering, (2010), 203-208.
5. Hu, G., Tay, W.P. and Wen, Y.: Cloud robotics: architecture, challenges and applications. In IEEE Network 26.3, (2012).
6. Jackson, J. and Coll, C.H.M.: Microsoft robotics studio: a technical introduction. In Robotic Automation Magazine 14.3, (2008).
7. Bonarini, A., Matteucci, M., Migliavacca, M., and Rizzi, D.: R2P: An open source hardware and software modular approach to robot prototyping. Robotics and Autonomous Systems, 62(7), (2014), 1073-1084.
8. Elkady, A. and Sobh, T.: Robotics middleware: A comprehensive literature survey and attribute-based bibliography. Journal of Robotics, (2012).
9. Petersen, K., Feldt, R., Mujtaba, S. and Mattsson, M.: Systematic mapping studies in software engineering. In Proceedings of International Conference on Evaluation and Assessment in Software Engineering, vol 8, (2008), 68-77.
10. Schofield, M.: Neither master nor slave... A practical case study in the development and employment of cleaning robots. In International Conference on Emerging Technologies and Factory Automation, vol 2, (1999), 1427-1434.
11. Kotonya, G. and Sommerville, I.: Requirements engineering processes and techniques. Wiley, New York, (1998).
12. Brugali, D.: From the editor-in-chief: a new research community, a new journal. In Journal of Software Engineering for Robotics, Nr 1, (2010), 1-2.
13. Brugali, D. and Reggiani, M.: Software Stability in the Robotics domain: issues and challenges. In Information Reuse and Integration, (2005), 585-591.
14. Oliveira, L.B.R., Osório, F.S. and Nakagawa, E.Y.: An investigation into the development of service-oriented robotic systems. In Proceedings of the 28th Annual ACM Symposium on Applied Computing, (2013), 223-228.
15. Pons, C., Giandini, R. and Arévalo, G.: A systematic review of applying modern software engineering techniques to developing robotic systems. Ingeniería e Investigación, Nr 32.1, (2012), 58-63.
16. Heineck, T., Gonçalves, E., Sousa, A., Oliveira, M. and Castro, J.: Model-Driven Development in Robotics Domain: A systematic literature review. In Software Components, Architectures and Reuse - SBCARS, (2016), 151-160.
17. Pereira, T., Albuquerque, D., Sousa, A., Alencar, F. and Castro, J.: Retrospective and Trends in Requirements Engineering for Embedded Systems: A Systematic Literature Review. In Workshop on Requirements Engineering (WER), part of IberoAmerican Conference on Software Engineering - CIBSE, (2017), 427-440.
18. Sousa, A., Agra, C., Melo, J. and Alencar, F.M.: Elicitação e Especificação de Requisitos em Sistemas Embarcados: Uma Revisão Sistemática. In Workshop on Requirements Engineering (WER), part of Conferencia Iberoamericana de Software Engineering- CIBSE, (2015).
19. Albuquerque, D. Heineck, T., Castro, J.: Requirements Engineering for Robotic System: A Systematic Mapping Study. In Workshop on Requirements Engineering (WER), part of Conferencia Iberoamericana de Software Engineering - CIBSE, (2017), 333-346.
20. Marwedel, P.: Embedded Systems Design. Kluwer Academic Publishers, (2003).
21. Bencomo, N., Whittle, J., Sawyer, P., Finkelstein, A. and Letier, E.: Requirements reflection: requirements as runtime entities. In International Conference on Software Engineering, ACM/IEEE 32nd, (2010), 199-202.
22. Chauhan, S.K.: Autonomic computing: a long term vision in computing. International Journal of Global Research in Computer Science (UGC Approved Journal), 3(5), (2012), 65-67.
23. Shuaib, H., Anthony, R.J. and Pelc, M.: A Framework for Certifying Autonomic Computing Systems. In proceedings of International Conference on Autonomic and Autonomous Systems, (2012), 122-127.

24. Shah, T. and Patel, S.V: A review of requirement engineering issues and challenges in various software development methods. *International Journal of Computer Applications*, (2014), 36-45.
25. Silva Souza, V.E., Lapouchnian, A., Robinson, W.N. and Mylopoulos, J.: Awareness Requirements for Adaptive Systems. In proceedings of International Symposium on Software Engineering for Adaptive and Self-Managing Systems, (2011), 60-69.
26. Qureshi, N.A., Perini, A., Ernst, N.A. and Mylopoulos, J.: Towards a Continuous Requirements Engineering Framework for Self-Adaptive Systems. In International Workshop on requirements, (2010), 9-16.
27. Johnson, V.: The San Francisco project: business process components and infrastructure. In ACM Computing Surveys, 32 (1es), (2010).
28. Cheng, B.H. and Atlee, J.M.: Research Directions in Requirement Engineering. In Future of Software Engineering. IEEE Computer Society, (2007), 285-303.
29. Broy, M.: The grand challenge in informatics: Engineering software-intensive systems. *Computer*, 39(10), (2006), 72-80.
30. Sawyer, P., Bencomo, N., Whittle, J., Letier, E. and Finkelstein, A.: Requirements-aware systems: A research agenda for re for self-adaptive systems. In Requirements Engineering Conference, (2010), 95-103.
31. Souza, V.E.S.: A Requirements-Based Approach for the Design of Adaptive Systems. In proceedings of International Conference on Software Engineering, (2012), 1635-1637.
32. Cheng, B.H., Sawyer, P., Bencomo, N. and Whittle, J.: A goal-based modeling approach to develop requirements of an adaptive system with environmental uncertainty. In International Conference on Model Driven Engineering Languages and Systems, (2009), 468-483.
33. Proud, R.W., Hart, J.J. and Mrozinski, R.B.: Methods for Determining the Level of Autonomy to Design into a Human Spaceflight Vehicle: A Function Specific Approach. NASA Johnson Space Center, Houston, Texas, (2003).
34. Bădică, C., Budimac, Z., Burkhard, H.D. and Ivanovic, M.: Software agents: Languages, tools, platforms. *Computer Science and Information Systems*, 8(2), (2011), 255-298.
35. Wohlin, C.: Guidelines for snowballing in systematic literature studies and a replication in software engineering. In Proceedings of the 18th international conference on evaluation and assessment in software engineering, (2014), 38.
36. Brereton, P., Kitchenham, B.A., Budgen, D., Turner, M. and Khalil, M.: Lessons from applying the systematic literature review process within the software engineering domain. *Journal of systems and software*, 80(4), (2007), 571-583.
37. Vashev, E. and Hinckey, M.: Autonomy Requirements Engineering. In Autonomy Requirements Engineering for Space Missions. Springer, Cham, (2014), 105-172.
38. Patoglu, V., Ertek, G., Oz, O., Zoroglu, D. and Kremer, G.: Design Requirements for a tendon rehabilitation robot: results from a survey of engineers and health professionals. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, (2010), 85-94.
39. Sun, Y., Gray, J., Bulheller, K. and von Baillou, N.: A model-driven approach to support engineering changes in industrial robotics software. In International Conference on Model Driven Engineering Languages and Systems, (2012), 368-382. Springer, Berlin, Heidelberg.
40. Giullian, N., Ricks, D., Atherton, A., Colton, M., Goodrich, M. and Brinton, B.: Detailed requirements for robots in autism therapy. In Systems Man and Cybernetics (SMC), (2010), 2595-2602.
41. Souza, R.S., Sanfilippo, F., Silva, J.R. and Cordero, A.F.: Modular exoskeleton design: Requirement engineering with KAOS. In International Conference on Biomedical Robotics and Biomechatronics (BioRob), (2010), 978-983.
42. Björndal, P., Rissanen, M.J. and Murphy, S.: Lessons learned from using personas and scenarios for requirements specification of next-generation industrial robots. In International Conference of Design, User Experience, and Usability, (2011), 378-387.