Requirements Engineering (2021) 26:115–141
https://doi.org/10.1007/s00766-020-00337-x

ORIGINAL ARTICLE

Safety requirements for symbiotic human–robot collaboration systems in smart factories: a pairwise comparison approach to explore requirements dependencies

Georgia Dede[1] · **Persefoni Mitropoulou**[1] · **Mara Nikolaidou**[1] · **Thomas Kamalakis**[1] · **Christos Michalakelis**[1]

Received: 2 October 2019 / Accepted: 8 July 2020 / Published online: 2 August 2020

Abstract

Industry 4.0 is expected to deliver significant productivity gain taking advantage of Internet of things (IoT). Smart solutions, enhanced by IoT, are constantly driving revolutionary approaches in multiple domains. Smart factories are one domain where intelligent integrated robotic systems will revolutionize manufacturing, resulting in a complex ecosystem, where humans, robots and machinery are combined. In this setting, human safety requirements are of paramount importance. This paper focuses on symbiotic human–robot collaboration systems (HRC), where human safety requirements are essential. Hence, it aims to explore and prioritize human safety requirement dependencies, as well as their dependencies with other critical requirements of smart factory operation, as effectiveness and performance. Toward this end, the proposed approach is based on SysML to represent the requirements dependencies and pairwise comparisons, a fundamental decision-making method, to quantify the importance of these dependencies. This model-driven approach is used as the primary medium for conveying traceability among human safety requirements as well as traceability from safety requirements to effectiveness and performance requirements in the system model. The analysis is based on the operational requirements identified in the European project HORSE, which aims to develop a methodological/technical framework for easy adaptation of robotic solutions from small-/medium-sized enterprises. Validation of the results is also performed to further elaborate on human safety requirement dependency exploration. The outcomes of this paper may be beneficial for symbiotic HRC systems in the early design stage. As the system is being developed with an emphasis on human safety, all these requirements that have been assessed with highly prioritized dependencies should be taken into account, whereas those with negligible ones have to be ignored since they do not significantly affect the rest of the process. Since operational requirements may be conflicted and incompatible, this approach may be very useful for other systems as well during the system design phase to find the appropriate solution satisfying the majority of the requirements, giving a priority to the ones with highly ranked dependencies and hence facilitating the implementation phase and afterward the production line. The outcomes may be used as a step in developing a model-driven approach which should be able to support the manufacturing process, facilitating the integration of systems and software modeling, which is increasingly important for robotic systems in smart factories incorporating HRC.

Keywords Symbiotic human–robot collaboration systems · Safety · Requirement analysis · Dependencies · SysML · Decision making · Pairwise comparison

1 Introduction

Centered around advanced robotics and automation, new ways of human–machine interaction (HMI) and vast troves of data and boosted connectivity, Industry 4.0 is poised to modernize manufacturing and boost industrial competitiveness. Coupled with the emerging Internet of things (IoT), Industry 4.0 offers manufacturers the ability to collect, analyze and act on immense stockpiles of data like never before and then set those actions in motion with highly efficient, automated robotics. The use of advanced robots in manufacturing is becoming more and more commonplace in industry. Where robots used to be applied mainly in large, high-tech manufacturing plants, their application becomes increasingly accessible for a diverse range of manufacturing companies,

✉ Georgia Dede
gdede@hua.gr

Extended author information available on the last page of the article

even small-/medium-sized enterprises (SMEs) which do not necessarily active in the high-tech market. The use of robots in the manufacturing process is, however, not always flexible and efficient. This is caused primarily by the human safety requirements: Their use may present a safety hazard for human workers in the same physical space. Consequently, spaces where robots and humans work are often physically separated leading to inflexibility and inefficiency [1].

Robotic technologies have generally been developed for capital-intensive, large-volume industrial manufacturing. This explains why SMEs and mid-capitals are currently lagging behind in their adoption. On the other hand, SMEs do not only represent the big potential robotic market covering a wide range of industries, but also face the same challenges in the global market which require rapid reconfiguration for their production systems, enhanced safety, smaller lots production and reduced costs. HORSE, a European Research and Innovation Project in the EU Horizon 2020 Framework [2], will support SMEs to overcome the difficulties they face in adopting robotics, e.g., low awareness of the technological improvements, low technical competence beyond their core business and hesitation to new long-term investment. HORSE proposes a comprehensive set of activities to speed up adoption of emerging advanced manufacturing technologies of highly flexible and near-autonomous robotics systems. These activities serve the entire value chain and propose new concepts and business models for robotics systems servitization and for product operation. The safety requirements of the HORSE project, on which this paper is based, primarily focus on human safety. Safety of the human worker as well as reduction in health risks through physical support by the robotized equipment will contribute to better overall manufacturing processes.

As more and more industrial robots are used in manufacturing, the need for robot-related safety standards during not only the design but also the implementation stage increases. Human safety requirements are considered to be of great importance for allowing human workers access to the robot work space during operation, minimizing the likelihood of accidents. Worker injuries or even deaths related to robot accidents have been reported worldwide confirming the fact that really dangerous situations may arise during human–robot interaction (HRI) in the same workplace [3, 4]. Robots are extremely useful and necessary industrial machines; however, they can manipulate dangerous tools and move rapidly with force and this can cause accidents. Identifying the sources of potential harm, the workers in the robot's vicinity who may be in greatest peril, the type of control abilities robots should acquire to be capable of conforming to new forms of operation and the factors which have the greatest impact on safety are some of the safety issues that need to be addressed in industrial settings [5].

Human safety is an important consideration in HRI. Industrial requirements for automation of manufacturing operations are driving technology into the direction of dealing with the hazards identified by safety analysis [6]. Safety analysis, therefore, provides the mechanism for the identification of the human safety-related requirements which are essential for improving the safety of the system and bringing HRI into common experience. Collaborative robots are usually installed to shared environments and are tested under different scenarios including not only human safety, but also effectiveness and performance aspects deriving human safety requirements for robotic systems able to resolve the relevant low-level injury risks [6]. In light of this, the aim of the present paper is to develop a model-driven approach in order to model, explore and prioritize the dependencies of human safety requirements for smart integrated robotic systems as an attempt to benefit robotic system decomposition in the early system development activities. This paper highlights the significance of a human safety requirement analysis, focused on the exploration of requirement dependencies, whose results are used in the robotics system design process for the development of the related functionality, of the required safety capabilities and in the standardized documentation of the requirements to be fulfilled by industrial robots. This work focuses on symbiotic HRC systems, an integral part of cyber-physical human systems in the Industry 4.0 era. In such symbiotic systems human safety requirements, both functional and non-functional are essential. The approach used is based on SysML for the representation of the safety requirements interdependencies as well as their dependencies with other categories, these of effectiveness and performance and a decision-making procedure, namely pairwise comparisons (PWC), in order to capture and prioritize these dependencies/interdependencies.

Since this paper focuses on symbiotic HRC systems in the Industry 4.0 era, exploring the dependencies of safety requirements with each other as well as with effectiveness and performance requirements have an important influence on software engineering activities, like project planning, architecture design and implementation phase. As the system is being developed with an emphasis on human safety, all these requirements, whose dependencies have been assessed as significant and highly prioritized, must be taken into account, whereas those with negligible dependencies have to be ignored since they do not significantly affect the rest of the process. Since the requirements are sometimes conflict and incompatible, this approach may be very useful for other systems, during the system design process, to find the appropriate solution satisfying the majority of the requirements, giving a priority to the ones with highly ranked dependencies and hence facilitating the implementation phase and afterward the production line. The outcomes may be used as a step in developing a model-driven approach which should be able to support the manufacturing process, facilitating the integration of systems and software modeling, which is increasingly important for robotic systems in smart factories incorporating HRC systems.

The results were further elaborated using a validation approach based on sensitivity analysis and Monte Carlo (MC) simulations to investigate the reliability of the final outcomes. The stability of the results was examined by incorporating uncertainty that may undermine the opinion of participants involved in the decision-making process. Interestingly enough, the outcomes seem to hold even under uncertainty which therefore enhances the accuracy of the approach.

The rest of the paper is organized as follows. A short overview of related work is presented in Sect. 2. In Sect. 3, the background of the HORSE system is analyzed. The proposed approach to explore human safety requirements dependencies is described in Sect. 4, whereas Sect. 5 deals with the discussion and analysis of the obtained results. Finally, some concluding remarks and future directions are given in Sect. 6.

2 Related work

2.1 System safety requirements and dependencies

Safety critical systems within different technological sectors are usually developed subject to the recommendations outlined in the corresponding official standards. These standards give guidance on the "determination" of requirements. Safety requirements derived through safety analysis often place integrity constraints on existing functions of a system resulting in new functional requirements which may be needed to prevent or mitigate the effects of failures identified in the analysis [7]. According to [8], it is also a good practice to treat safety-related functional requirements in a manner consistent with other requirements applicable at the development phase, since they are subject to the same obligations as other requirements with respect to traceability.

Existing literature focuses especially on modeling and deriving safety requirements for software systems or components using different methodologies depending on each case study. In [9], the authors examine how the results of one safety analysis technique, fault trees, are interpreted as software safety requirements to be used in the program design process. The proposed model is formalized in a real-time, interval logic, based on a conventional dynamic systems model with state evolving over time. Another approach used in [10] for a train-set crossing incorporates fuzzy set modeling and evidential reasoning to assess the safety associated with safety requirements specifications. The developed methodology using specific parameters, such as failure likelihood, consequence severity and failure consequence probability, is capable of dealing with multiple safety analysts who make judgments on each safety rule. Furthermore, the application of model-based design, by means of SysML, is explored for e-Health systems in [11] emphasizing criticality requirements, which are modeled as SysML requirements, while SysML constraints and parametric diagrams are employed to describe and verify quantitative criticality requirements. The approach illustrates the diverse criticalities of the case study in the form of—manageable—SysML requirements, and mathematical relationships and validation expressions among the components and operational requirements of the examined system.

More specifically, exploring related research about dependencies and prioritization of system safety requirements, Firesmith [12] illustrates safety's position within a quality model showing how safety requirements are related to other quality requirements by decomposing safety into its quality sub-factors. The resulting aggregation hierarchy of safety sub-factors is used to identify a corresponding hierarchy of safety requirements built upon safety metrics and system-specific safety criteria for these safety sub-factors that may be useful for identifying potentially missing types of safety requirements. In addition, according to [13] the difficult task of prioritizing requirements is addressed so that the highest priority requirements can be implemented first as part of the scheduling of an incremental, iterative and time-boxed development cycle. Requirements at a lower tier in the overall system structure "implement" requirements on a higher tier. Thus, software requirements implement subsystem requirements which implement system requirements. Dependency relationships between use cases and usage scenarios imply dependencies between their priorities. Derived requirements are usually engineered to support more fundamental requirements, which depend on the implementation of the derived requirements. Related literature puts emphasis especially on requirements dependencies, since understanding these dependencies is proven to improve the requirements process. The different occurrences of requirements changes throughout a project's life cycle point out dependencies among functional requirements. The proposed approach provides a modular way to organize requirements and a proper granularity to analyze requirements dependencies [14]. In [15], a requirements dependencies matrix is used as a practical tool to assess to which extent software functional requirements depend on each other and finally support software product-line engineering and identify an effective set of system functions such that to reduce disturbing dependencies. In addition, traceability research is gaining increasing attention in many areas such as requirements engineering and model-driven architecture. It includes not only the forward and backward links between artifacts, but also links between items within a software development

Table 1 Summary of the related work on the analysis of safety requirements

Topic	References	Type of approach	Description	Results/contribution
System safety requirements and their dependencies	[9]	Fault tree analysis	The model is formalized in a real-time, interval logic Based on a conventional dynamic systems model with state evolving over time	Fault trees are interpreted as temporal formulas It is shown how such formulas can be used for deriving safety requirements for software systems and be used in the design process
	[10]	Subjective analysis with fuzzy set modeling and evidential reasoning	Three parameters: failure likelihood, consequence severity and failure consequence probability are used to analyze a safety rule in terms of membership functions. The subjective safety description associated with the safety rule is then mapped back to the defined safety expressions The information produced for all safety rules is synthesized using evidential reasoning A case study based on a train-set crossing is used	The mapping results in the production of the safety evaluation associated with the safety rule. Evidential reasoning is used to obtain the safety evaluation associated with the safety requirements specifications The developed methodology is capable of dealing with multiple safety analysts who make judgments on each safety rule
	[11]	Model-based approach using SysML	Identified criticalities are modeled as SysML requirements, while SysML constraints and parametric diagrams are employed to describe and verify quantitative criticality requirements The remote elderly monitoring system (REMS) use case of IoT e-Health systems is used	The approach illustrates the diverse criticalities of the REMS Home Subsystem in the form of—manageable—SysML requirements, and mathematical relationships and validation expressions among the components and operational requirements of the examined system It is the first concrete step toward formal SysML models to shed light to the design and management of complex mixed-criticality healthcare systems
	[12]	Quality model	The model illustrates safety as a quality factor by decomposing safety into its quality sub-factors.	It shows how safety requirements are related to other quality requirements The resulting aggregation hierarchy of safety sub-factors identifies a corresponding hierarchy of safety requirements built upon safety metrics and system-specific safety criteria for these safety sub-factors that may be useful for identifying potentially missing types of safety requirements
	[13]	Prioritization technique	The highest priority requirements are implemented first as part of the scheduling of an incremental, iterative and time-boxed development cycle Requirements at a lower tier in the overall system structure "implement" requirements on a higher tier Dependency relationships between use cases and usage scenarios imply dependencies between their priorities	It shows that prioritizing requirements is both a critical and difficult task for the requirements engineering team A set of recommendations, including a recommended prioritization process, are proposed in order to incorporate requirements prioritization into the requirements engineering process

Table 1 (continued)

Topic	References	Type of approach	Description	Results/contribution
	[14]	Feature-oriented approach	A feature is a set of tight-related requirements from user/customer-views Static feature dependencies (i.e., refinements and constraints), feature dependencies at the specification level (namely influences) and dynamic feature dependencies (namely interactions) together with their underlying connections are explored	The approach provides a modular way to organize requirements and a proper granularity to analyze requirements dependencies It shows that understanding requirements dependencies is proven to improve the requirements process
	[15]	Requirements dependencies matrix	The methodology takes into account product features that may enhance the ability in monitoring and controlling requirements evolution. Empirical investigations of two industrial case studies are used	The results show to which extent software functional requirements depend on each other and finally support software product-line engineering and identify an effective set of system functions such that to reduce disturbing dependencies
	[16]	Traceability tools	It provides an overview of traceability research and practice in software requirements engineering and why it has gained increasing attention and importance in this area The goal is to follow the whole life of requirements from its sources to the products generated from them; basically, the whole software development process is addressed	It shows that the more holistic view of traceability in the requirements engineering domain is a good foundation for further advances regarding challenges in this area
	[17]	Dependency model-based approach	The approach evaluated the usefulness and applicability of two well-known generic dependency models covering 25 dependency types The case study was conducted in a real-world industry project with three participants who offered different perspectives	The evaluation found that there is a number of overlapping and/or ambiguous dependency types among the current models Understanding the effect of these requirement dependencies to software engineering activities is useful A new dependency model is proposed to classify requirements relationships into dependency types based on the structural and semantic properties of requirements
Industrial human–robot system safety requirements	[18]	Integrated HRI strategy	The robot ensures human safety by planning and modifying its trajectory at three different time horizons: long-term path planning, medium-term trajectory planning and short-term reactive control At each stage, a quantitative level of danger is used to guide the decision-making process	A novel methodology for ensuring safety during human–robot interaction through planning and control is presented, based on an explicit quantification of the level of danger in the interaction
	[4]	Real-time methodology for HRI safety	The level of danger in the HRI due to a potential collision is explicitly defined as the danger index A sequential one-step ahead trajectory planner is presented which generates robot motion by minimizing the danger index	A new methodology is proposed for eliminating the risk during HRI deriving safety requirements in real-time industrial environments The basic features of approaches for eliminating the risk during HRI are presented

Table 1 (continued)

Topic	References	Type of approach	Description	Results/contribution
	[6]	Survey of industrial robots and safety requirements	A survey of the developments of industrial robots and of the robot safety standards is presented	This work outlines the steps that have brought research in this area to the present status and also the opening of possibilities for HRI in industrial production The goal of each safety analysis procedure is considered to be a safe and human friendly interaction in an industrial environment, resulting in the imposition of safety measures, such as the utilization of system control and planning
	[5]	Survey of safety issues in HRI	A review of safety issues in industrial settings, where autonomous mobile robots operate in crowded (human-inhabited) environments and of safety issues related to assistive robots, is presented	Safety standards in HRI are stated Technical and effective features such as the size, configuration and environment of the robot to be guarded are also considered very important for a safety requirements analysis
	[19]	Review of Research Issues of the Industry 4.0 and Smart Manufacturing	The related research mostly focuses on new ways to ensure the safety of human workers and limit the restrictions of a divided workspace	It shows that human safety in manufacturing intelligence is a crucial topic for researchers The production process in a smart factory is proposed to run without threats and interruptions, achieving the level of security and safety that meets worker safety legal requirements on the shop floor
	[20]	Review of advancement of cyber-physical systems (CPS) in manufacturing	A review of the current status and the latest advancement of CPS in manufacturing is presented Research and applications are outlined to highlight the latest advancement in the field of Symbiotic human–robot collaboration	Symbiotic human–robot collaboration is defined for a fenceless environment in which productivity and resource effectiveness can be improved by combining the flexibility of humans and the accuracy of machines
	[21]	Safety evaluation method of design and control for human-care robots	A danger evaluation method is developed as an approach to analyze the factors which affect the potential impact force between the robot and the human, taking into account usability and performance features	The derived danger index is used for improved mechanical design and control, which is often considered as the most effective safety strategy
	[22]	Review of research in CPS	An integrated research agenda of CPS is presented Safety requirements are stated	A requirement engineering analysis for industrial robots is necessary from the beginning, meaning the adaption of robotics system to needs and competences of the users, the specification of formal requirements models, the detailing of requirements and finally mapping them to system elements

Table 1 (continued)

Topic	References	Type of approach	Description	Results/contribution
	[23]	Natural language processing	A method to translate safety non-formal requirements to formal descriptions is presented	This approach enables automated information processing and helps writing specifications for industrial robots by transforming requirements in natural language into discipline-specific models
Industrial human–robot system safety requirements and dependencies	Current	Pairwise comparison approach	An approach based on pairwise comparisons is applied to quantify the importance of requirements dependencies SysML is employed to model smart factory's related requirements from the human safety perspective The European project HORSE dealing with robotic solutions in manufacturing is used as a case study	The proposed approach aims to identify human safety requirements of symbiotic human–robot collaboration systems and most importantly to explore their dependencies with each other as well as with other critical requirements, such as these of effectiveness, usability and performance This approach may be very useful for other systems during the system design process to find the appropriate solution satisfying the majority of the requirements, giving a priority to the ones with highly ranked dependencies

artifact. The more holistic view of traceability in the requirements engineering domain is considered to be a good foundation for further advances regarding challenges in this area [16]. The dependencies between individual requirements have an important influence on software engineering activities, like project planning, architecture design and implementation phase. This is mainly why multiple requirement dependency types have been suggested in the literature from different points of interest together with various dependency models which aim to assist in identifying and classifying requirements relationships into dependency types based on the structural and semantic properties of requirements [17]. Some techniques for prioritizing requirements are identified in the existing research studies, including business care analysis, pairwise comparisons and weightings, and finally, a set of recommended prioritization steps of a general process are proposed, some of which are followed in the context of the proposed methodology of this paper.

2.2 Industrial human–robot system safety requirements

Within the industry 4.0 initiative, industrial safety standards (RIA/ANSI, 1999) have been used to ensure safety by employing robots in isolated work cells, away from humans, and thus are not directly applicable to HRI situations. Recently, research has focused on the potential for using industrial robots to more unstructured and interactive environments, where they must be able to aid humans in a safe and friendly manner while performing their tasks. A novel methodology for ensuring safety during human–robot collaboration through planning and control is presented in [18]. Previous elicitation of safety requirements has identified three main approaches for eliminating the risk during HRI: (a) design the system from the beginning in order to mitigate the danger, (b) control the hazard through electronic or physical safeguards and (c) send signals to the users, either during operation or by training [4]. The goal of each safety analysis procedure, studied in previous literature, was to ensure a safe and human friendly interaction in an industrial environment, resulting in the imposition of safety measures, such as the utilization of system control and planning [6]. In addition to safety control, there are many safety standards, for instance ISO 10218, which states that a robot shall operate at slow speed mode when a human is present in a robot cell. However, technical and effectiveness features such as the size, configuration and environment of the robot to be guarded have been also proven to change the efficiency of the safety requirements of a corresponding analysis [5].

The related research mostly focuses on new ways to ensure the safety of human workers and limit the restrictions of a divided workspace. The production process in a smart factory is usually proposed to run without threats and interruptions, achieving the level of security and safety that meets worker safety legal requirements on the shop floor [19]. Symbiotic human–robot collaboration is defined for a fenceless environment in which productivity and resource effectiveness can be improved by combining the flexibility of humans and the accuracy of machines [20]. In [21], the development of a danger evaluation method is presented as an approach to analyze the factors which affect the potential impact force between the robot and the human, taking into account usability and performance features. The derived danger index is then used for improved mechanical design and control, which is often considered as the most effective safety strategy. Furthermore, according to [22], a requirement engineering analysis for industrial robots is necessary from the beginning, meaning the adaption of robotics system to needs and competences of the users, the specification of formal requirements models, the detailing of requirements and finally mapping them to system elements. Wiesner et al. [23] propose natural language processing (NLP) as a way to translate safety non-formal requirements to formal descriptions, thus enabling automated information processing and writing specifications by transforming requirements in natural language into discipline-specific models.

All these approaches have helped safety requirements for critical systems and industrial robots evolve and have made manufacturing intelligence a crucial topic for researchers and industries worldwide [24]. Despite these findings, exploring safety requirements interdependencies and their dependencies to others related to different aspects from the human safety perspective still remains a challenge. Therefore, in the context of smart manufacturing, an adequate requirement engineering analysis is considered to be the key to success or failure of every safe smart factory. Ensuring communication and consistency of different requirements is an interesting task owing to the variety of stakeholders from different sectors involved. While many efforts have been made to investigate safety requirements of industrial robots, there has not yet been such an analysis combined with a decision-making process in order to evaluate the dependencies and grade the importance of requirements and assign a corresponding weight to each of them. Toward this end, this paper tries to fill this gap in the literature by proposing an approach based on Pairwise Comparisons methodology, utilizing SysML as a representation language. Safety requirements, which address the continuously available manufacturing operation ability, are graphically modeled in a SysML diagram, so that their relationships are explicitly mapped and defined. The approach is illustrated by a list of requirements from the European project HORSE, which aims to develop a methodological/technical framework for easy adaptation of robotic solutions from SMEs.

In summary, the related work analysis of system safety requirements and symbiotic human–robot collaboration safety requirements in smart factories is presented in Table 1, including the proposed approach of this paper taking into account the most important contributions of the research.

3 Background: HORSE system and requirements

3.1 HORSE system

The EU project HORSE aims to bring a leap forward in the manufacturing industry proposing a new flexible model of smart factory involving collaboration of humans, robots, AGVs (autonomous guided vehicles) and machinery to realize industrial tasks in an efficient manner. The project proposes a smart integrated immersive and symbiotic HRC system controlled by the IoT based on dynamic manufacturing processes with weak emergent behaviors, since all of them have consistently reproduced in simulations of the system and could be easily understood through the reduced complexity of HORSE models during experiments and after observation, but not consistently predicted in advance. HORSE proposes a comprehensive set of activities to speed up adoption of emerging advanced manufacturing technologies of highly flexible and near-autonomous robotics systems. AGVs and static robots will be used to enable flexible and versatile production lines. These activities serve the entire value chain and propose new concepts and business models for robotic systems servitization and for product operation: HORSE defines and implements a technological framework that adopts novel information and communication technologies (ICT) approaches and standards (Open Service Gateway Initiative—OSGi) that enable the robots to be considered as centrally and remotely scheduled resources, dynamically allocated to new and varying production tasks in collaboration with humans in working cells without fences. This provides flexibility for fast configuration and take-up, improvement in quality (process control) and safety of the operator. The project aims to foster technology deployment toward SMEs by developing a methodological and technical framework for easy adaptation of robotic solutions and by setting up infrastructures and environments that will act as clustering points for selected application areas in manufacturing and for product life cycle management (production and/or maintenance and/or product end of life). More specifically, the novel approaches of HORSE are the integration of concepts such as (physical) human–robot interaction (HRI), intuitive human–machine interfaces and interaction between different robots and machines into an integrated environment with preexisting machines and workflows. Safety of the human worker as well as reduction in health risks through physical support

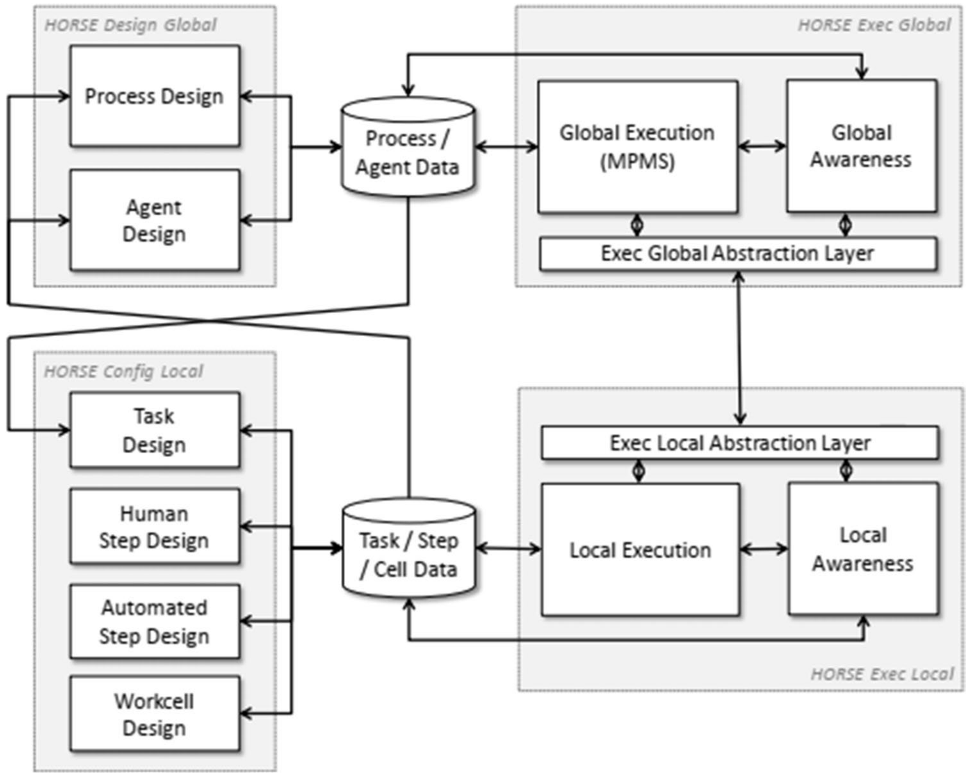

Fig. 1 Overall logical software architecture

by the robotized equipment will contribute to better overall manufacturing processes. In these, predefined workflows to be customized are the basis for servitization, for the entire value chain that allows rapid reconfiguration of the robots-based collaborative production processes. The purpose of this project is to foster advanced manufacturing technology deployment by industries and especially SMEs that will stimulate their interest [2].

In addition, the proper elicitation of the system requirements has a fundamental role for the specification of the HORSE system architecture which is acting as a blueprint for the implementation of the technical solution of the HORSE framework. The collection of requirements is really significant, so that they are analyzed, evaluated and prioritized, thus forming a configuration record defining the scope of the architecture and implementation work. The derived set of requirements is actually an agreement between identified stakeholders as to what will be considered a successful output of the HORSE project. It also represents the bare minimum of what the system must be able to satisfy with emphasis on the human safety feature [2].

The HORSE architecture is specialized for very small batch smart reconfigurable manufacturing systems. At first, a standard architecture is designed and then is implemented at the three different pilot cases of the project for verification and validation of the system. Figure 1 presents the overall logical software architecture, since the software aspect is leading in a system development project like HORSE. The integration of the architecture of four different subsystems is obvious. The HORSE software architecture is divided into a HORSE global level, which covers the site, area and production line levels of the hierarchy (as all these levels require coordination between work cells), and a HORSE local level, which covers the work cell level. Furthermore, the architecture distinguishes between support for design/configuration (Design/Config) of manufacturing activities on the one hand and execution (Exec) of manufacturing activities on the other hand—both on HORSE global and HORSE local levels. The database and connections to it together with interfaces to the hardware platform and human operators have been omitted for simplicity reasons [2].

This logical software architecture has been confronted with the general requirements following the requirements elicitation of the project, and it has been proven that not only all functional requirements are covered, but also all modules in this logical software architecture have a functionality linked to a same-level requirement; thus, the architecture is complete at this level and contains no superfluous modules [2].

Toward the direction of the development of the HORSE system components and their integration into an overall platform, it has been really important to identify stakeholders and their corresponding expectations and describe the different pilot cases based on which the HORSE system functions and their functional requirements are defined [25].

3.2 Requirement elicitation process

In the context of the HORSE project, the requirement elicitation process led to a list of requirements for the HORSE system, considering the usage of robots in a shop floor, involving HRC with no fences. The process of requirements elicitation was based on the state of the art and literature review about robotics in the industry, taking also into consideration the alignment with the Robotics 2020 Multi-Annual Roadmap from the euRobotics aisbl [26]. As a next level, these requirements were refined by studying the operations of the system at the pilot sites, identifying and analyzing the needs of each pilot case and classifying the typical use cases involving robotics in the production line. The requirements were matched with the technologies provided by the partners to define the actual functionalities of the HORSE framework [27]. The collected requirements were analyzed, evaluated and prioritized, thus forming a configuration record defining the scope of the architecture and implementation work. This evaluation includes the evaluation of the individual requirements against the project objectives and goals, technical feasibility, time, etc., leading to the final list of requirements, which were selected to be addressed by HORSE. The set of final requirements represents an agreement between identified stakeholders as to what will be considered a successful output of the HORSE project. It also represents the bare minimum of what the system must be able to satisfy. Upon agreement by identified stakeholders, the set of requirements were frozen to allow for harmonious system development [28].

The purpose of the HORSE system requirements elicitation was to provide a description of what the system should do and its interactions or interfaces with its external environment, capturing all inputs, outputs and required relationships between inputs and outputs in a way that does not bind the realization to a single product or technology. The proper specification of the system requirements has a fundamental role for the specification of the system architecture, which is acting as a blueprint for the implementation of the technical solution of the HORSE framework [29]. The designers and developers of HORSE platform components and functions aligned their work with the extracted requirements. These requirements act as criteria for evaluation of this work. The main goal of the design, implementation and execution of system test cases is the validation of the developed system against the set of requirements specified. Customized instances of the HORSE framework have been set up at the three pilot sites. The planning and the execution of the field tests also aim to validate the specific instances against the requirements and constraints elaborated. A traceability matrix based on the requirements specification was also developed before the implementation phase, since correct traceability is the basis for requirements analysis [17] which is important for all aspects of a software development project.

Figure 2 depicts the conceptual framework of the HORSE system and requirements. The HORSE system is decomposed into two main aspects: the integration of robotics and human activities, which drills down from the manufacturing process level to the level of the work cell that executes a task, and the integration of horizontal and vertical processes, which integrates the various vertical sub-processes (one for each work cell) with the horizontal end-to-end manufacturing process. Both aspects are explained and further decomposed into main functions and finally into specific operational requirements, as shown in Fig. 2 [2]. HORSE main functions present the general intended abilities of the HORSE system. These functions are based mostly on the expectations of the stakeholders, the technology to be developed and the feasibility of creating such technology. The main functions will drive system component development, to create technology which can provide the described functions. In the context of HORSE, taking into account the industry 4.0 needs [30], operational requirements [31] are those qualitative and

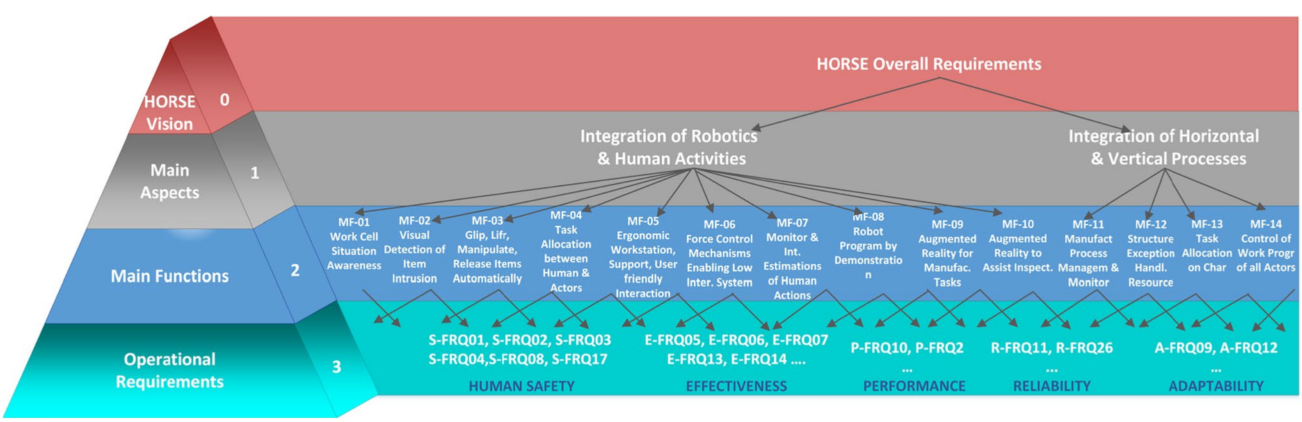

Fig. 2 The conceptual framework of HORSE system and requirements

quantitative statements that identify the essential capabilities, measurements (measures of effectiveness, performance, safety, reliability, adaptability, usability) and the process or series of actions to be taken in effecting the desired results of the system [32, 33]. They serve as a basis for determining the operational effectiveness and suitability of the system prior to deployment. Human safety requirements within HORSE in industry 4.0 are those requirements that are defined for the purpose of health risk reduction during the HRC [27]. This places emphasis on hazard identification and risk assessment in setting up robots and providing physical safeguards to separate robots from humans as much as possible to minimize the possibility of collision. The effectiveness requirements deal with how effective or efficient the HORSE system should be in performing its mission [34]. The effectiveness requirements were part of usability requirements of the project, in order to highlight the efficiency of operation and capture the efficiencies with which operators can exploit the services provided by the system [2]. Performance requirements refer to requirements that quantitatively measure the extent to which a system or a system part satisfy a required capability or condition.

Inspection of all these main aspects and functions of the HORSE system reveals that human safety is an important aspect in any step of the whole procedure. The human safety requirements seem to be entirely connected or even dependent on other requirements that affect robots' effectiveness and performance, so that the potential of robotics applications in new situations is to be fully realized, e.g., in SMEs and in service and domestic environments. HORSE is dedicated to provide HRC with an emphasis on worker safety as well as reduction in health risks through physical support by the robotized equipment that will contribute to improve processes' quality, reduce costs, enable flexibility and make better overall manufacturing processes. During the HORSE evaluation process of the operational requirements, apart from human safety, both effectiveness and performance requirements have been highly prioritized against other categories such as reliability and adaptability, thus forming a configuration record defining the focus of this paper not only on human safety, but also on these two categories. The outcomes of the evaluation process also revealed a great interconnection between human safety and effectiveness as well as performance operational requirements. Though we explicitly discuss the safety operational requirements related to the safety zone and the way they may be interrelated with operational requirements prescribing robot and human behavior in their operating zone, inside and outside the safety zone. The safety zone is defined as a special zone inside the operating zone, where the robotic actors operate in a safe way if a human is present [2]. This term is used as defined in directive 2009/104/EC concerning the minimum safety and health requirements for the use of work equipment by workers at work. Figure 3 depicts the HRI safety zone as well as the robot and actor operating zones. The safety requirements are the requirements inside the safety zone while requirements from other categories that are related to safety belong to either robot or actor operating zones. The operational requirements outside the safety zone may be requirements from other categories, but the project focuses on effectiveness and performance requirements, since they are highly ranked over the others. The HORSE project has to implement protective measures to reach acceptable risk level. User has also the responsibility to ensure that health and safety conditions are maintained. Uses of a flexible system pass through risk assessment and definition of new procedure for backup, reprogramming, quick machinery configuration. Then, again regulation is existing but robotic collaboration will create new technical problems which are today not in the common user field.

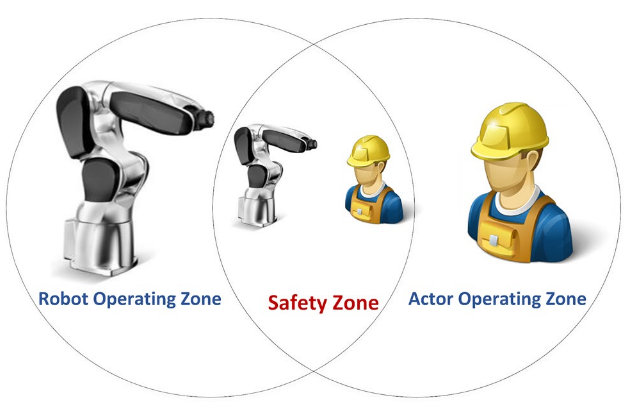

Fig. 3 HORSE safety zone

4 An approach to explore and prioritize human safety requirement dependencies

4.1 System requirements from human safety perspective

Ensuring human safety is a key requirement for all robots in symbiotic HRC systems [27]. Owing to their weight and the power required to move that weight rapidly and precisely, they can become quite formidable machines. Human safety is a key issue because without confidence that robots will not harm humans, their application and performance will remain limited [27–29]. The need for human safety applies to any kind of industrial type of robot in the manufacturing process. A basic requirement for service robots, for instance, is to ensure that they do not fall on or collide with the people they are supposed to be serving. This places emphasis on hazard identification and risk assessment in setting up robots and providing physical safeguards to separate robots from humans as much as possible to minimize the possibility of collision.

As a consequence, the human safety requirements seem to be entirely connected or even dependent on other requirements that affect robots' effectiveness and performance, so that the potential of robotics applications in new situations is to be fully realized. During the HORSE evaluation process of the operational requirements, apart from human safety, both effectiveness and performance requirements have been highly prioritized against other categories such as reliability and adaptability, thus forming a configuration record defining the focus of this paper not only on human safety, but also on these two categories. The outcomes of the evaluation process also revealed a great interconnection between human safety and effectiveness as well as performance operational requirements. In light of this, the present paper focuses on the human safety requirements of the HORSE system as well as the requirements from effectiveness and performance categories that are related to or may affect the human safety of the system and are presented in this section. In this context, we attempt to investigate the interdependencies of human safety requirements as well as the dependencies of requirements from the effectiveness and performance categories on safety requirements either directly or not. Toward this end, we adopt the HORSE terminology of operational requirements as mentioned in the above section.

Trying to capture the concept of dependencies of the human safety requirements, one may consider, for example, an effectiveness requirement E-FRQ05 that directly affects a safety requirement S-FRQ02 or a performance requirement P-FRQ10 that affects the effectiveness requirement E-FRQ05 which in turn affects S-FRQ02 (indirect dependency).

Table 2 contains a list of the collected requirements either functional or not, stated according to the following syntax, as advocated by [30]:

$$\mathbf{the}\ <\mathbf{system\ name}>\ \mathbf{shall}\ <\mathbf{system\ response}><\mathbf{optional\ preconditions}><\mathbf{optional\ trigger}>\cdots.$$

This simple structure forces the separation of the conditions in which the requirement can be invoked (preconditions), the event that initiates the requirement (trigger) and the necessary system behavior (system response). Preconditions and trigger are optional, depending on the requirement type. The order of the clauses in this syntax is also significant, since it follows temporal logic:

1. The system is required to achieve the stated system response if and only if the preconditions and trigger are true.
2. Any preconditions must be satisfied; otherwise, the requirement can never be activated.
3. The trigger must be true for the requirement to be "fired," but only if the preconditions were already satisfied.

4.2 Requirements dependencies problem statement

The aim of the present paper is to develop an approach in order both to identify and prioritize the dependencies of human safety requirements for smart integrated robotic systems in an attempt to provide the first step in developing a model-driven procedure which should be able to support the manufacturing process, facilitating the integration of systems and software modeling, which is increasingly important for robotic systems in smart factories incorporating HRC. This paper focuses on symbiotic HRC systems in the Industry 4.0 era, and exploring the dependencies of safety requirements with each other as well as with effectiveness and performance requirements has an important influence on software engineering activities, like project planning, architecture design and implementation phase.

Toward this end, a model-driven approach is introduced at first in this paper targeting the identification of safety requirements as well as the assessment of their interdependencies and the dependencies from requirements of other categories (effectiveness, performance) in order to assess their impact on human safety. More specifically, the dependencies are identified utilizing SysML requirement relationships. In this context, a SysML requirement diagram is constructed as the primary medium for conveying traceability among safety requirements as well as traceability from safety requirements to effectiveness and performance requirements in the system model. With any new addition of requirements to the model, new relationships from those requirements back to the existing ones that drove the need for their creation are created [35]. In this way, the establishment of requirements traceability becomes an ongoing activity throughout the design throughout the design and development of the system. Pairwise comparisons, a fundamental part of decision-making processes, are used in order to prioritize these safety requirements dependencies with each other and other critical requirements of smart factory operation, these of effectiveness and performance. The following subsections focus on the analysis of SysML and PWC approach.

4.3 SysML

A smart factory which involves collaboration of humans, robots and machinery is a fully connected and flexible

Table 2 System requirements related to/affect human safety requirements

Code	Title	Description	System function
S-FRQ01	Robot operation in safety zone	The *Robotic Actor* shall be able to operate in a defined space where a spatial zone of safe operation (*safety zone*) is programmed	MF-01
S-FRQ02	Robot human safety level monitoring	The *Robotic Actor* shall be able to monitor the safety level of all *Humans* inside its *operating zone*	MF-01
S-FRQ03	Dynamic set of safety zone	The *Robotic Actor* shall be able to dynamically set a *safety zone* immediately from detecting at least one *Human* inside its *operating zone*	MF-01
S-FRQ04	System human safety level monitoring	The HORSE system shall be able to monitor the safety level of all *Humans* inside the HORSE *operating zone*	MF-01
S-FRQ08	Work pieces and load consideration for safety contours	The HORSE system shall be able to take into account the work pieces and loads of the *Robotic Actors* while computing *safety zone* contours	MF-02
S-NR17	Not human harming	The HORSE system shall not harm humans	Non-functional
E-FRQ05	Simulation of robot movements	The HORSE system shall be able to simulate the *Robotic Actors'* movements for a task before executing it	MF-02
E-FRQ06	Collision-free planning	The *Robotic Actor* shall be able to plan collision-free manipulation trajectories	MF-02
E-FRQ07	Robot navigation	The *Robotic Actor* should be able to navigate avoiding obstacles and collisions, while it is equipped with autonomous mobility features	MF-02
E-FRQ13	Robot stiffness altering	The *Robotic Actor* shall be able to alter its stiffness immediately from detecting that its operation may endanger a *Human*	MF-06
E-FRQ14	Arm motion with force control	The *Robotic* Actor shall be able to support arm motion with force control in order to avoid damage of the products	MF-06
E-FRQ16	Robot motion altering	The *Robotic Actor* shall be able to alter its motion immediately from detecting that its operation may endanger a *Human*	MF-07
E-FRQ24	Info presented to operator	The HORSE system shall be able to present information to the *Operator* including at least (a) condition, (b) state and (c) alerts, for each *Robotic Actor* and *Production human actor*	MF-11
E-FRQ29	Actors reallocation	The HORSE system shall be able to re-allocate *actors*, in response to external events, including at least (a) safety alerts and (b) sensor failures	MF-12
E-FRQ32	Actors to tasks reallocation on safety risk	The HORSE system shall be able to dynamically re-allocate *actors* to tasks based on task safety risk and ergonomic information	MF-13
E-FRQ36	Safety risk notification	The HORSE system shall be able to accept notifications from *actors* in the production line regarding a change of manufacturing system status, including at least (a) actor availability and (b) safety risks	MF-14
P-FRQ10	Robot monitoring	The HORSE system shall be able to monitor for every *Robotic Actor*: (a)Its capabilities, including at least (a.1) maximum load and (a.2) availability, (b) Its performance, including at least (b.1) task actual completion time, (b.2) task estimated completion time, (b.3) task successful execution estimation	MF-04

system consisting of many different subsystems that can no longer be treated as stand-alone, but operate as part of a larger whole that includes other systems, robots or humans. The increase in its system complexity is demanding more rigorous and formalized systems engineering practices that differ from a document-based approach and rather move to a more model-based approach which focuses mainly on creating a coherent model of the system. Model-based systems engineering (MBSE) is proposed as a way to manage complexity, while it improves design quality and cycle time and facilitates knowledge capture and design evolution [36].

A standardized and robust modeling language is considered a critical enabler for MBSE. The Object Management Group's Systems Modeling Language (OMG SysML™) is a general-purpose graphical modeling language that supports the specification, analysis, design, verification and validation of a broad range of systems and is easily extendable. These systems may include hardware, software, data, people, facilities and procedures. SysML is a modeling language with a semantic foundation for representing requirements, behavior, structure and properties of the system and its components. It is an extension of the Unified Modeling Language (UML), version 2, which has become the de facto standard software modeling language [37, 38]. Into this context, SysML is also selected because it can provide modeling constructs, such as graphical diagrams to represent text-based requirements and relate them to other modeling elements or requirements of different categories. Most requirement relationships in SysML are based on the UML dependency. The direction of the arrows points from the dependent model element (client) to the independent model element (supplier). Hence in SysML, this is in the opposite direction that is often used to represent requirements flow-down, where the higher-level requirement points to the lower-level requirement. The direction represents a dependency from the derived requirement to the source requirement, such that if the source requirement changes, the derived requirement should also change [39]. Several requirements relationships are specified in SysML that enable the modeler to relate requirements to other requirements as well as to other model elements. These include relationships for defining a requirements hierarchy, relating requirements, deriving requirements, satisfying requirements, verifying requirements and refining requirements. Copy relationship is a dependency between a supplier requirement and a client requirement that specifies that the text of the client requirement is a read-only copy of the text of the supplier requirement. The relate dependency is used to define that a certain requirement is affected by another. The derive relationship relates a derived requirement to its source requirement. The satisfy relationship describes how a design or implementation model satisfies one or more requirements. The verify dependency defines how a test case or other model element verifies a requirement. Finally, the refine requirement relationship can be used to describe how a model element or set of elements can be used to further refine a requirement.

Since SysML is particularly effective in specifying requirements, structure, behavior, allocations and constraints on system properties to support engineering analysis, the present paper adopts it as a modeling language for describing the interrelations among requirements of the smart factory using symbiotic HRC, emphasizing on human safety perspective.

4.4 Pairwise comparisons

A fundamental problem in decision making is to grade the importance of a set of requirements and assign a weight to each of them. Their importance usually depends on several criteria which can be evaluated within the decision-making processes. In this paper, the PWC framework is used to evaluate the dependencies among the three categories of HORSE requirements, namely safety, effectiveness and performance from the human safety perspective.

Pairwise comparisons are widely used in multi-criteria decision analysis (MCDA) and have successfully been applied in many practical decision-making problems either as stand-alone method [40] or as an essential ingredient of MCDA processes, such as the AHP [41, 42], the weighted product method (WPM) [43], the preference ranking organization method for enrichment evaluation (PROMETHEE) [44] and the analytic network process (ANP) [45, 46]. PWC provides a structured process for the effective ranking of attributes, aiming at identifying their importance of influence on a general goal [45, 46]. The PWC framework enables the ranking of dependencies of requirements by allowing a number of experts, say M, to compare the various requirements R_i $(1 \le i \le N)$ in pairs, in order to explore their dependencies with the safety requirements, instead of assigning their dependencies in a single step [47]. This reduces the influence of subjective point of views, associated with eliciting weights directly.

We want to explore and prioritize the dependencies of requirements S-FRQ03, S-FRQ01, S-FRQ02, S-FRQ17 and S-FRQ04. We denote these safety requirements as S_k $(1 \le k \le 5)$ and the requirements with which are related (derive or relate relationship) as R_i. According to PWC, each expert m $(1 \le m \le M)$ compares all possible combinations of R_i and R_j, in order to explore which dependency is more strong (i.e., the R_i with S_k or the R_j with S_k). The outcome of these judgments for the mth expert is stored in a square $N \times N$ reciprocal matrix $\mathbf{P}^{(m)} = [P_{ij}^{(m)}]$, which will henceforth be referred to as a pairwise comparison matrix. Each $\mathbf{P}^{(m)}$ depicts the dependencies of the S_k requirement with the requirements R_i presented in the matrix. The value of the element $P_{ij}^{(m)}$ reflects the degree of the relation with S_k of requirement R_i over R_j. The experts need to complete only the upper triangular elements $(i < j)$ of $\mathbf{P}^{(m)}$ since by definition we have $P_{ij}^{(m)} = 1/P_{ji}^{(m)}$ and $P_{ii}^{(m)} = 1$ for a reciprocal matrix. The weights $w_i^{(m)}$ of requirement R_i according to expert m can be calculated with various ways. The most widely adopted approach is to solve the eigenvalue problem $\mathbf{P}^{(m)}\mathbf{x}_q^{(m)} = \lambda_q \mathbf{x}_q^{(m)}$, where λ_q are the eigenvalues of $\mathbf{P}^{(m)}$ and $\mathbf{x}_q^{(m)} = [x_{pq}^{(m)}]$ are the corresponding eigenvectors. Assuming that the eigenvalues are ordered so that λ_1 is the largest eigenvalue, then the weight of dependency of the requirement R_i with the S_k is estimated by normalizing the elements of the principal eigenvector $\mathbf{x}_1^{(m)}$ as follows [47, 48]:

$$w_i^{(m)} = x_{1i}^{(m)} \left[\sum_{l=1}^{N} x_{1l}^{(m)} \right]^{-1}. \tag{1}$$

In order to further simplify the comparisons, [41] introduced the nine-level scale shown in Table 3.

One way of measuring the inconsistency of a pairwise comparison matrix is to calculate the Consistency Ratio (C.R.) defined as C.R. = C.I./R.I., where C.I. $= (\lambda_1 - N)/(N - 1)$ is the consistency index and R.I. is an average random consistency index derived from a sample of randomly generated reciprocal matrices with elements scaled according to [41]. If C.R. is smaller or equal than 0.1 considered

Table 3 Nine-level scale

$P_{ij}^{(m)}$	Explanations
1	R_i and R_j are equally important
3	R_i is slightly more important than R_j
5	R_i is strongly more important than R_j
7	R_i is very strongly more important than R_j
9	R_i is absolutely more important than R_j
2, 4, 6, 8	Intermediate values
Reciprocals of above	Used in analogous manner when R_j is more important than R_i

acceptable and in this case, the matrix is said to be nearly consistent. In our case, the CR values were less than 0.1, which is considered acceptable and the matrices are considered to be consistent.

After all the comparisons have been completed, the average weight w_i for each R_i is calculated by averaging out the weights $w_i^{(m)}$ obtained by the M experts,

$$w_i = \frac{1}{M} \sum_{m=1}^{M} w_i^{(m)}. \tag{2}$$

The weights w_i are the weights of dependencies of the requirements R_i with the examined S_k, and hence the outcome of the PWC process.

In this paper, in order to rate the dependencies among the HORSE requirements from a human safety perspective, one must first indicate the different PWC matrices and then evaluate the weights of the dependencies of the requirements of each matrix (as analyzed in Appendix). Toward this end, each expert m performs a series of PWCs according to the aforementioned procedure and the weights of dependencies are finally estimated.

4.5 Surveys and participants

A number of $M = 15$ experts, members of the HORSE consortium have filled out the PWCs matrices. This group size is considered to be adequate for such decision-making problems, since it was shown in previous literature [48] that there is no much sense in using more than $M = 15$ participants, because the rate of decrease in an important measure named the probability of rank reversal of the final ranking is already small for $M > 15$. The survey was conducted by a Web-based decision support platform incorporating all elements of the PWC framework where experts log on to the platform and fill out the questionnaires. The Web platform has been developed in PHP as open source code by the authors and maintained in the Harokopio University of Athens. The open-source code has been also uploaded in Github (https://github.com/gdede-hua/decision-survey-platform) and supports the PWC framework as well as additional decision-making methods. The data supplied by the users are saved in a database, and the survey designer can perform the algorithm of pairwise comparisons in order to estimate the weights that signify the importance of categories according to the PWC framework.

The experts are employees of various organizations inside the HORSE project, which constitutes a well balanced blend between industry and academia from many parts of Europe (Netherlands, Germany, Spain, Slovenia, Poland, France and Greece). Their expertise lies primarily in the fields of requirement engineering, robotic systems, decision making and IoT. After the HORSE requirement specification and before the system implementation phase, one representative from each partner of the OMEGA consortium (i.e., Technical University of Eindhoven, Thomas Regout International, BOSCH, Harokopio University of Athens etc.) [2] participated in the pairwise comparison surveys, conducted during a period of 2 months between M5 and M6 of the HORSE project as depicted in the following Gantt Chart (Table 4). After the completion of the surveys, the data were analyzed and the dependencies have been ranked and analyzed during M7 and M8. The results of the proposed approach were very beneficial for the system design process and hence the system development, as they captured the requirements with highly ranked dependencies from the safety perspective. Given that human safety is the primary focus of the HORSE project, exploring safety dependencies has facilitated the implementation phase and gave key directions for the development of the HORSE system.

5 Results and discussion

5.1 SysML diagram and dependencies

The SysML requirement diagram depicting human safety requirements and their interdependencies is depicted in Fig. 4. Requirements are represented as SysML requirement

Table 4 Gantt chart depicting the phase of exploring safety dependencies

Phases	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	...	M24	...
Manufacturing and End-user Requirements														
Exploring Safety Requirements Dependencies														
System Design														
System Development														

elements, characterized by their unique identifier and text properties. The safety requirements are depicted as white boxes, while effectiveness and performance requirements as gray and yellow, respectively. Furthermore, the requirements relationships are specified in order to interrelate the safety requirements or relate them with requirements from the other two categories. The proposed model manages to confirm the results of the HORSE evaluation process of the selected set of requirements, since apart from safety, both effectiveness and performance categories have been highly prioritized and connected during the evaluation of the collected requirements. The study of the operations of the system at the HORSE pilot sites, the identification and analysis of the needs of each pilot case resulted in conclusions which were aligned with the extracted requirements and validate both the interrelations of the selected requirements of this model and the successful output of the HORSE project.

The estimated weights of dependencies, according to the performed PWCs, are also depicted. Inspection of the SysML diagram reveals that there are two main types of dependencies, "derive" and "relate." The "derive" relationship relates a derived requirement to its source requirement. This typically involves analysis to determine the multiple derived requirements that support a source requirement. On the other hand, "relate" dependency is used to define that a certain requirement is affected by another. For example,

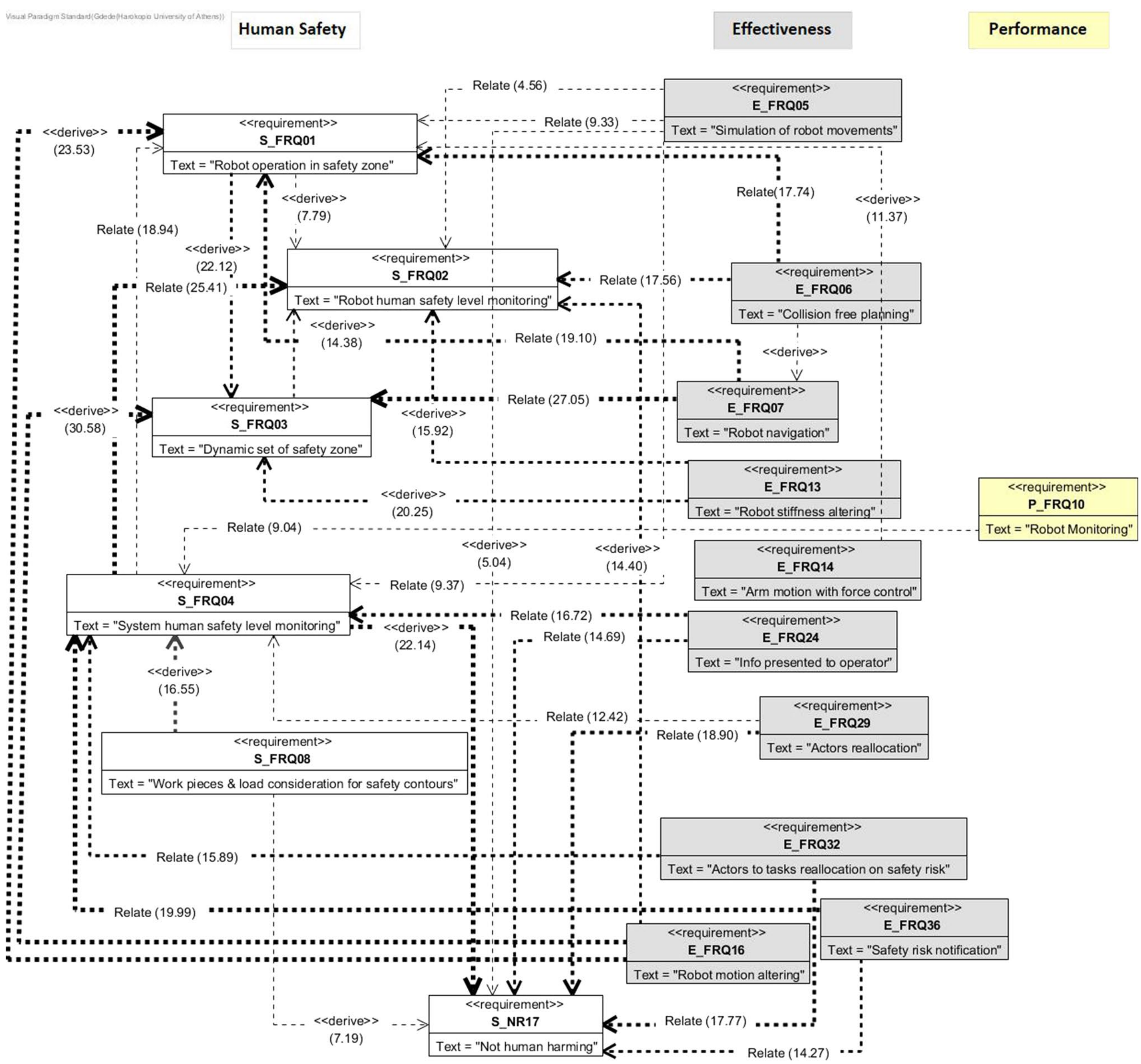

Fig. 4 SysML requirements diagram and dependencies

the dependency S-FRQ04 $\xrightarrow{\text{Relate}}$ S-FRQ01 denotes that the requirement S-FRQ04 influences S-FRQ01.

5.2 Prioritizing dependencies

In this section, the results of PWCs, regarding the evaluation of the dependencies among the requirements depicted in the SysML diagram, are presented and further analyzed. Figure 5 illustrates the weights of dependencies for each safety requirement, namely S-FRQ01, S-FRQ02, S-FRQ03, S-FRQ04 and S-NR17. As shown in Fig. 1, for each safety requirement the more important dependencies are highlighted with bold lines, whereas faded lines are used to depict the less important dependencies, according to the discussion of the results that is presented below. Since the paper is focused on exploring the dependencies from a safety perspective, the results below analyze the dependencies for each requirement lying on the safety category. Requirements S-FRQ08 is only derived from other requirements, but not related to others.

Considering S-FRQ03 (dynamic set of safety zone) dependencies, the results show that the E-FRQ16 (robot motion altering) is most strongly related to S-FRQ03, as the weight of "derive" relationship reaches 30.57%. It is rather clear that the robotic actor shall be able to alter its motion immediately from detecting that its operation may endanger a human, in order to ensure that the actor will be able to dynamically set a safety zone immediately from detecting at least one human inside its operating zone, something that comes in accordance with the Robotics 2020 Multi-Annual Roadmap from the euRobotics aisbl [23]. The "relate" dependency of E-FRQ07 (Robot navigation) comes second, rated with a high weight of 27%, indicating that the Robotic Actor should be able to navigate avoiding obstacles and collisions, while it is equipped with autonomous mobility features, in order that the actor to dynamically set a safety zone, when detecting a human. The "derive" dependencies of S-FRQ01 (Robot operation in safety zone) and E-FRQ13 (Robot stiffness altering) seem to have almost the same lower bearing of 22% and 20%, respectively, but not negligible however. This indicates that they are important requirements but the aforementioned requirements E-FRQ16 and E-FRQ07 seem to have a greater impact in order to ensure a dynamic safety zone setup.

As far as S-FRQ01 dependencies are concerned, according to Fig. 5, E-FRQ16 seems to have the most important relation accumulating weight of 23.52%. Robot motion altering is of paramount importance in order to ensure that the robotic actor shall be able to operate in a defined space where a spatial zone of safe operation is programmed. The effectiveness requirement E_FRQ07 and the safety requirement S-FRQ04 follow with almost the same weight of dependency of around 19%, whereas the effectiveness requirement E-FRQ06 is very closed with a dependency of 17.73%, indicating that robot navigation avoiding obstacles and collisions, system human safety level monitoring in the operating zone and planning collision-free manipulation trajectories have also a significant impact on robotic actor operation in a defined space with programmed safe operation zone. The rest effectiveness requirements (E-FRQ05, E-FRQ14) seem to be of lower importance.

Regarding the S-FRQ02 dependencies, according to Fig. 6, S-FRQ04 seems to take precedence over the dependencies with the other requirements with a weight of 25.41%, since system human safety level monitoring inside the HORSE operating zone is a strongly prerequisite

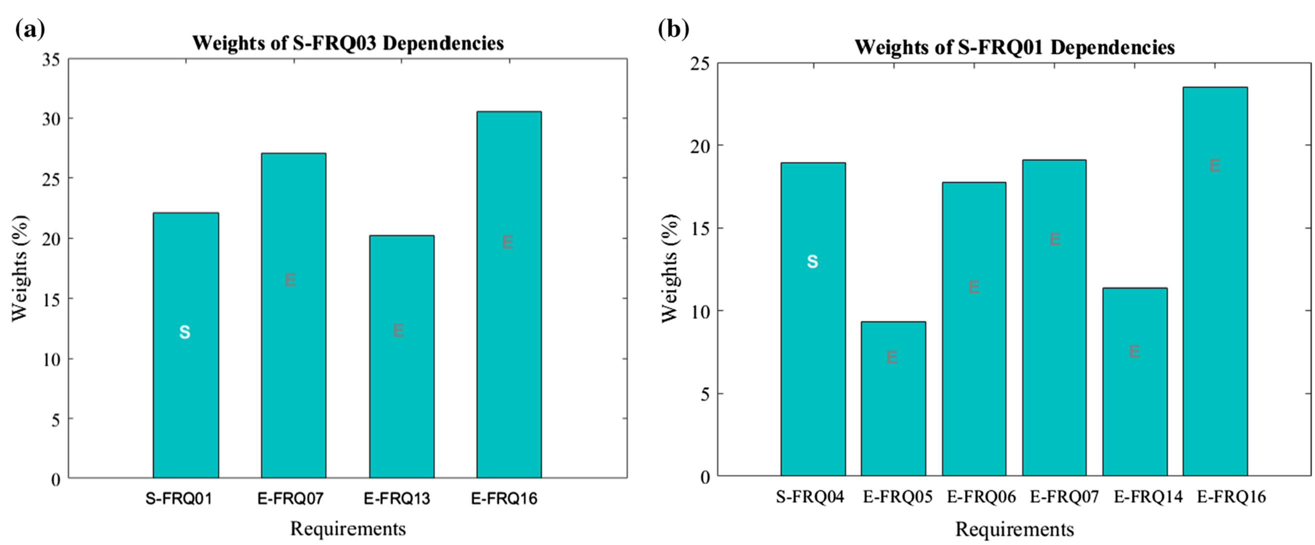

Fig. 5 Weights of dependencies for **a** S-FRQ03, **b** S-FRQ01

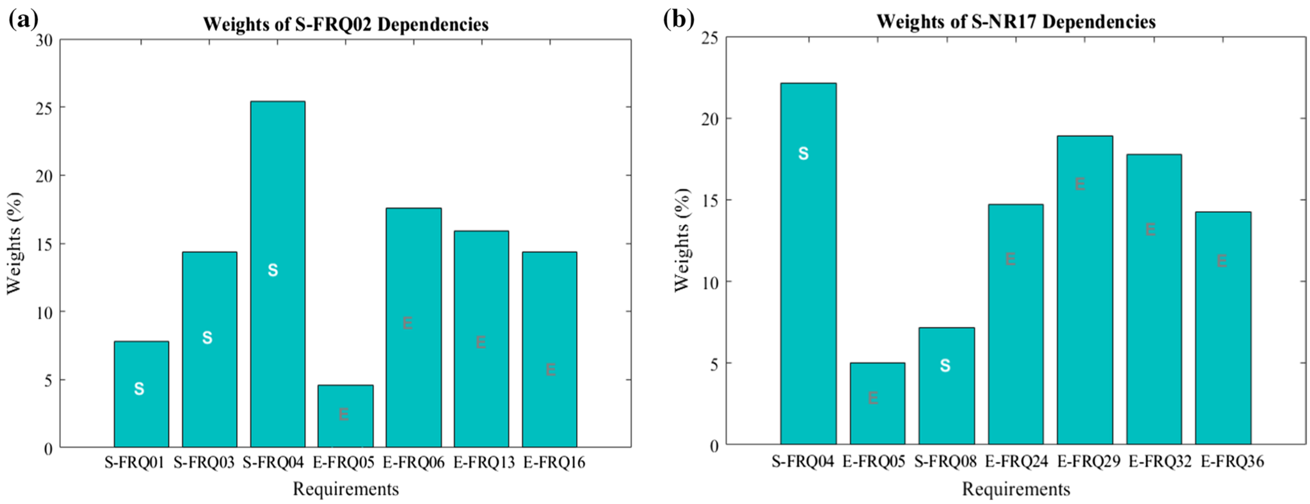

Fig. 6 Weights of dependencies for **a** S-FRQ02, **b** S-NR17

requirement for the robot to monitor the safety level of all humans inside its operating zone.

Concerning S-NR17 dependencies, as shown in the figure, the experts seem to strongly relate the safety requirement that the HORSE system shall not harm humans with the ability of the system to monitor the safety level of all humans inside the operating zone (S-FRQ04). Indeed, there is a strong interrelation of about 22.13% between the aforementioned safety requirements and it is rather obvious that requirements from effectiveness category seem to have a lower impact on ensuring that HORSE system will be safe for humans. However, the effectiveness requirement of system reallocation of actors including safety alerts (E-FRQ29) and the dynamic reallocation of actors to tasks based on safety risk (E-FRQ32) are the most dominant effectiveness requirements that affect S-NR17 (Fig. 7).

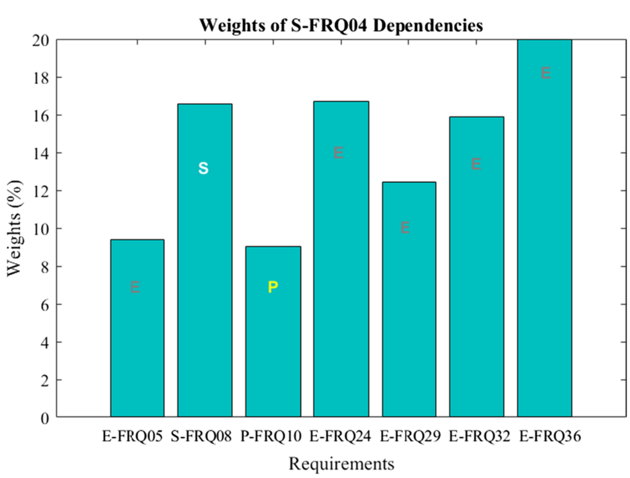

Fig. 7 Weights of dependencies for S-FRQ04

Finally, inspection of the results presented in the SysML diagram and Fig. 8 reveals that the effectiveness requirement of safety risk notification acceptance from actors regarding a change in manufacturing system status, including at least safety risks (E-FRQ36), is a requirement of paramount importance in order to ensure that the system will be able to monitor the safety level of humans in the operating zone (S-FRQ04). S-FRQ08 as well as the effectiveness requirements E-FRQ-24 and E-FRQ32 come second with almost equal importance of dependency around 16%, whereas the rest requirements seem to be related to lower impact dependencies with S-FRQ04.

It seems therefore that even though there are strong dependencies among the majority of safety requirements, effectiveness requirements also have a significant impact on safety, as depicted in the figures presented above. This is an important outcome since it highlights that safety is not ensured only by satisfying the safety-related requirements

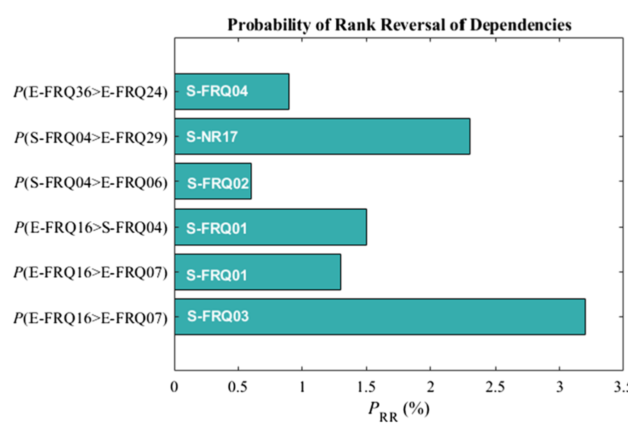

Fig. 8 P_{RR} of highly rated dependencies for each safety requirement

but also requirements from the effectiveness category that maybe a system designer or developer did not have in mind. Toward a successful system design and implementation, in the context of symbiotic HRC systems in the Industry 4.0 era, effectiveness requirements are aspects of paramount importance that have to be taken into account. Utmost care should be taken to fulfill not only critical safety requirements but also effectiveness requirements with highly ranked dependencies in order to ensure safe symbiotic human–robot collaboration systems. This is in accordance with the results of previous literature which revealed that effectiveness features have been proven to change the effectiveness of the safety requirement [5].

The aforementioned results also depict that the proposed approach can be useful to complement the requirements process, meaning to better define and trace requirements from the safety perspective, and it can therefore become a powerful managerial tool for decision makers and system participants in order to drive the safety of the system. The aim is to develop an approach in order to both model and explore the dependencies of safety requirements for smart integrated robotic systems and finally provide information that can be used as a guideline on where efforts are to be targeted and particular importance should be given during the design and implementation phase of the system. As the system is being developed with an emphasis on safety, all these requirements that have been assessed as significant with highly prioritized dependencies should be taken into account, whereas those with negligible ones have to be ignored since they do not significantly affect the rest of the process. Furthermore, paying attention to the great blend of different requirements of robotic systems, including safety, effectiveness and performance aspects is necessary for a complete requirements design process and analysis.

Toward this end, this study examines the safety requirements and explores their interdependencies as well as the dependencies from requirements of other categories in order to assess their impact on safety. The present approach develops the graphical SysML diagram of safety-related requirements defining their relations and prioritize their dependencies using the decision-making process of PWC. The results seem to be very beneficial for robotic system decomposition in the early system development activities. Since the requirements are sometimes conflict and incompatible, this approach may be very useful during the system design process to find the appropriate solution satisfying the majority of the requirements, giving a priority to the ones ranked with high dependencies and hence facilitating the production line. It seems therefore that this methodology can be seen as a step in developing a model-driven approach which should be able to support the manufacturing process, facilitating the integration of systems and software modeling, which is increasingly important for robotic systems in smart factories incorporating HRC. It is rather important that PWC may facilitate the integration of systems and software modeling, since it can be further applied in the community regardless of the requirement elicitation process used. Given a set of requirements, one can explore their dependencies applying the PWC, in order to investigate the impact of significant requirements to others and accordingly adapt the design and implementation process of the system.

5.3 Sensitivity analysis

The validity threats of the ranking outcomes of the dependencies deal with the decision-making process and more specifically with the PWC matrices filled in by the participants as well as with the uncertainty that may undermine their opinion. In this context, the CR index was estimated in order to examine the consistency and it was deduced that the judgments were consistent. The ranking outcomes of the dependencies were also further elaborated using sensitivity analysis and MC simulation. In this section, we discuss the reliability of the results, given the level of uncertainties involved, by carrying out a sensitivity analysis. It was found that the priorities of the dependencies are not significantly influenced by the uncertainties that may undermine the judgments of the experts, which enhance the validity and accuracy of the results.

We use MC simulation to estimate the effect of introducing random perturbations in all parameters of the decision-making process. More specifically, MC simulations were carried out by randomly varying the elements of the PWC matrices used in the estimation of the dependencies' weights. Such random perturbation may be due to inconsistencies of the PWC matrices [39]. Assuming the PWC matrices $\mathbf{P}^{(m)}$ filled out by the experts, we estimated the intervals $O_{ij} = [P_{\mathrm{ij}}^{(\min)}\ P_{ij}^{(\max)}]$ by calculating $P_{ij}^{(\max)} = \max\{P_{ij}^{(m)} | 1 \le m \le M\}$ and $P_{ij}^{(\min)} = \min\{P_{ij}^{(m)} | 1 \le m \le M\}$, where M is the number of experts involved in the surveys. In each MC iteration, we created M random matrices $\Delta\mathbf{P}^{(m)} = [\Delta P_{ij}^{(m)}]$ by randomly selecting $\Delta P_{ij}^{(m)}$ from a uniform distribution inside O_{ij}.

Carrying out 10^4 iterations, we estimated the probabilities of rank reversal P_{RR} [38] between the dependencies of each safety requirements and the P_{RR} was less than 4% for all the cases. In this context, the P_{RR} of the most prominent dependencies for each safety requirement is depicted in Fig. 8. As illustrated in the figure, we define as P(E-FRQ36 > E-FRQ24) the probability of rank reversal between the dependencies E-FRQ36 and E-FRQ24 for the safety requirement S-FRQ04 and the rest probabilities depicted are defined in a similar manner. The figure indicates that the P_{RR} remains sufficiently low (less than 3.5%) for all the cases. The outcomes provide an indication of the reliability of the PWC results against uncertainties in the PWC carried out by the participants.

6 Conclusion

Since symbiotic HRC systems will play a key role in the IoT era, there is an increased demand for robot safety standards and requirements during not only the design but also the implementation stage of such a system. Given the fact that human workers are now able to have access to the robot work space during operation, human safety is an important consideration in HRI in order to prevent accidents. Therefore, an approach is proposed to complement the requirements process and better define and trace the requirements from the human safety perspective by exploring their interdependencies as well as their dependencies with other categories, these of effectiveness and performance. Toward this end, the present paper aims to develop an approach to identify, explore and prioritize the dependencies of human safety requirements for symbiotic HRC systems. This approach may become a powerful managerial tool for decision makers and system participants for driving the general human safety of each system examined.

The proposed approach is based on SysML for the representation of the requirements dependencies and the decision-making process PWC as well, to assess and explore these dependencies. More specifically, SysML is used as a language for creating a coherent model of the system in order to represent text-based requirements and relate them to effectiveness and performance requirements emphasizing on their human safety perspective. The model depicts in a SysML diagram the safety requirements as well as their interdependencies and the dependencies with effectiveness and performance requirements, and therefore, their impact on human safety is assessed. The decision-making method PWC is selected to explore the dependencies and grade their importance. This model-driven approach is used as the primary medium for conveying traceability among human safety requirements as well as traceability from safety requirements to effectiveness and performance requirements in the system model. The dependencies among the human safety, effectiveness and performance requirements, as identified in the European project HORSE, are evaluated from the human safety point of view, and the results seem to be very beneficial for robotic system decomposition in the early system development activities. For a stronger validation of the stability of the final outcomes, the authors have performed a sensitivity analysis and the inspection of the results reveals that the dependencies' priorities are not significantly affected by the uncertainties and may undermine the experts' judgments.

The proposed methodology of this paper may be used as a step in developing a model-driven approach which should be able to support the manufacturing process, facilitating the integration of systems and software modeling, which is increasingly important for robotic systems in smart factories incorporating HRC. As the system is being developed with an emphasis on human safety, all these requirements that have been assessed with highly prioritized dependencies should be taken into account, whereas those with negligible ones have to be ignored since they do not significantly affect the rest of the process. In addition, paying attention to the great blend of different requirements of the robotic systems, including safety, effectiveness and performance aspects, is necessary for an effective design process and analysis of the system. Since the requirements are sometimes conflict and incompatible, this approach may be very useful for other systems during the system design process to find the appropriate solution satisfying the majority of requirements, giving a priority to the ones ranked with high dependencies and hence facilitating the production line. It is rather important that this approach may facilitate the integration of systems and software modeling, since it can be further applied in the community regardless of the requirement elicitation process used. Given a set of requirements, one can explore their dependencies applying the PWC in order to investigate the impact of significant requirements to others and accordingly adapt the design and implementation process of the system.

As a future research direction, it seems very interesting to apply the proposed methodology to an existing project with a rich collection of human safety and safety-related requirements and finally compare the outcome with the already reported results. Furthermore, the extension of the approach by defining requirements and evaluating their dependencies to requirements of different, additional categories from a different perspective other than human safety seems to be a challenging topic for further research and study.

Acknowledgements The research leading to these results has received funding from the European H2020-FoF-2015 Project "Smart Integrated Robotics System for SMEs Controlled by Internet of Things Based on Dynamic Manufacturing Processes (HORSE)" (680734).

Appendix

Questionnaire

Requirements definition

In this section, Table 1 of Sect. 4.1 is presented to the experts in order to understand the requirements and their description, before proceeding to the questionnaire. Moreover, the SysML diagram without the weights is also given to understand the requirements relationships.

Questionnaire Completion

Table 3 of Sect. 4.4 is also given to the experts in order to understand the nine-level scale and fill in the pairwise comparison matrices.

How to complete the questionnaire

The following questionnaire aims at prioritizing the **dependencies between the requirements** in order to evaluate their importance. For example, if S-FRQ01 depends on S-FRQ02 and S-FRQ03, we have to evaluate the importance of the dependencies in order to examine whether S-FRQ02 affects more S-FRQ01 than S-FRQ03 or the opposite.

Toward this end, you have to compare the requirements in pairs of two (pairwise comparisons) by allocating a value from the nine-level scale presented in Table 2. Please read carefully Table 3 (nine-level scale) and Table 1 (brief description of requirements) in order to complete the questionnaire.

Making the following pairwise comparisons, please allocate a number from the nine-level scale at each box. You compare the requirement presented in each **row** with all the other requirements presented in the **columns**, keeping in mind which requirement has more or less strong dependency and how much to the requirement that they affect.

For example, we know that both S-FRQ02 and S-FRQ03 requirements affect (derive/relate) the requirement S-FRQ01, then if we compare the S-FRQ02 with S-FRQ03 and put in the box the value 3, we mean that S-FRQ02 slightly affects more than the S-FRQ03 the requirement S-FRQ01.

Pairwise comparison for **S-FRQ01** Requirement

	S-FRQ03
S-FRQ02	3

Questionnaire

Pairwise Comparison for S-FRQ03-Dynamic set of safety zone “Dependencies”.

	E-FRQ07-Robot navigation	E-FRQ13-Robot stiffness altering	E-FRQ16-Robot motion altering
S-FRQ01-Robot operation in safety zone			

	E-FRQ13-Robot stiffness altering	E-FRQ16-Robot motion altering
E-FRQ07-Robot navigation		

	E-FRQ16-Robot motion altering
E-FRQ13-Robot stiffness altering	

Pairwise Comparison for S-FRQ01-Robot operation in safety zone "Dependencies".

	E-FRQ05-Simulation of robot movements	E-FRQ06-Collision free planning	E-FRQ07-Robot navigation	E-FRQ14-Arm motion with force control	E-FRQ16-Robot motion altering
S-FRQ04-System human safety level monitoring					

	E-FRQ06-Collision free planning	E-FRQ07-Robot navigation	E-FRQ14-Arm motion with force control	E-FRQ16-Robot motion altering
E-FRQ05-Simulation of robot movements				

	E-FRQ07-Robot navigation	E-FRQ14-Arm motion with force control	E-FRQ16-Robot motion altering
E-FRQ06-Collision free planing			

	E-FRQ14-Arm motion with force control	E-FRQ16-Robot motion altering
E-FRQ07-Robot navigation		

	E-FRQ16-Robot motion altering
E-FRQ14-Arm motion with force control	

Pairwise Comparison for S-FRQ02-Robot human safety level monitoring " Dependencies".

	S-FRQ03-Dynamic set of safety zone	S-FRQ04-System human safety level monitoring	E-FRQ05-Simulation of robot movements	E-FRQ06-Collision free planning	E-FRQ13-Robot stiffness altering	E-FRQ16-Robot motion altering
S-FRQ01-Robot operation in safety zone						

	S-FRQ04-System human safety level monitoring	E-FRQ05-Simulation of robot movements	E-FRQ06-Collision free planning	E-FRQ13-Robo stiffness altering	E-FRQ16-Robo motion altering
S-FRQ03-Dynamic set of safety zone					

	E-FRQ05-Simulation of robot movements	E-FRQ06-Collision free planning	E-FRQ13-Robot stiffness altering	E-FRQ16-Robot motion altering
S-FRQ04-System human safety level monitoring				

	E-FRQ06-Collision free planning	E-FRQ13-Robot stiffness altering	E-FRQ16-Robot motion altering
E-FRQ05-Simulation of robot movements			

	E-FRQ13-Robot stiffness altering	E-FRQ16-Robot motion altering
E-FRQ06-Collision free planning		

	E-FRQ16-Robot motion altering
E-FRQ13-Robot stiffness altering	

Pairwise Comparison for S-NR-17-Not human harming "Dependencies".

	E-FRQ05-Simulation of robot movements	S-FRQ08-Work pieces & load consideration for safety contours	E-FRQ24-Info presented to operator	E-FRQ29-Actors reallocation	E-FRQ32-Actors to tasks reallocation on safety risk	E-FRQ36-Safety risk notification
S-FRQ04-System human safety level monitoring						

	S-FRQ08-Work pieces & load consideration for safety contours	E-FRQ24-Info presented to operator	E-FRQ29-Actors reallocation	E-FRQ32-Actors to tasks reallocation on safety risk	E-FRQ36-Safety risk notification
FRQ05-Simulation of robot movements					

	E-FRQ24-Info presented to operator	E-FRQ29-Actors reallocation	E-FRQ32-Actors to tasks reallocation on safety risk	E-FRQ36-Safety risk notification
S-FRQ08-Work pieces & load consideration for safety contours				

	E-FRQ29-Actors reallocation	E-FRQ32-Actors to tasks reallocation on safety risk	E-FRQ36-Safety risk notification
E-FRQ24-Info presented to operator			

	E-FRQ32-Actors to tasks reallocation on safety risk	E-FRQ36-Safety risk notification
E-FRQ29-Actors reallocation		

	E-FRQ36-Safety risk notification
E-FRQ32-Actors to tasks reallocation on safety risk	

Pairwise Comparison for S-FRQ04- System human safety level monitoring "Dependencies".

	S-FRQ08-Work pieces & load consideration for safety contours	P-FRQ10-Robot Monitoring	E-FRQ24-Info presented to operator	E-FRQ29-Actors reallocation	E-FRQ32-Actors to tasks reallocation on safety risk	E-FRQ36-Safety risk notification
E-FRQ05-Simulation of robot movements						

	P-FRQ10-Robot Monitoring	E-FRQ24-Info presented to operator	E-FRQ29-Actors reallocation	E-FRQ32-Actors to tasks reallocation on safety risk	E-FRQ36-Safety risk notification
S-FRQ08-Work pieces & load consideration for safety contours					

	E-FRQ24-Info presented to operator	E-FRQ29-Actors reallocation	E-FRQ32-Actors to tasks reallocation on safety risk	E-FRQ36-Safety risk notification
P-FRQ10-Robot Monitoring				

	E-FRQ29-Actors reallocation	E-FRQ32-Actors to tasks reallocation on safety risk	E-FRQ36-Safety risk notification
E-FRQ24-Info presented to operator			

	E-FRQ32-Actors to tasks reallocation on safety risk	E-FRQ36-Safety risk notification
E-FRQ29-Actors reallocation		

	E-FRQ36-Safety risk notification
E-FRQ32-Actors to tasks reallocation on safety risk	

Example of dependency computation

The estimation of dependencies is based on the PWC procedure described in Sect. 4.4. We want to explore and prioritize the dependencies of requirements S-FRQ03, S-FRQ01, S-FRQ02, S-FRQ17 and S-FRQ04. We denote these safety requirements as S_k $(1 \le k \le 5)$ and the requirements with which are related in terms of relate and derive relationship as R_i. Toward this end, each expert m from a group of M experts fills in the PWC matrices mentioned in the above section of "Appendix" in order to explore the dependencies of each S_k requirement mentioned in the title of the PWC matrices. Each PWC matrix depicts the dependencies of the S_k requirement with the requirements presented in the matrix. Each of the aforementioned PWC matrices, filled in by the mth expert, corresponds to the $\mathbf{P}^{(m)}$ matrix of the PWC process. The estimated weights $w_i^{(m)}$ of the matrix (according to Eq. 1) are the dependency of the requirement R_i of the mth expert with the related S_k requirement of each PWC matrix. Then, the average weights w_i for the M experts are estimated, based on Eq. (2). The weights w_i define the weights of dependencies of the requirements R_i with the related S_k.

For example, we consider the first PWC of the questionnaire, namely the matrix depicting the dependencies of the requirement S_1, namely S-FRQ03. We want to find the dependencies with the requirements S-FRQ01, E-FRQ07, E-FRQ13 and E-FRQ16. These requirements are the R_i requirements of the PWC process, where $1 \leq i \leq 4$. Each expert $1 \leq m \leq \mathrm{M}$ fills in this matrix, and hence, we have M PWC matrices for the dependencies of S-FRQ03. We consider each of these matrices as $\mathbf{P}^{(m)}$. For each $\mathbf{P}^{(m)}$, we apply the eigenvalue method and we estimate the weights $w_i^{(m)}$ (according to Eq. 1) which is the dependency of the requirement R_i of the mth expert to the S_1. We then estimate the average of these weights, say w_i, based on Eq. (2). The weights w_i are the weights of dependencies of the R_i to the S_1. These are the weights depicted in the SysML diagram (Fig. 4) as well as in Figs. 5, 6 and 7.

The procedure is the same for the exploration and prioritization of the dependencies of the other requirements.

References

1. Grefen P, Vanderfeesten I, Boultadakis G (2016) Supporting hybrid manufacturing: bringing process and human/robot control to the cloud (short paper). In: 2016 5th IEEE international conference on cloud networking (cloudnet). https://doi.org/10.1109/cloudnet.2016.39
2. Home | HORSE. http://horse-project.eu/. Accessed 18 Apr 2019
3. Sikora E, Tenbergen B, Pohl K (2012) Industry needs and research directions in requirements engineering for embedded systems. Requir Eng 17(1):57–78
4. Kulić D, Croft EA (2006) Real-time safety for human–robot interaction. Rob Auton Syst 54(1):1–12
5. Vasic M, Billard A (2013) Safety issues in human–robot interactions. IEEE
6. Fryman J, Matthias B (2012) Safety of industrial robots: from conventional to collaborative applications. VDE
7. Westman J, Nyberg M (2019) Providing tool support for specifying safety–critical systems by enforcing syntactic contract conditions. Requir Eng 24(2):231–256
8. Allenby K, Kelly T. Deriving safety requirements using scenarios. In: Proceedings fifth IEEE international symposium on requirements engineering. https://doi.org/10.1109/isre.2001.948563
9. Hansen KM, Ravn AP, Stavridou V (1998) From safety analysis to software requirements. IEEE Trans Softw Eng 24(7):573–584. https://doi.org/10.1109/32.708570
10. Wang J (1997) A subjective methodology for safety analysis of safety requirements specifications. IEEE Trans Fuzzy Syst 5(3):418–430. https://doi.org/10.1109/91.618277
11. Kotronis C, Nikolaidou M, Dimitrakopoulos G, Anagnostopoulos D, Amira A, Bensaali F (2018) A model-based approach for managing criticality requirements in e-health IoT systems. In: 2018 13th annual conference on system of systems engineering (SoSE). https://doi.org/10.1109/sysose.2018.8428764
12. Firesmith D (2004) Engineering safety requirements, safety constraints, and safety–critical requirements. J Obj Technol 3(3):27. https://doi.org/10.5381/jot.2004.3.3.c3
13. Firesmith D (2004) Prioritizing requirements. J Obj Technol 3(8):35. https://doi.org/10.5381/jot.2004.3.8.c4
14. Zhang W, Mei H, Zhao H (2005) A feature-oriented approach to modeling requirements dependencies. In: 13th IEEE international conference on requirements engineering (RE'05). https://doi.org/10.1109/re.2005.6
15. Anderson S, Felici M (2001) Requirements evolution from process to product oriented management. In: Product focused software process improvement, pp 27–41. https://doi.org/10.1007/3-540-44813-6_6
16. Winkler S, von Pilgrim J (2010) A survey of traceability in requirements engineering and model-driven development. Softw Syst Model 9(4):529–565. https://doi.org/10.1007/s10270-009-0145-0
17. Zhang H et al (2014) Investigating dependencies in software requirements for change propagation analysis. Inf Softw Technol 56(1):40–53. https://doi.org/10.1016/j.infsof.2013.07.001
18. Kulić D, Croft E (2007) Pre-collision safety strategies for human–robot interaction. Auton Robots 22(2):149–164
19. Thoben K-D, Wiesner S, Wuest T (2017) 'Industrie 4.0' and smart manufacturing—a review of research issues and application examples. Int J Autom Technol 11(1):4–16
20. Wang L, Törngren M, Onori M (2015) Current status and advancement of cyber-physical systems in manufacturing. J Manuf Syst 37:517–527
21. Ikuta K, Ishii H, Nokata M (2003) Safety evaluation method of design and control for human-care robots. Int J Rob Res 22(5):281–297
22. Geisberger E, Broy M (2012) agendaCPS: integrierte forschungsagenda cyber-physical systems, vol 1. Springer, Berlin
23. Wiesner S, Gorldt C, Soeken M, Thoben K-D, Drechsler R (2014) Requirements engineering for cyber-physical systems. Springer, Berlin
24. dos Santos J, Martins LEG, de Santiago VA Junior, Povoa LV, dos Santos LBR (2019) Software requirements testing approaches: a systematic literature review. Requir Eng 14:247
25. Ellis-Braithwaite R, Lock R, Dawson R, King T (2017) Repetition between stakeholder (user) and system requirements. Requir Eng 22(2):167–190
26. Aisbl EUR (2014) Robotics 2020 multi-annual roadmap n for robotics in Europe, call 1 ICT23–horizon 2020. Initial Release B 15.01 (2014)
27. Grefen P, Vanderfeesten I, Boultadakis G. Developing a cyber-physical system for hybrid manufacturing in an Internet-of-Things context. In: Advances in business information systems and analytics, pp 35–63. https://doi.org/10.4018/978-1-5225-3805-9.ch002
28. Vanderfeesten ITP, Erasmus J, Grefen PWPJ (2018) The HORSE project: IoT and cloud solutions for dynamic manufacturing processes. In: 5th European conference on service-oriented and cloud computing presented at the ESOCC 2016, pp 303–304
29. Modugno F, Leveson NG, Reese JD, Partridge K, Sandys SD (1997) Integrated safety analysis of requirements specifications. Requir Eng 2(2):65–78
30. RIoT Control (2017) https://doi.org/10.1016/c2011-0-06998-3
31. Operational requirements, August 2013
32. Bartneck C, Kulić D, Croft E, Zoghbi S (2009) Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. Int J Soc Robot 1(1):71–81
33. Kossiakoff A, Sweet WN (2002) Systems engineering principles and practice. https://doi.org/10.1002/0471723630
34. Berenbach B, Paulish D, Kazmeier J, Rudorfer A (2009) Software & systems requirements engineering: in practice. McGraw Hill Professional, New York
35. Filiopoulou E, Mitropoulou P, Tsadimas A, Michalakelis C, Nikolaidou M, Anagnostopoulos D (2015) Integrating cost

analysis in the cloud: a SoS approach. In: 2015 11th international conference on innovations in information technology (IIT). https://doi.org/10.1109/innovations.2015.7381554
36. Friedenthal S, Moore A, Steiner R (2014) A practical guide to SysML: the systems modeling language. Morgan Kaufmann, Burlington
37. OMG Systems Modeling Language (OMG SysMLTM) (2015) Version 1.4
38. Hause M (2009) OMG systems modeling language (OMG SysML™) tutorial. In: INCOSE international symposium, vol 19, no 1, pp 1840–1972. https://doi.org/10.1002/j.2334-5837.2009.tb01046.x
39. Roques P (2015) Modeling requirements with SysML. Requir Eng Mag, no. 2015–02
40. Dede G, Kamalakis T, Sphicopoulos T (2016) Theoretical estimation of the probability of weight rank reversal in pairwise comparisons. Eur J Oper Res 252(2):587–600. https://doi.org/10.1016/j.ejor.2016.01.059
41. Saaty TL (2003) Decision-making with the AHP: why is the principal eigenvector necessary. Eur J Oper Res 145(1):85–91. https://doi.org/10.1016/s0377-2217(02)00227-8
42. Dede G, Kamalakis T, Varoutas D (2011) Evaluation of optical wireless technologies in home networking: an analytical hierarchy process approach. J Opt Commun Netw 3(11):850. https://doi.org/10.1364/jocn.3.000850
43. Supriyono H, Sari CP (2018) Developing decision support systems using the weighted product method for house selection. https://doi.org/10.1063/1.5042905
44. Brans JP, Vincke P, Mareschal B (1986) How to select and how to rank projects: the Promethee method. Eur J Oper Res 24(2):228–238. https://doi.org/10.1016/0377-2217(86)90044-5
45. Triantaphyllou E (2000) Multi-criteria decision making methods: a comparative study. Appl Optim. https://doi.org/10.1007/978-1-4757-3157-6
46. Yager RR (2004) Modeling prioritized multicriteria decision making. IEEE Trans Syst Man Cybern Part B (Cybern) 34(6):2396–2404. https://doi.org/10.1109/tsmcb.2004.837348
47. Saaty TL, Vargas LG (2001) Models, methods, concepts & applications of the analytic hierarchy process. Int Ser Oper Res Manag Sci. https://doi.org/10.1007/978-1-4615-1665-1
48. Dede G, Kamalakis T, Sphicopoulos T (2015) Convergence properties and practical estimation of the probability of rank reversal in pairwise comparisons for multi-criteria decision making problems. Eur J Oper Res 241(2):458–468. https://doi.org/10.1016/j.ejor.2014.08.037

Affiliations

Georgia Dede[1] · Persefoni Mitropoulou[1] · Mara Nikolaidou[1] · Thomas Kamalakis[1] · Christos Michalakelis[1]

Persefoni Mitropoulou
persam@hua.gr

Mara Nikolaidou
mara@hua.gr

Thomas Kamalakis
thkam@hua.gr

Christos Michalakelis
michalak@hua.gr

[1] Department of Informatics and Telematics, Harokopio University, Kallithea, Greece