

Article

Not peer-reviewed version

Modeling Requirements for Collaborative Robotic Services

Oscar Stiven Morales Zapata , Yaney Gomez Correa , [Leopoldo Rideki Yoshioka](#) , [Jose Reinaldo Silva](#) *

Posted Date: 21 September 2023

doi: 10.20944/preprints202309.1365.v1

Keywords: Systems Design; Requirements Engineering; Collaborative Robots; Human-robot interaction; value co-creation



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Modeling Requirements for Collaborative Robotic Services

Oscar Stiven Morales Zapata ¹, Yaney Gomez Correa ¹, Leopoldo Rideki Yoshioka ^{1*}
and Jose Reinaldo Silva ^{1,*}

¹ Escola Politécnica de São Paulo, São Paulo, Brazil

* Correspondence: reinaldo@usp.br

Abstract: Collaborative robots have experienced low acceptance in applications, especially in industry. This fact has attracted the attention of researchers and practitioners, who point to different causes for this low acceptance. One of the main reasons is the difficulty in converging on suitable methods for modeling collaborative interactions between robots and their surrounding context during the requirements phase. These interactions must be elicited and modeled during requirements modeling to maximize value creation through collaboration and must be formally verified, taking into account the risks of human-robot interaction. However, such modeling is often not present in collaborative robot design, and the choice of an appropriate approach remains an open problem. In this paper, this problem is addressed from a requirements engineering perspective, a goal-oriented model and a service-based approach supported by an additional verification method based on Petri nets is proposed. A case study based on collaborative robots used in a hospital environment is presented.

Keywords: systems design; requirements engineering; collaborative robots; human-robot interaction; value co-creation

1. Introduction

According to Federico Vicentini, collaborative robots “is an umbrella term that conveys the general idea of proximity between machines and humans for a useful task” [32]. A definition that summarizes the current trend to generalize the use of collaborative robots in different application domains [17] from manufacturing [7,19] to health applications [31], including RPA (Robotics Process Applications) [2,3]. However, widening the scope of human-robot to a generic cognitive human-machine interaction would raise a challenge on how to design these systems as *human-in-the-loop* [16][15].

Therefore, even the terminology must be revised to take a more comprehensive understanding of the robotic elements beyond its functionality, focusing on the value that collaboration with humans or other machines will add. The value co-creation added by collaboration must be treated as the focus for the robotic and human system components instead of the result of the composition of individual machines and humans in a hybrid network [8] [25].

There is a convergent view among researchers and practitioners about collaborative automated systems¹. Its primary design factors - inherited from the design of collaborative robots - are: actuation, control, physical interaction, usability, and productivity [24,32]. To this list, it would be added also the perceptron, the increased ability to sense the state of the environment, and especially collaborative goals [14]. Perceptron’s capabilities also enlarge the theoretical background of collaborative design to include cognitive models and artificial intelligence.

Together, all these features fit the service concept derived from Service Science. In an advanced approach, an automated service provider is an artifact that depends on its coupling with a service client to achieve value creation collaboratively. A provider is designed to offer a service similar to a collaborative automated system, which aims to couple in a hybrid network of humans and machines.

¹ This generalized terminology to cover the widening of the scope in collaborative robotic systems.

The only difference is the nature of the provider-client coupling, which is more robust in service systems.

Service Science [21] emerged with IBM researchers about 2004 and since then was conceptually and scientifically developed, gaining new topics such as Service Engineering, specifically its application to robotics [4]. Service Engineering is a derivation of this development, looking for methods for requirements modeling and formalization [27]. Some authors proposed an approach based on objectives (Goal-Oriented Requirements)[9], revisited here as an alternative to provide a human-centered method for collaborative automated systems (a generalization of collaborative robots). The proposed model-based method generates a model-based requirement cycle formalized in Petri Nets - a suitable representation to model the behavior and communication of a network of automata.

The following section will introduce collaborative robot design methods, highlighting their advantages and weaknesses. Section 3 will present the service-oriented approach proposal for the requirements modeling cycle, including the user coupling (where the term *user* is applied to any collaboration component, even if not human). Section 4 presents a case study based on mobile robots that work in a hospital. Section 5 will show some analysis of the general proposal and detaches the modeling of human-robot interaction during requirements. Section 6 will point to further general developments and the case study.

2. DESIGN METHODS FOR COLLABORATIVE ROBOTS

Collaborative robots or “cobots” are core pillars for the future of factories and complex humans task. The combination of robot characteristics (repeatability and tirelessness) and operator (or user) experience improves productivity and can be critical to the success of automated processes. A Human-robot collaboration (HRC) is a growing area of robotics research where humans can work side by side with robots. Cobots are more flexible, perform different tasks, and are easy to install, program, and reprogram, even by non-experts. Existing technologies for sensing and communication [33] incorporate security devices, such as laser sensors and vision systems, for recognizing gestures and human expressions or systems that allow collision avoidance or voice command recognition while diversifying the number of tasks to be executed. Human safety is one of the most important considerations within the field of human-robot interaction, preventing collisions between humans and robots operating within a shared space, besides considering all possible ways to avoid harming a person [11]. However, besides safety, issues such as ergonomics and productivity must be considered in the design.

Two major approaches for robotic task execution have emerged [5]: path/motion planning (hardly applicable to human-robot collaboration because of the unpredictable and dynamic nature of humans) and sensor-based control (more efficient and flexible for human-robot interaction, since it closes the perception-to-action loop at a lower level than path/motion planning). Recent developments in measurement technologies have made sensors affordable and lightweight, easing their use on robots.

Different combinations of sensing modalities consider four robot senses [5]: *vision*, *touch*, *audition*, and *distance*. Control techniques used for HRC in the industrial sector [19] structured a classification addressing five main problems: *collision avoidance*, *collision detection*, *motion planning*, *control system design*, and *scheduling*. Due to the increase in complexity and functionalities, robots use Artificial Intelligence techniques to solve tasks and learn [8].

2.1. Requirements for modelling Collaborative Robots

The human factor sets dynamics requirements that change during the design process that should be revised during maintenance or innovative improvements. Cobots often rely on special-purpose hardware and operating software and attend requirements directed toward specific purposes. Consequently, specifications for user interactions could not be explicitly modeled, resulting in further loss of flexibility and design reusability. Additionally, bringing intelligence to the project requires tightening robotic systems’ sensing, processing, and acting capabilities.

In this scenario, the specification of requirements plays a crucial role in embodying robotic system functionalities into a proper and complete model [1]. An approach for the requirements in any system domain is unsuitable due to the diversity and complex interactions involved, including the surrounding environment, human agents, machines and automated systems. There are some attempts to map generic features, even if missing a formal verification, that mention difficulties like *Dealing with uncertainty, defining appropriate models, integration of RE (Requirements Engineering) Models, technique selection, requirements reuse, synchronization between requirements and architecture, addressing NFR (Non-Functional Requirement), conflicting and ambiguous requirements*. In practice, different applications present a different subset of those difficulties.

According to Wang [33], the relationship involving humans and robots in a shared domain can be summarized in *coexistence* (when a robot and a human are placed within the same physical space but without overlapping each other's workspace); *interaction* (when a human and a robot sharing the same workspace communicate with each other, and the human and the robot work on the same task stepwise, in sequential order); *cooperation* (developed among human and robot agents who have their autonomy, expressed in terms of goals, objectives, utility or profit), *collaboration* (when there is a joint activity of humans and robots in a shared workspace to accomplish together a set of given working tasks). Different collaborative interactions between human-robot should be defined (during requirements specification) and modeled in an initial phase, especially if it could be formally verified.

In this paper, a goal-oriented approach is proposed to models human-robot interactions with humans and a surrounding environment to achieve goals defined by collaboration requirements. The model obtained can be modeled by goal-oriented service-based diagrams, transferred to XML, and verified using Petri Nets formalism.

3. A Requirements Cycle in the Design of Collaborative Automated Systems

Goal-oriented (service) requirements engineering (GORE) implies identifying service objectives according to the needs and desires of users and stakeholders extracted through an elicitation process [23]. Primary goals are refined into sub-goals or requirements down to the lowest level, where each sub-goal or requirement is assigned to an agent or another system (hardware or software). GORE is considered a valuable tool in analyzing requirements and alternative decision-making processes, which makes the service approach more efficient and reliable.

Some of the reasons why objectives are relevant in RE processes are [12]:

- (i) Achieve requirements integrity. Objectives (goals) provide precise criteria in specifying requirements.
- (ii) Avoid irrelevant requirements. Objectives provide a precise criterion on the pertinence of the requirements; a requirement is relevant for a set of objectives in the considered domain if its specification is used to prove at least one objective.
- (iii) Explain requirements to interested parties. Objectives provide a rationale for requirements similar to design goals in design processes.
- (iv) Purpose refinements provide a natural mechanism for structuring complex requirements documents, essential for better readability.
- (v) Requirements engineers deal with many relevant alternatives during requirements modeling. Analysts responsible for decision-making can be involved in validating the options chosen to suggest other ignored or forgotten alternatives.
- (vi) Alternative goal refinements allow optional proposals for the system to be explored.
- (vii) Manage conflicts between multiple viewpoints. Objectives can be recognized as providing the roots for detecting conflicts between requirements that must be solved before designing solutions.

Model-based Requirements Engineering (MBRE) is characterized by identifying and structuring requirements in models, recognizing partial or incomplete requirements, and facilitating communication between users and suppliers (in the case of services). In other words, MBRE contributes

to model-oriented requirements modeling and analysis, proposing a cycle that goes from eliciting and identifying requirements to their subsequent formalization and verification, closing the cycle with an analysis of the evolution of the process, as shown in Figure 1.

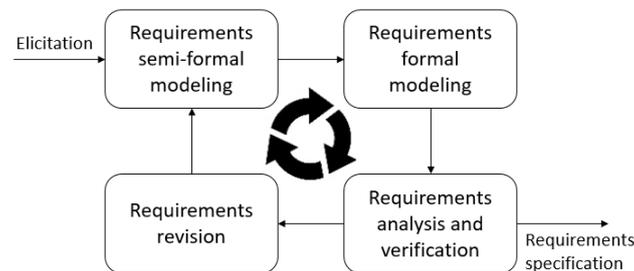


Figure 1. Requirements life cycle [29]

Service design has a human-centered approach due to its iterative exploration, ideation, reflection, and implementation process. Typically the inception process promotes user and stakeholder participation in the co-creation of solutions [13]. Therefore, the main challenge relies on searching for new solutions or implementing a better set of sub-processes that satisfy the requirements while leading to better performance. Frequently, different processes demand greater flexibility in the tasks developed. The parties involved (suppliers and consumers) should be aware of objects or agents (people, machines, resources, among others) that can change the system's state. Therefore, service design is responsible for modeling, analyzing, and specifying design processes that effectively cover a set of goals called service requirements [17].

Value co-creation is the final goal of service systems, mainly when based on collaboration, based only on human-robot interaction, or including other automated systems. Recently, value co-creation gained more momentum because of the underlying logic of organizations [20]. Cover tools and procedures are developed to support new techniques applied to different service domains, such as robotic restaurants, where robots interact, assist and serve customers in activities such as hosting, taking orders, delivering food, and cleaning.

In the value co-creation process, the consumer is the actor in focus. The co-creation process involves active interaction between the user and the product-service provider. The value generated by this interaction cannot be obtained by the action of any of the parts separately and is associated with the satisfaction of the system's objectives and expectations. The different interactions between users and services that generate values will be modeled, in this paper, using the operations diagram of the KAOS method.

Goal-oriented requirements are modeled diagrammatically using KAOS (Knowledge Acquisition Object System) since the requirements elicitation phase and proceed during the cycle depicted in Figure 2. Figure 2 has two additional modules, unlike Figure 1, which refer to the user-supplier interaction process. These models have an influence on the elicitation of requirements, since this stage is marked by direct communication between the different parties. Thus, each time the cycle is completed, the user will have an active participation in the decisions that may affect the product-service, ensuring that the user is no longer seen as a static user (end user), but as a dynamic user who participates in each new stage of the product. The basic elements of KAOS are shown in Figure 3.

KAOS representation, Figure 3, comprises four related (but not redundant) modeling diagrams: goal, object, operation, and responsibility. One of the advantages of KAOS is to eliminate the dichotomy between modeling functional and non-functional requirements. Unlike functional methods, such as UML, KAOS uses a reduced set of diagrams, including restrictions, expectations, and conflicts between elements, and can deal with large packages of subsystems.

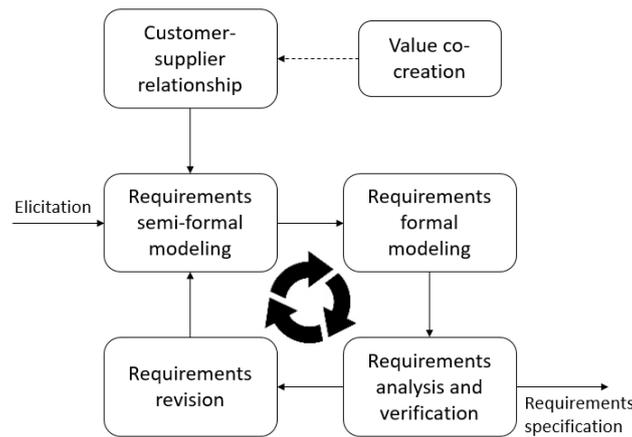


Figure 2. Proposal for modeling and requirements analysis

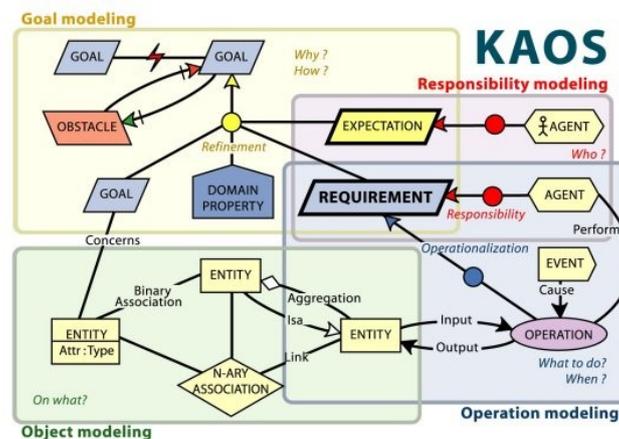


Figure 3. KAOS diagram

Semi-formal models for requirements can be formalized into a dynamic (discrete) presentation language Petri Nets, a method with a strong mathematical base that allows to represent different situations that appear in discrete dynamic systems, that captures all interaction among agents and the system [17,29,31]. The formal model is analyzed and verified to check for the invariants' preservation, consistency, and closeness. A revision process will look for gaps that justify enhancing the requirement's initial model.

Working with requirements can be intensive work if performed manually. Thus, once the requirements are modeled, they are documented in *eXtensible Markup Language* (XML), called *KAOS Markup Language* (KML) and proposed by the authors, format for storage and later transferred to Petri Nets.

The proposed method models user-provider (human-robot) interaction, relying on KML. The KML approach can provide the necessary attributes - in a hierarchical and structured way - to describe all the relationships between the objectives, objects, responsibilities, and operations diagrams.

The general structure of an KML document (proposed by the authors) is composed of two parts:

1. Prolog: Contains a sequence of processing instructions and/or document type declaration. It can be divided into two parts:
 - KML declaration. Defines the kml version, the encoding type and if it is a independent document. `<?kml version="1.0" encoding="ISO-8859-1"?>`
 - Document type declaration. Defines the type of document it is. (Optional)
2. Body: It is the information content of the document, organized as a single tree of marked elements.

```

<kml>
  <Element>
    <Model>....</Model> <!--Refers to the 4 possible models of KAOS (Goals, Objects,
Responsibility, Operational)-->
      <Id>....</Id> <!--Unique identifier for each element of the diagram -->
      <Name>....</Name> <!--Name of the element in the diagram-->
      <Type>....</Type> <!--Diagram elements type: (Goals, Subgoals, Requirements,
Expectations, Obstacles, Domain, Agents, Object, Events, Operations, Refinements)-->
      <Pattern>....</Pattern> <!--Pattern of the element: Maintain (□), Achieve (◇)-->
      <Class>....</Class> <!--Classification for requirements: Functional,
Non-Functional <!--Classification for agents: Stakeholders, Final users, System agents-->
      <Precondition>....</Precondition> <!--For operation type.-->
      <Postcondition>....</Postcondition> <!--For operation type.-->
      <Refinements> <!--Refinements between elements-->
        <ToRefineAnd>...</ToRefineAnd> <!--Refinements between elements of Goal
diagram-->
        <InRefineAnd>...</InRefineAnd> <!--Refinements between elements of Goal
diagram-->
        <ToRefineOr>...</ToRefineOr> <!--Refinements between elements of Goal
diagram-->
        <InRefineOr>...</InRefineOr> <!--Refinements between elements of Goal
diagram-->
        <ConflictTo>....</ConflictTo> <!--Refinements between elements of Goal
diagram-->
        <Resolution>....</Resolution> <!--Refinements between elements of Goal
diagram-->
        <Obstrution>....</Obstrution> <!--Refinements between elements of Goal
diagram-->
        <ExpectationOf>....</Expectation
Of> <!--Refinements between elements of Goal diagram-->
        <ConcernsTo>....</ConcernsTo> <!--Refinementsbetween elements of Object
diagram-->
        <AssociateTo>....</AssociateTo> <!--Refinements between elements of Object
diagram-->
        <IsA>....</IsA> <!--Refinements between elements of Object diagram-->
        <AssignedTo>....</AssignedTo> <!--Refinements between elements of
Responsibility diagram-->
        <ResponsabilityOf>....</Responsa
bilityOf> <!--Refinements between elements of Responsibility diagram-->
        <CauseTo>....</CauseTo> <!--Refinements between elements of Operational
diagram-->
        <InputTo>....</InputTo> <!--Refinements between elements of Operational
diagram-->
        <OutputTo>....</OutputTo> <!--Refinements between elements of Operational
diagram-->
        <PerformanceOf>....</Performance
Of> <!--Refinements between elements of Operational diagram-->
        <OperationalizationOf>....</Opera
tionalizationOf>
        <!--Refinements between elements of Operational diagram-->

```

```

</Refinements>
<Graphics> <!--Graphic characteristics of elements-->
  <Height>....</Height>
  <Widtht>....</Widtht>
  <Position>
    <x>....</x>
    <y>....</y>
  </Position>
</Graphics>
</Element>
</kml>

```

This structure is stored in a document with an extension .kml, that could be read and interpreted favoring an automatic transfer between the modeled requirements and the Petri nets. The proposed KML2PNML transfer (Figure 4) starts with a user-supplier interaction process that when modeled in KAOS diagrams (semi-formal language) generates a KML file. KML file is converted into a PNML file to formalize the elicited requirements. To develop this proposal, due to the lack of tools, VBA and Python programming languages were required. A part of the developed code is stored in a public repository on GitHub <https://github.com/osmz/KML2PNML>.

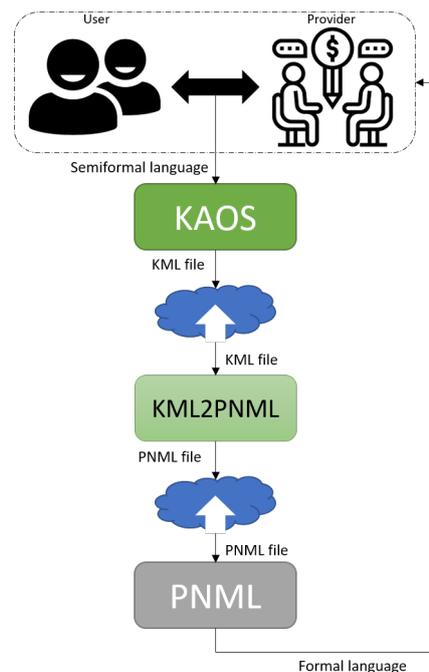


Figure 4. KML2XML transfer proposal

3.1. The Goal-oriented Service Approach

The goal-oriented treatment for requirements engineering has been discussed since the late 90s and is used in different academic and commercial tools. MBRE aims to add traceability and reusability to improve diagram analysis and validation and introduced its use to design service networks. Therefore, it includes migrating the whole process to the cloud [28].

To strengthen the analysis and verification of requirements, the XML description of KAOS diagrams called KML was proposed, as discussed in the previous section. The proposed method improves the documentation and reuse of services, this issue allows the pairing of the diagrams with a

formal representation, the Petri Nets (frequently used in automation projects), described by an XML representation called PNML [10]².

Therefore, stating that the Goal-Oriented Service Approach is suitable for modeling Collaborative Robotics requirements, is a possibility that is found to be justified whose main characteristics can be summarized as follows:

- (i) Using goal-oriented modeling is suitable to describe a service network architecture involving different automation systems and collaborative robots.
- (ii) The method focuses on all interactions among services and especially in the relation human-robot, responsible for the value co-creation.
- (iii) The proposal revises KAOS diagrams to improve service network requirements and design.
- (iv) Developed an algorithm to transfer KAOS diagrams to KML, which supports documentation, reusability, and the requirements cycle described in Figure 2.
- (v) Developed an algorithm to transfer KML to PNML to get a formal model in Petri Nets (Place/Transition). The formal model verifies the intended behavior of a service network architecture and the human-robot interaction.

Synthesizing, the proposed method starts from a reverse engineering process that aims to identify a series of initial requirements (elicitation process), to throw pass to the development of the relevant KAOS diagrams (generating a sytem-as-is). The obtained diagrams can reveal new suggestions of requirements or solutions to possible conflicts observed, in which case the respective modifications must be made, generating a more complex and complete model (system-to-be). Once the analysis and validation with the user are completed, the requirements are formalized (supported by the properties of the Petri nets) as will describe in the following section.

4. A Case Study of Collaborative Robots in a Hospital Environment

The role of collaborative robots was enlightened during the pandemic, where the logistics of hospitals and medical centers were affected by the need for isolation and extra care to avoid contamination. It was clear that besides consuming human support, a large amount of “services” could be managed by collaborative machines and sometimes by regular service-delivering automation.

The term service has a specific meaning, taken from the Service Science, related to the interaction carried out by a minimum of two agents: the service provider and the service consumer. In some situations, the same agent can occupy a duplicated mission, for instance, managing and controlling smart-grid systems based on co-production. In such cases, the service consumer is also a producer [18]. A collaborative robot working in a hospital environment has different categories of service consumers but deals with one at a time. These consumers are clearly defined and do not perform a dual role simultaneously. A case that can also be observed is the fact that the robot in a certain case and depending on the working environment must deal with automated systems (machines) - which reinforces the demand for flexibility and careful design [22] to arrive at a diversified collaborative interaction.

As a case study, a robot deployed by Professor Yoshioka (co-author of this paper) based on functionality will be taken, without consideration of collaborative modeling in the requirements phase. The focus was on control and functional approach. Therefore, the goal (objectives) related to collaboration and coupling with service consumers were mixed with control instead of being used to guide the interaction process. The prototype works very well, but attention needs to be paid to the outcome of the co-creation, as well as to the diversity of situations in which the robot could intervene in a critical environment such as a hospital.

² PNML is part of Petri Net Standard ISO/IEC 15.909, adopted since 2004

The project is a partnership between the Innovation Center of the University Hospital³ and the College of Engineering. The idea was to have a flexible, autonomous robot to automate logistic demands in a hospital, transporting items from place to place or from the automated pharmacy to a destination. The robot could be quickly sanitized after going to sensitive regions concerning contamination. The robot's mission would be to transport medication or goods (small equipment and medical resources, including personal protective equipment (PPE)).

To clarify the difference in approach, the legacy system requirements of the original project must first be captured to then build a revised version. The intention is not to make a direct comparison, as the requirements follow different approaches, but to focus on the architecture of the service network, especially the modeling of collaboration.

4.1. The system-as-is: a functional robot

The collaborative robot has as functional specification a mobile system weighing 40 kg, capable of carrying up to 20 kg. It is 120 cm high and can move with a maximum of 5 km/h. Figure 5 shows the robot's layout with open and closed doors for the cargo compartment. The robot has doors that open and close automatically, high-capacity rechargeable batteries, sensors for the perception of the environment, a wireless communication system, a high-performance hardware platform, and navigation software based on Artificial Intelligence (AI).



Figure 5. The collaborative mobile system functionally designed and implemented.

Concerning collaboration, the robot has to interact with humans and also with an automated system that dispenses medicines automatically. As mentioned, delivering medicines from the internal pharmacy is one of the most demanding actions in the hospital processes. This process is fully automated, as shown in Figure 6.



Figure 6. The automated system to dispense pharmacos.

The hospital automation environment is integrated through Wifi communication with the Automated Logistic Center. Thus, a user can request medication from the automated pharmacy

³ The University Hospital at University of São Paulo is capable to attend 258 patients, besides the emergency unit.

and deliver it using the robot service or request that an item be transported from one plane to another in the hospital. If the robot is not busy with another mission, it goes to the pharmacy (or the pickup point), which internally prepares to transfer the requested medication to the robot compartment. The robot should deliver the cargo (the medication or an item) to a specified delivery place. All mission data is stored in the supervision center for further analysis.

Figure 7 captures the above design using the goal-oriented requirements method described in Section 3. At that point one of the advantages of MBRE should be mentioned and that is the ability it offers analysts to enter and identify requirements. In the obtained system-as-is it is observed that "Hospital Logistic Automation" is divided by two objectives "Collaborative Robot" and "Automatic Medication Dispenser", each objective presents some characteristics such as: Movement, Collision Detection, Avoid collision and Planning. Objectives that are refined by certain requirements and finally attributed/responsible to certain agents. There is a conflict related to the failure in the delivery of the order, and three actions that will depend on the user (expectations).

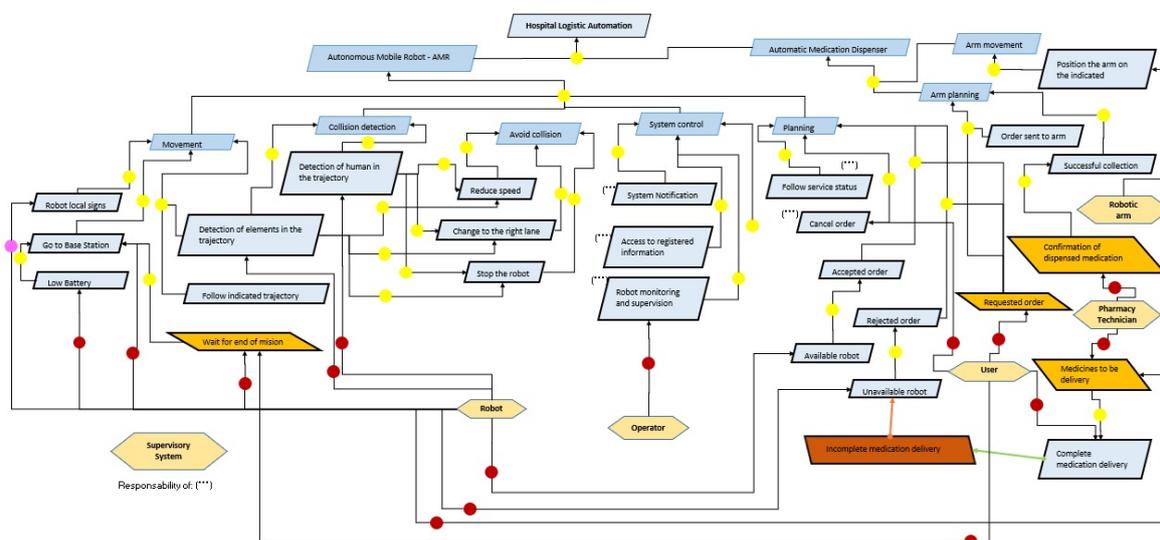


Figure 7. Goal-oriented model for the original project.

All objects, either active (which can change the system state) or passive (resources and material), appear explicitly in the goal diagram (Figure 7). Passive elements could be physical (as the medicine to be delivered from the pharmacy or the items to be) or virtual - as the requests or confirmation to dispense medicines. Active objects (agents) are artificial (the collaborative robot, the robotic arm of the dispenser), virtual (the supervisory system, the control in the automatic dispenser), or human (the final user, the operator of the logistic center, the operator in the pharmacy). Collaboration means different journeys involving two or more of these agents.

The goal model depicts the relationship and dependencies among the objectives (goals). Therefore, even following the original specification strictly, it focuses on the relationship with the (human) agents requesting the transportation of equipment, PPE, or other medical resources except for medication. Consequently, the (human) user journey does not appear explicitly in the former design, nor the goal mode generated.

The following section will show a different approach focusing on the (human) user and its journey to clarify the difference in approaching collaborative automated projects using MBRE.

4.2. The system-to-be: introducing human-robot collaboration

In the MBRE approach, a project may start from scratch or include enhancements or revisions to a project under development, have a prototype or a deployed solution. Figure 7 reflects a goal-oriented diagram extracted from the project's functional requirements specification. The requirements review

reinforces the separation between three services: the collaborative robot, the Logistics Automation Center, and the Automated Medication Dispenser (service network). In this architecture, the Logistics Automation Center interacts with the (human) user by receiving and acknowledging requests, which are also stored for future flow and demand analysis, as originally specified.

The service interaction between the Logistic Center and the Collaborative Robot is based on sending *mission plans* - derived from the requests - and receiving acknowledgments about the mission accomplishment.

A *mission plans* is composed of:

- (i) A *pre-plan* requesting the robot to move from its current position to the point where it should get some item to deliver.
- (ii) The *plan*, fitting classic AI definition, where the robot follows the trajectory between the pickup and delivery point, avoiding obstacles and managing emerging occurrences (such as being out of power).
- (iii) The *post-plan*, regarding the path the robot has to follow to a base station where it does not interfere with the hospital workflow and logistics.

However, the focal point is how to approach collaborative robots design when they have to integrate an environment populated by humans and machines using the service network approach. Requirements specifications should reflect the integration, as it occurs in systems design [6]. Collaboration would benefit from a service-oriented approach, mainly if it is expected to cover further innovations or maintenance. Finally, the application of a goal-oriented service approach allows specifying, during the design phase, the coupling and value co-creation that arise when dealing with humans and other machines, characteristics found in MBRE.

Based on the original project requirements, a goal-oriented diagram called system-as-is was modeled and shown in Figure 7, a revision of these requirements resulted in the system-to-be (Figure 8), corresponding to a service-oriented network. The overall interaction of the automated logistics system is centered on the operator agent and the user (by receiving and handling requests). A human user interacts with the robot in specific transactions concerning the loading/unloading of the items to be transported and the encounter with the robot during the execution of the plan, in a situation where the human is a privileged obstacle.

The goal diagram corresponding to the system-to-be (Figure 8) was substantially reduced compared to the original system-as-is (Figure 7). Due to the implementation of the service network, which offers the possibility to make use of certain targets in one action as well as in another, based on the theory that the processes are the same. It should be noted that with such a distribution it was possible to generate a better refinement of the identified objectives, leading to a better understanding of what is desired and expected from the product-service. Another major difference is the integration of "Supervisory Sys" and "Supervisory System" in the system-to-be. However, the comparison should take into account that the review started from a model extracted from functional specifications, characterized by mixing functionality, resources, actions and conditions, which undoubtedly influenced the objectives model, while other diagrams represent operations, objects and responsibility.

The revision aims to:

- (i) Identify and separate services (Autonomous Mobile Robot - AMR, Supervisor control system and Automatic Medication Dispenser) to focus on service-oriented architecture.
- (ii) Collapse functionalities, actions, rules and conditions into targets (including possible non-functional requirements). Objectives are understood as those aspects identified in the user-supplier interaction that must be satisfied by the projected system.
- (iii) Identifies the essential communication between the services and the agents responsible for issuing and receiving it, essential for preparing test routines. In other words, it allows to attribute/responsabilize each identified requirement to an agent. It is important to mention that

the lowest level of refinement of the KAOS method is obtained when identifying the actions that each agent is in charge of.

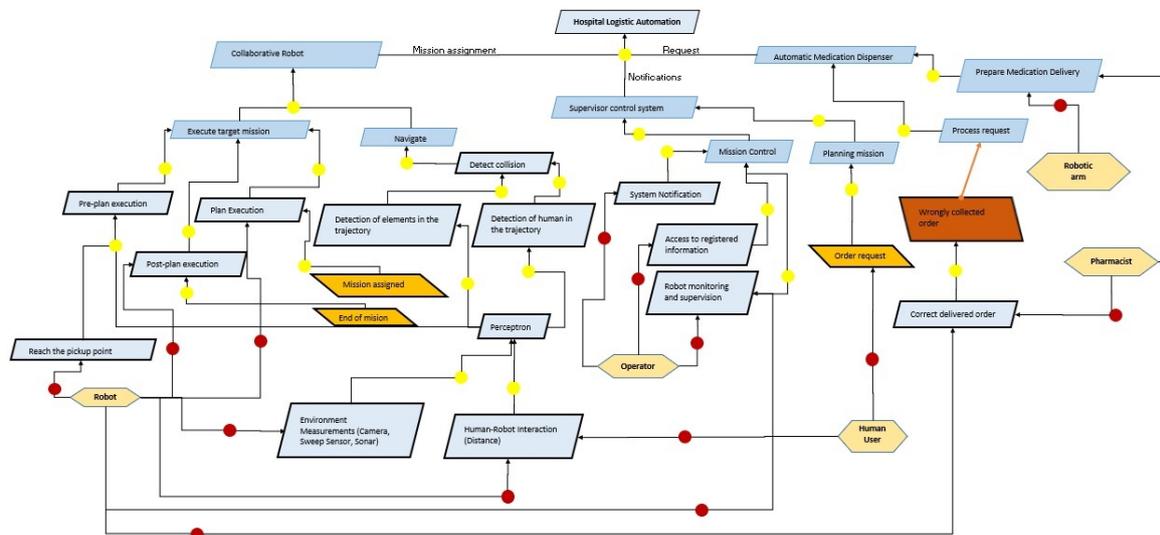


Figure 8. Goal-oriented model for the revised specifications.

As depicted in the model of Figure 8, one of the objectives is to model logistics automation in the hospital, composing a Collaborative Robot and the Medication Dispenser belonging to the Hospital Pharmacy. Therefore, it is a service composed of other services. Machine-to-machine value co-creation is already included (which was not explicit in the original project, although it was in the background specification). The goal-oriented service approach includes the collaboration among services over the former conceptualization of the system. This modeling appeared as a restriction, more than a requirement specification, opening space for a revision.

Consequently, details of service relations (and their operation) were not explicitly analyzed. The system was conceived interactively but was not modeled and analyzed formally. Therefore, it is possible that considering the former project as a collaborative service of services leads to some collaboration flaws. Some details of the execution route of specific goals will appear explicitly in the operations diagram, where the user journey concerning the collaboration between humans and robots will be clarified.

Figure 9 corresponds to the Petri net generated from the KML2PNML transfer. The network allows to analyze or workflow, with the particularity of observing if there are conflicts or possible interference between identified requirements. It is also possible to identify which requirements depend on an external action, known as expectations (16, 284297, 284039). For a proper understanding of the Petri net, it should be mentioned that the locations (represented by circles) represent requirements, objectives, sub-objectives, and expectations. On the other hand, a place cannot be connected to another place directly, it is necessary to implement a transition (represented by rectangles). The arcs represent the direction in which the flow in the network takes place.

Once a reliable net has been synthesized, the verification of requirements can be carried out to obtain a complete and consistent system, making it possible to analyze the structural and dynamic properties in order to determine inconsistencies in the modeled requirements and other characteristics of interest in the design process. Every system modeled in Petri Nets admits one and only algebraic representation, which is given by matrices that represent the structure of the model (distributed states and transitions) and by an equation of state that represents the evolution of states (Equation 1) by the occurrence of enabled transitions.

$$M_i = M_0 + A^T \sigma_{i-1} \quad (1)$$

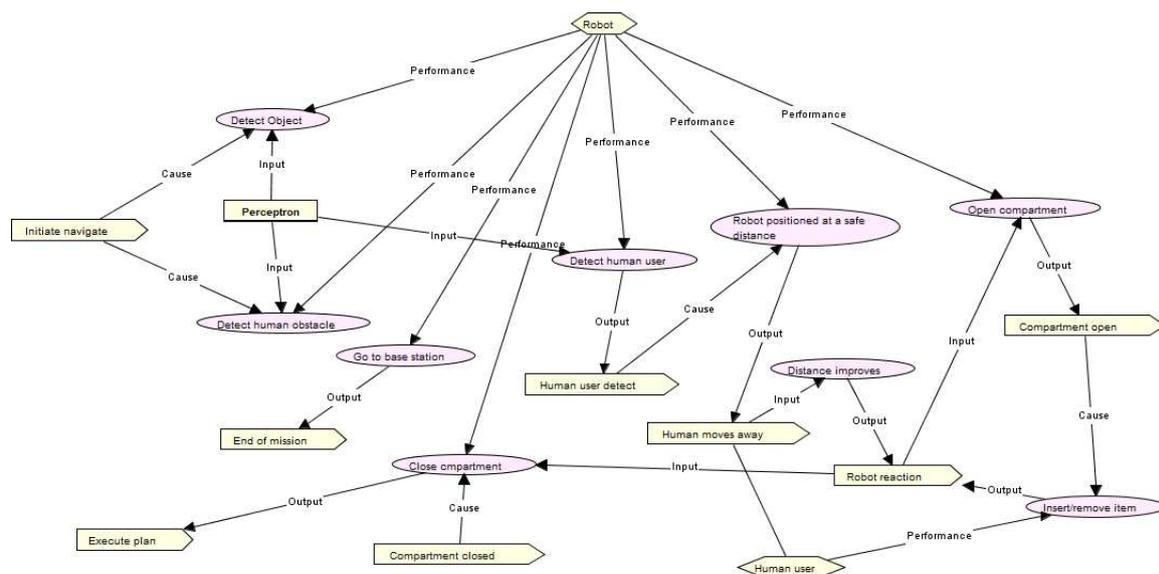


Figure 10. Operation diagram.

The human agent interacts with the robot during the pickup and delivery of items, as seen in Figure 8. It is important to stress that goal-oriented service modeling must go strictly to the collaboration journey, avoiding prescriptive statements which emerge from functional descriptions. Therefore, the formal representation of this journey using Petri Nets (or any other tool that models workflow), can reflect potential problems and flaws in the collaboration process.

Figure 11 show the Petri net model synthesized using the algorithm transferring from KML to PNNML, which could be visualized in any Petri net environment⁴ unnecessary requirements.

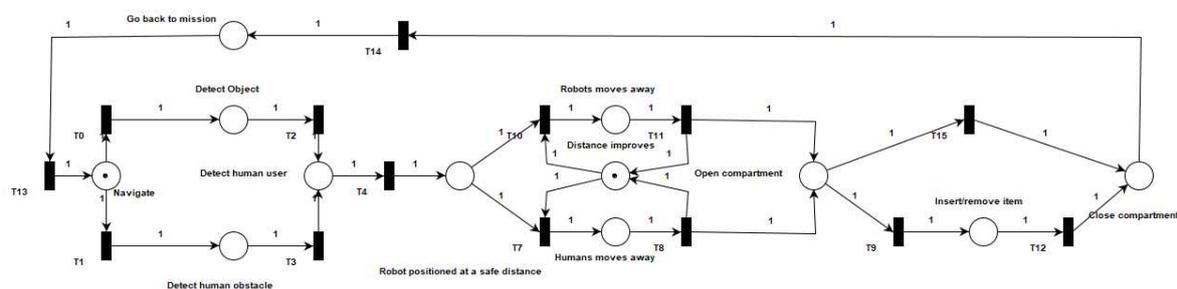


Figure 11. Petri Net of operation diagram.

Interpreting the Petri Net flow, the robot navigates until it detects a human (and the perceptron can distinguish a human from an inanimate object⁵). Assuming the detected human is an object agent (not an obstacle), it is a collaboration target. Thus, the robot stays at a safe distance, where it cannot hit the human with the compartment opening and waits. The human (or the robot) can improve the distance between them (for security reasons), and the robot opens the compartment. The human should insert or retrieve an item from the compartment and improve the distance again. The robot then closes the compartment and goes back to the mission.

However, it can be noticed that the human (for any reason) can improve the distance without inserting or retrieving an item, in which case the robot returns to its mission empty. Although possible, this is not desirable.

⁴ In this work, the tool Pipe3, v.4.3, available at <https://www.softpedia.com/get/Others/Home-Education/PIPE2.shtml>, was used.

⁵ It is not expected to find any pet in a hospital, but the transaction will work with them the same way.

The relation is almost linear, and a short invariant analysis will show that the summation of all marks is $\sum m_i = 2$, denoting a conservative net. The only problem is the conflict generated by firing transitions T9 and T15, leading to an empty mission.

This problem can be solved by embedding a movement detection system in the robot, analyzing if an insertion/retrieval movement was performed, and eventually asking for confirmation from the user by pressing a button or communicating by voice message. The next version of the robot would improve the hardware to include that.

5. Conclusions

This paper focuses on applying the goal-oriented service approach to collaborative robotics. Specifically, a stepwise cycle of requirements development called Model-based Requirements Engineering (MBRE) was proposed [29] to support a systemic method for synthesizing a concise and close requirements model. An evolution of this proposal to include the modeling of value co-creation and the relationship between services, or between services and different categories of users, motivated the joint effort to adapt and apply the proposal to collaborative robotics. Reasons for that were punctuated in the article. However, shortly, the benefit is to have a service architecture and all its communication, including collaboration, in the same model that could be formally verified.

Petri Nets was introduced as the formal presentation to formal analysis and verification [30] because it is typically used in the design of automated systems. It should be noticed that Carl Adam Petri proposed this formalism in 1962 precisely to represent “communication between automata” [26].

Using XML to transfer models from KAOS to Petri Nets is nothing new and follows the tendencies of ISO/IEC standards. The novelty is to apply that to semi-formal diagrams (like KAOS), explore its formation rules, and adapt the result to documentation and reusability support. Another significant contribution is assembling the whole requirements cycle in the MBRE, adding system modeling to the service network concept. In MBRE, each refinement cycle comes up with a model with a different degree of abstraction.

Adapting this framework to collaborative robotics poses the new challenge of combining control and perceptrons with mission workflow and Artificial Intelligence (planning), in addition to the characterization of collaboration. In the case study, start from the proposal of a robot initially focused on control and automation, without leaving aside the analysis of the model to include the mission workflow, in order to analyze and detail the collaborative operation.

The distance between the robot and the human user characterizes collaboration. However, error recovery mainly demands improving perceptron to avoid empty missions. Missing this improvement would result in unwanted situations of undelivered (or unloaded) items. The interaction should be enhanced by adapting the robot control with perceptrons that can detect and analyze (human) movements, suggested for further developments.

Independently of the sophistication level of the robotic system, investing in a preliminary design process can bring new ideas and open possibilities (and functionalities) to be included in the design process. Revising requirements elaborated by other approaches - as developed above - is also an option but opens the possibility to emerging requirements that conflict with the implementation adopted. Intense conflicts did not appear in the case study because the original design was transparent, flexible, and based on a lean solution.

For future research, it is proposed to work with the formalization of the method in a framework (so far, it is a collection of tools) implemented in the cloud. As for the application in collaborative robotics, the challenge is to apply the same approach to systems with multiple collaborative problems, starting with adding motion detection, voice communication or cognitive anticipation of user intentions.

Another interesting possibility is to explore knowledge engineering for planning to extend the possibilities of the automatic planning system to cope with multiple missions or changes in the current plan [27].

Acknowledgments: The authors would like to thank the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) for the research financing. Also, for the infrastructure collaboration in the area of Mechanical Control and Automation Engineering of the Polytechnic School at the University of São Paulo. The authors acknowledge the University Hospital Innovation Center for the support in the project used as a case study.

References

1. Albuquerque D and Castro J and Souza A (2018) *A Requirement Definition Framework for Robotic Systems Domain: An exploratory study*, In: Proc. of the Workshop in Requirements Engineering. Rio de Janeiro.
2. Cabello R and Escalona MJ and Enríquez JG (2020) *Beyond the Hype: RPA Horizon for Robot-Human Interaction*, Lecture Notes in Business Information Processing, 393 LNBIP:185–99. [DOI: 10.1007/978-3-030-58779-6 13]
3. Cabello Ruiz R and Jiménez Ramírez A and Escalona Cuaresma MJ and González Enríquez J (2022) *Hybridizing humans and robots: An RPA horizon envisaged from the trenches*, Computers in Industry, 138:103615. Available from: <https://doi.org/10.1016/j.compind.2022.103615>. DOI:10.1016/j.compind.2022.103615]
4. Čaić M and Odekerken-Schröder G and Mahr D (2018) *Service robots: value co-creation and co-destruction in elderly care networks*, Journal of Service Management, 29:178–205. [DOI:10.1108/JOSM-07-2017-0179]
5. Cherubini A and Navarro-Alarcon D (2021) *Sensor-Based Control for Collaborative Robots: Fundamentals, Challenges, and Opportunities*, Frontiers in Neurorobotics, 14:1–14. [DOI: 10.3389/fnbot.2020.576846]
6. Cloutier RJ and Hutchison N (2022) *SeBok: Guide to the Systems Engineering Body of Knowledge*, 2nd ed. INCOSE.
7. Francesco P and Paolo GG (2017) *AURA: An Example of Collaborative Robot for Automotive and General Industry Applications*. *Procedia Manufacturing*, 11, 338–45. Available from: <http://dx.doi.org/10.1016/j.promfg.2017.07.116>. [DOI: 10.1016/j.promfg.2017.07.116]
8. Gualtieri L and Rauch E and Vidoni R (2021) *Emerging research fields in safety and ergonomics in industrial collaborative robotics: A systematic literature review*, Robotics and Computer-Integrated Manufacturing, 67:101998. Available from: <https://doi.org/10.1016/j.rcim.2020.101998>. [DOI: 10.1016/j.rcim.2020.101998]
9. Horkoff J and Aydemir FB and Cardoso E and Li T and Maté A and et al (2019) *Goal-oriented requirements engineering: an extended systematic mapping study*, Requirements Engineering, 24:133–60. [DOI: 10.1007/s00766-017-0280-z]
10. Kindler E (2006) *The Petri Net Markup Language and ISO/IEC 15909-2: Concepts, Status, and Future Directions*, In: Schnieder E, editor. Design of Complex Automation Systems. May. Braunschweig: ifak - Institut für Automation und Kommunikation, pp. 35–55.
11. Lasota PA and Fong T and Shah JA (2017) *A Survey of Methods for Safe Human-Robot Interaction*, Foundations and Trends in Robotics, 261–349. [DOI: 10.1561/23000000052]
12. van Lamsweerde A (2001) *Goal-oriented requirements engineering: a guided tour* In: Proc. of the 5th. IEEE Int. Symposium in Requirements Engineering. Toronto, Canada, pp. 249–63.
13. Liu Z and Ming X and Song W and Qiu S and Qu Y (2018) *A perspective on value co-creation-oriented framework for smart product-service system*. *Procedia CIRP*, 73:155–60. Available from: <https://doi.org/10.1016/j.procir.2018.04.021>. [DOI: 10.1016/j.procir.2018.04.021]
14. Lv Z and Qiao L (2020) *Deep belief network and linear perceptron based cognitive computing for collaborative robots*, Applied Soft Computing Journal, 106300. Available from: <https://doi.org/10.1016/j.asoc.2020.106300>. [DOI:10.1016/j.asoc.2020.106300]
15. Maurice P and Padois V and Measson Y and Bidaud P (2017) *Human-oriented design of collaborative robots*, International Journal of Industrial Ergonomics, 57:88–102. Available from: <http://dx.doi.org/10.1016/j.ergon.2016.11.011>. [DOI:10.1016/j.ergon.2016.11.011]
16. Maurice P and Schlehner P and Padois V and Measson Y and Bidaud P (2015) *Automatic selection of ergonomie indicators for the design of collaborative robots: A virtual-human in the loop approach*, IEEE-RAS International Conference on Humanoid Robots, [DOI:10.1109/HUMANOIDS.2014.7041455]
17. Nof SY and Silva JR (2018) *Perspectives on Manufacturing Automation Under the Digital and Cyber Convergence*, Polytechnica, 1, 36–47. [DOI:10.1007/s41050-018-0006-0]
18. Orellana MA and Silva JR and Pellini EL (2021) *A model-based and goal-oriented approach for the conceptual design of smart grid services*, Machines, 9:1–23. [DOI: 10.3390/machines9120370]

19. Proia S and Carli R and Cavone G and Dotoli M (2021) *A Literature Review on Control Techniques for Collaborative Robotics in Industrial Applications*, IEEE International Conference on Automation Science and Engineering 2021; 591–96. [DOI: 10.1109/CASE49439.2021.9551600]
20. Proper HA and Bjeković M and Feltus C and Razo-Zapata I (2018) *On the development of a modelling framework for value co-creation*, CEUR Workshop Proceedings, 2239:122–32.
21. Qiu R (2014) *Service Science*, Hoboken, NJ: John Wiley & Sons.
22. Roy RB and Lillrank P and reekanth VK and Torkki P (2019) *Designing Service Machines*, Singapore: Springer, [DOI: 10.1007/978-981-13-0917-5 3]
23. Sadiq M and Hassan T and Nazneen S (2017) *AHP-GORE-PSR: Applying analytic hierarchy process in goal oriented requirements elicitation method for the prioritization of software requirements*, 3rd IEEE International Conference, 1–5. [DOI: 10.1109/CIACT.2017.7977366]
24. Saenz J and, Elkmann N and, Gibaru O and, Neto P (2018) *Survey of methods for design of collaborative robotics applications- Why safety is a barrier to more widespread robotics uptake*, ACM International Conference Proceeding Series, Part F1376:95–101. [DOI:10.1145/3191477.3191507]
25. Shaumburg H and Korablev V and Laszlo U (2020) *Technological Transformation: A New Role For Human, Machines And Management*, vol. 157. Cham: Springer International Publishing [DOI: 10.1007/978-3-030-64430-7]
26. Silva M (2012) *50 years after the PhD thesis of Carl Adam Petri: A perspective*, IFAC Proceedings Volumes (IFAC-PapersOnline), 45:13–20. [DOI: 10.3182/20121003-3-MX-4033.00006]
27. Silva JR and Silva JM and Vaquero TS (2020) *Formal Knowledge Engineering for Planning: Pre and Post-Design Analysis*, In: Vallati M, Kitchin DE, editors. Knowledge Engineering Tools and Techniques for AI Planning. Springer, pp. 47–66.
28. Silva JR and Vital E (2020) *Towards a Formal Design to Service-Oriented Cloud Manufacturing*, In: Bernardon DP, Vargas FL, editors. Proc. of CBA. Santa Maria: Blucher; 2020. pp. 8–16.
29. Silva JR and Macedo ECT and Correa YG and Medeiros RP (2021) *A Multilayer Proposal to a Smart Home Applied to Healthcare*, Polytechnica, 4:1–14. Available from: <https://doi.org/10.1007/s41050-021-00029-7>. [DOI: 10.1007/s41050-021-00029-7]
30. Silva JM and DelFoyo PMG and Olivera A and Silva JR (2022) *Revisiting requirement engineering for intelligent manufacturing*, Int Journal of Interactive Design and Manufacturing, 16:1–14. [DOI: <https://doi.org/10.1007/s12008-022-00968-0>]
31. Souza RS and Sanfilippo F and Silva JR and Cordero AF (2016) *Modular exoskeleton design: Requirement engineering with KAOS*, In: Proceedings of the IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics. vol. 2016-July; 2016. pp. 978–83. [DOI:10.1109/BIOROB.2016.7523756]
32. Vicentini F. (2021) *Collaborative Robotics: A Survey*. *Journal of Mechanical Design, Transactions of the ASME*, 143, 1–20. [DOI: 10.1115/1.4046238]
33. Wang L and Gao R and Váncza J and Krüger J and Wang XV and et al (2019) *Symbiotic human-robot collaborative assembly*, CIRP Annals, 68:701–26. Available from: <https://doi.org/10.1016/j.cirp.2019.05.002>. [DOI:10.1016/j.cirp.2019.05.002]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.