Survey of methods for design of collaborative robotics applications- Why safety is a barrier to more widespread robotics uptake

José Saenz, Norbert Elkmann
Business Unit Robotic Systems
Fraunhofer IFF
Magdeburg, Germany
jose.saenz@iff.fraunhofer.de
norbert.elkmann@iff.fraunhofer.de

Prof. Olivier Gibaru
Arts et Métiers Paristech
Lille, France
olivier.gibaru@ensam.eu

Pedro Neto
Department of Mechanical Engineering
University of Coimbra
Coimbra, Portugal
pedro.neto@dem.uc.pt

ABSTRACT
While collaborative robots have made headlines through recent industrial applications, they are not as widespread in industry as it may seem. The authors of this paper believe that one reason for this slow uptake is due to the high requirements on the safety and the lack of engineering tools for analyzing collaborative robotics applications. Systems engineering provides a good framework for creating the engineering tools needed for faster and more reliable deployment, but has only recently been applied to robotics challenges. In this paper, we discuss the state of the art for designing robotics applications featuring human-robot collaboration (HRC) and then review existing systems engineering approaches, which could offer support. Our review aims to support the robotics community in the future development of engineering tools to better understand, plan, and implement applications featuring collaborative robotics.

CCS Concepts
• Applied computing→ Physical sciences and engineering → Engineering→ Computer-aided design.

Keywords
Survey; collaborative robot application design; model-driven systems engineering; human-robot safety

1. INTRODUCTION
The proceedings are the records of the conference. ACM hopes to give these conference by-products a single, high-quality appearance. To do this, we ask that authors follow some simple guidelines. In essence, we ask you to make your paper look exactly like this document. The easiest way to do this is simply to download a template from [2], and replace the content with your own material.

ICMRE 2018, February 7–11, 2018, Valenciennes, France

ACM ISBN 978-1-4503-6365-5/18/02…$15.00
https://doi.org/10.1145/3191477.3191507

Recently the field of collaborative robots has been making impressive gains. There have been many reports of new applications featuring them in industrial production, working safely side by side next to humans. However, despite these headlines and all the media attention paid to the subject, there are relatively few collaborative robots in industrial applications compared to standard industrial robots applications [1], [2]. Furthermore, their full potential has not yet been fully explored, and many initial applications feature collaborative robotics doing the same work as industrial robots, but simply without the fence. One reason for this is that, despite the recent publication of safety standards for robotics, the strict safety requirements pose a difficult challenge for system integrators and robotics applications designers. The most relevant safety standards for collaborative robotics applications are the ISO 10218-1 and -2, as well as the ISO-TS 15066. The ISO 10218-1 describes the general design requirements for inherent safety of industrial robots for use in collaborative applications. The ISO 10218-2 addresses systems integrators and describes the hazards specific to a complete system, the means for safeguarding against them, and the requirements when introducing a system to the market. ISO-TS 15066 is currently only a technical specification. It defines the four safeguarding modes and defines limits, especially for the cases of speed and separation monitoring and power and force limiting. The approach taken by all three standards is an acceptance of the fact that every application is different and questions regarding the process, the choice of safety components, and their impact on the human and their work need to be answered specifically for each case. Under no circumstances should collaborative robots cause injuries to humans working near or with them. From the business perspective, there is furthermore the requirement that any process to which HRC is introduced should not be impeded by it.

Current methods and tools for designing robotics applications are at best only able to help answer individual questions but don't allow support to study the system as a whole. As a result, it currently takes very long to develop and implement a simple application featuring collaborative robotics, there is a lot of uncertainty during that process, and there are no means for determining whether the design decisions made are the best.

Systems engineering has been successfully applied to a number of industries, most notably defense, automotive (AUTOSAR) [3] and aerospace. These systems help reduce complexity, allow for reuse of engineering artefacts including software, allow for early system verification, and ultimately support more cost-effective engineering efforts while at the same time reducing errors. While there are a few instances of applying systems engineering to robotics, there is no known instance to our knowledge of its application with the aim of supporting the development and commissioning of collaborative robotics in industry.

In this paper, we review work regarding the application of systems engineering methodology to the design of robotics applications, in order to understand whether it can be used as a basis for supporting future work, or if we can learn lessons from their efforts in the development of new tools to support designers, applications engineers, and system integrators. Prior to reviewing existing systems engineering approaches, we will briefly describe the current process of designing and implementing a collaborative robotic application, and formulate specific requirements on the process.

2. DESIGNING COLLABORATIVE ROBOTICS APPLICATIONS

In the following, we will briefly describe the current process of designing and implementing collaborative robotics applications in manufacturing industry.

2.1 Overview Current Design Process

The starting point for the design of the collaborative application is a general idea of how the application should look. Figure 1 shows the overall flow, with feedback loops where specific sets of requirements are checked before the design can move to the next step. The initial design will usually include the robot type and a simple layout (including position of robot, pathways, material flows, etc.). Once this layout seems sufficient to satisfy the most basic requirements, the analysis will be expanded to include the process. In general, the design becomes more refined as further requirements regarding the process, safety, and conformity to standards are considered.

In industry, there are a number of domain specific tools for designing collaborative robotics applications. The standard method consists of a project manager working with mechanical and electrical engineers. The mechanical engineers work in CAD, possibly with a robotics simulation environment or plug-in. They model the overall layout, the material flow, and define the type of collaboration with the human by also defining the human's tasks. They check that the initial requirements are met and choose the robot type based on a few criteria such as payload, reach, and specific customer preferences. The electrical system can be designed in an electrical computer aided design (ECAD) program. There is usually no direct digital connection between the mechanical and electrical domains. Electrical components are physically modeled and the cable paths are considered by the mechanical engineers. Logical pathways, communication, and electrical energy can be modeled in the ECAD. Once the system is physically ready, the programmer can begin the process of programming the robot. They rely on a variety of tools and middlewares, and can sometimes use the simulation built beforehand for their purposes.

The project manager oversees all sides and ensures that the status is compatible and that any updates or changes are communicated so that each individual team can update their individual models.

Usually the safety for a collaborative robotic application is important, and either an extra safety expert is brought into regular meetings or a member of one of the existing teams (electrical or mechanical). There are a number of dedicated tools such as PAScal from Pilz which support the choice of the electrical components and help ensure conformity according to functional safety standard ISO 13849-1.

Looking at this, we can see that the engineering tools are fragmented. Any questions regarding "what-if" scenarios involve teams of people with competing interests, unclear requirements, and unclear means of ensuring that an optimal goal is reached. This is a particular challenge with HRC applications, which feature a large number of actors and components with many interdependencies that cannot be easily managed.

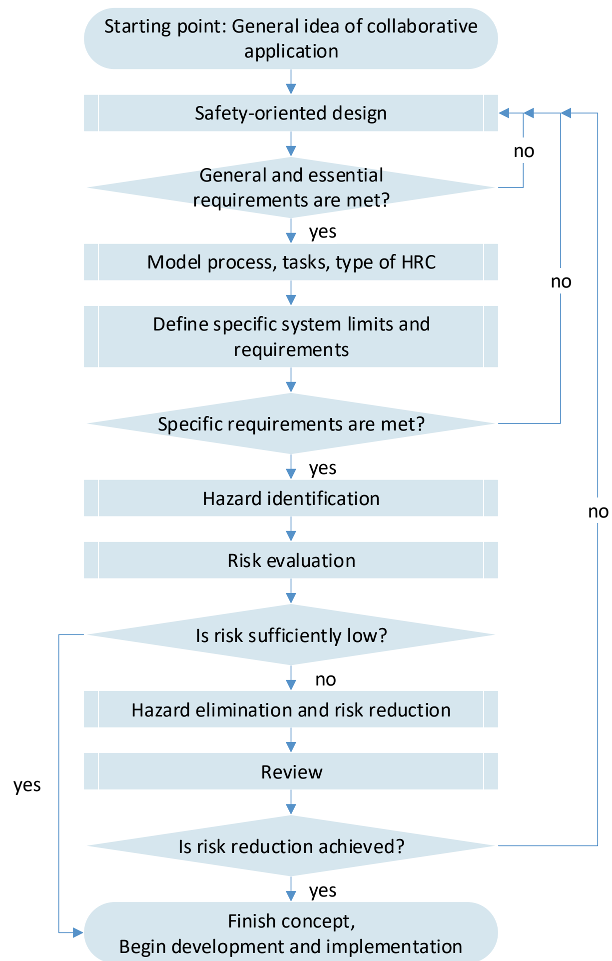

Figure 1. Flow model of different phases during concept (design) of a HRC application in manufacturing

In order to further define the scope of the work, the authors would like to use the generic life cycle model for a project or commercial system and refine it for a project regarding an industrial application featuring HRC. The top row of Figure 2 shows the structure of a generic life cycle model, starting with the concept phase, and moving from there to the development, system production phase to utilization and finally retirement of the system. Underneath the generic structure is a refinement we propose, showing the development phases of hardware and software development, as well as safety validation necessary for the commissioning and CE mark of the system. A further specific challenge in collaborative robotics is the issue that any changes to the robot program require a review of the safety validation and an update to the CE mark. Returning to the example from the previous section, we see that the activities where we identified current challenges are specifically located in the concept and hardware development phases.

Concept	Development			Production	Utilization / Support	Retirement
	HW	*SW*	*Safety Validation*	*Commissioning*	*Re-programming*	
				CE Mark	*Safety Validation*	
					CE Mark Renewal	

Figure 2. Life Cycle Model for industrial HRC application, based on Generic Life Cycle Model from ISO/IEC/IEEE15288:2015

2.2 Example Industrial Applications

In order to better illustrate the current challenges the authors see in the design of collaborative robotic systems and narrow down the scope of the work ahead, we would like to describe a few example industrial applications. Behrens, Saenz, Vogel, and Elkmann [4] present a series of simple questions to characterize the type of collaboration. They define four types of interaction as:

- co-existence (no shared workspace),

- sequential cooperation (humans and robots are in the same space at different times),

- parallel cooperation (humans and robots in the same space at the same time, without contact), and

-collaboration (humans and robots in the same space, at the same time, whereby contact is necessary and/or possible).

We will use their terminology to characterize a range of robotics applications that are in use today. Laboratory applications will not be considered since in laboratory environment the safety aspects are normally not fully considered.

2.2.1 Machine tending

There are many examples of collaborative robots used for machine tending. An example from the project MR_KOOP [5], an industrialrobot is to be used to insert and remove parts into a metal-working machine. An operator can travel through the workspace at any time to access other machines and to take over the robot's tasks in special circumstances. In those situations, the robot should stop. This application can therefore be classified as sequential cooperation.

Figure 3: Sample HRC application from MR_KOOP project

2.2.2 Robots in automotive assembly (final assembly line)

There are a growing number of robots used in the automotive industry in the final assembly line. Documented examples include the certified system PART4you in use by Audi, which uses a KR5 with a bin-picking vision system to pick canisters from a box and hand them to an operator at an ergonomic position [6]. While the robot and the human have a handover, the system is set-up so that the human and robot are not in the same workspace during robot motion. Given that the robot is stationary during the handover, this application is a case of sequential cooperation.

Another example is from BMW, where a Universal Robot pushes the snap fittings of a door panel into place on a car door sub-assembly [7]. This scenario can be characterized as sequential cooperation, as the robot and the operator do not work at the same time on the door panel.

Figure 4. Collaborative door assembly application for Adam Opel AG

A final example is the door assembly system made for Adam Opel AG [8]. Here a high payload robot picks up the completed door from a separate line and positions it next to the car while on the moving line. An operator then can come to the robot, has the option to hand-guide it within certain limits to fine-position the door relative to the car. Then while the robot holds the door in place, the operator can fasten the door to the chassis and connect the cables between them. The operator then signals to the robot that the door is correctly fastened and the robot releases the door and starts the process anew. This is a much more complex task, featuring sequential and parallel cooperation, as well as collaboration (when the robot is hand-guided).

2.2.3 Palletizing robots

In addition to non-collaborative robots, there are known cases of robots used for palletizing that are able to work in close collaboration with humans [9]. The system removes boxes from two parallel conveyors and stacks them on two pallets, located at the end of the conveyors. Operators can remove a full pallet and place an empty pallet in its place on one side, while the robot is still stacking on the other side. This application can be characterized as sequential collaboration during the nominal situation whereby the robot stacks a pallet, and as collaboration when the operator enters the workspace to change a pallet. This system is safeguarded through power and force limiting.

2.2.4 Summary existing industrial applications

While each system may seem relatively simple from an outsider's perspective, the time from the first concept until the certified system in operation can take upwards of 1-2 years' time. This lead-time is longer than standard automation projects, and uncertainty during the concept and development phases with regard to conformity to the new HRC standards is a major contributor. The majority of the examples explained here and in use feature sequential or parallel cooperation, and as a first step towards developing new engineering tools and methods to support the concept and development process for industrial applications featuring HRC, we propose focusing on these types of applications.

2.3 Requirements on Improved Design Process

Using the current method as a starting point, we would like to formulate a few requirements for an engineering tool and/or approach to streamline the design process.

- Requirements tracking (can the process of checking whether a design fulfills the specified and explicitly formulated requirements be automated or supported)?

- Support for consideration of safety-related questions (according to ISO 10218-1, -2 and ISO-TS 15066) in addition to other related standards and directives

- Support for what-if analyses covering all aspects of an HRC application to improve/ optimize designs (e.g. How is my application affected when I speed up / slow down the robot? How does the required minimum protective distance change depending on the sensor choice, position in the workspace, etc.?)

- Support for data round-tripping so that system models can be used together with existing engineering tools such as CAD and simulation software.

- Allow for verifiable and certifiable results. As an example, if the engineering tool results in a specific configuration for a safety sensor, then an output of the engineering tool should be the sensor configuration for the specific sensors used in the application.

- Allow for definition of specific verification steps to ensure that the real implementation corresponds to the model with a high level of integrity.

The engineering tool and its connections to CAD during concept and development phases and to the real implementation during production, utilization and support phases is schematically shown in Figure 5.

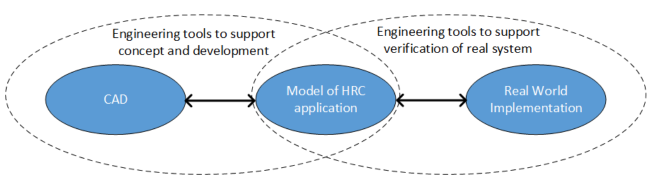

Figure 1. Schematic drawing showing connection between proposed engineering models, CAD/CAE for design, and real world implementations

This approach is very specific for the task of concept and design of HRC systems, as well as the later work during development and implementation. The concerns of the mechanical, and electrical engineering teams are that the system is built in a way that all the requirements are fulfilled, and also that the real implementation corresponds to the modeled system. The second step, if done correctly, could potentially save lots of time and eliminate a large source of error during the commissioning phase.

3. SYSTEMS ENGINEERING APPROACHES FOR ROBOTICS

Systems engineering (SE) [10] is a broadly defined field which seeks to understand man-made systems by concentrating on the whole system as distinct from the parts. It starts by defining requirements at an early stage and bringing together different engineering disciplines to design and validate a solution while maintaining consideration for the complete problem. This approach has been successfully used for a number of other industries featuring a high level of complexity, including defense, aerospace, and automotive industries. SE is particularly well suited for a complex system such as a robotics application featuring HRC. In this case, the parts of the system include the robot, its tools, the safety sensors and safety control system, the human operators, the environment, and other production systems (including other applications featuring HRC). The application of SE can help reduce complexity, will allow for better reuse of engineering artefacts including software, and most importantly from our point of view, using SE for HRC applications will lead to more cost-effective engineering efforts with a reduction in errors and uncertainty about the final system. While there are a few instances of applying systems engineering to robotics, there is no known instance of its application with the aim of supporting the development and commissioning of collaborative robotics in industry in the life cycle phases as indicated above.

In the following, we will review known systems engineering approaches in the field of robotics and describe their application to the formulated problem

3.1 Model-Based Software Engineering

Model-based software engineering uses an SE approach to address the programming of a robotic system. Issues addressed by this work include the fact that writing software code is a time-consuming and expensive process, that this code is nevertheless often not reusable and can become obsolete with changes to the robotic systems' hardware configuration. The BRICS project [11] supported the idea of model based engineering and aimed to support components reuse in robotics through dissemination of best practices. The models used were however not sufficiently specified to allow for significant re-use.

Recent work in this field [12] [13] focuses therefore on the meta-models to allow for more generic modeling of the robotic system with the specific aim of automatically generating software code, both offline (prior to robotic action being initiated) and during run-time. Here an entire toolchain for modeling was developed in a UML environment. In these cases, the aim is for a programmer to specify high level tasks in the task model such as "grasp cup" versus programming a specific set of actions. Furthermore, generic interfaces were created so that the models could be used with a number of different specific tools for software creation. In addition to generating software for a robot based upon the models, other quality of service aspects such as run-time ability were able to be evaluated. This work focuses on the software engineer's point of view, and less on the overall design and consideration of safety aspects from the mechanical point of view.

When viewed according to the life cycle model from Figure 2, we can see that the focus of these efforts starts in the development process and is relevant throughout the production, utilization, and support phases. This is in contrast to the stated aim of the authors to support designers and system integrators during the concept and development phases.

More recently, the H2020 project RobMoSys [14] seeks to build upon this work by offering cascaded funding to develop a larger set of models and engineering tools that will serve the entire robotics community. The project places a focus on applications featuring mobile manipulation, and it remains to be seen whether an approach as described in this paper will be considered by the project.

3.2 Model-Based Risk Analysis

There has also been recent research focused on the use of model based engineering methods to support risk analysis for robotics. Earlier work [15] initially used UML as the modeling language to analyze the safety of a medical robot. In particular, the work focused on an analysis of the task and of possible human errors, to understand and handle human errors while working with the system. Further, the authors sought to support the risk analysis by identifying hazards and applying the Failure Modes, Effects and Criticality Analysis (FMECA) technique. Further work [16] proposes an approach to apply the HAZard Operability (HAZOP) technique with UML models. The authors chose the HAZOP method since it can be used earlier in the engineering process than FMECA, and that it can include human activity as a source of hazard. Also of interest is the author's focus on the use of use-cases, sequence and state diagrams to analyze the system. The system is not able to identify all hazards, but focuses on operational hazards linked to human-robot interactions. The method has been applied to a number of robots from research projects, including lightweight industrial robots, mobile manipulators, and an assistive robot for healthcare applications. Further work [17] describes a domain specific modeling language for robotic systems called RobotML, which is used to generate a fault tree analysis and supports formal verification methods.

While the topics of safety in the aforementioned works are covered, they are not addressed in a way that a mechanical engineer approaches the issue during the concept and development of an application featuring HRC. The size of safety zones and the overall effect of safety requirements on the environment, on the type of interaction, and on the overall process are not considered. Furthermore, this approach does not sufficiently address the concept of requirements engineering to ensure conformity with the collaborative robotics standards 10218-1 and -2, as well as the ISO-TS 15066.

Gribov and Voos [18] use SysML models and requirements engineering to check whether the standard ISO/DIS 13482 is fulfilled in the early design stages. However, this work assumes that the robotics hardware is not changing and only focuses on software issues, including how to represent software issues in a formalized way and how to define an engineering process which provided evidence of safety and allows for software reuse.

We therefore see that future work to support the design of industrial applications featuring collaborative robots can build upon this previous work. In particular, the application of systems engineering methodology and the use of requirements engineering to ensure conformity to a specific standard early in the design process will be a good starting point.

3.3 Ontologies for Robotics

In researching model based system engineering approaches, the concept of ontologies and their use in robotics also becomes of interest. Ontologies formally describe a system so that it is machine understandable. With the advent of the internet, there were a lot of efforts put into developing both general and domain specific ontologies to make content on the internet machine-readable and in an effort to organize information. When creating models of robotic systems which are reusable, existing ontologies could be a useful source of inspiration or starting point. Nevertheless, there is the caveat that ontologies are imperfect and are made to represent a specific domain or area of interest.

Notable efforts regarding robotics and sensor ontologies include the Roboearth project [19], whose aim was to use existing information from the World Wide Web (WWW) to support robotic task accomplishment and to create a world wide web for robots so that they can exchange information between one another and which focused on the semantic representation for actions, objects and environments. Further work in this direction [20] focused on using these ontologies to support AI-assisted task planning. In this case, the aim was to let a robot reason about its movements and generate executable motions that are adaptable for different robots, objects and tools. Another notable effort at developing an ontology for the robotics community is the IEEE Core Ontology for Robotics and Automation (CORA) [21]. At this point it has only specified a relatively high-level ontology, which nevertheless allows for the use with other existing ontologies. Working groups are currently working on refined the ontology by describing tasks from a variety of robotics domains, including industrial robotics, in order to make the CORA more usable within the community. Sensors are also an integral part of a robotic system featuring HRC, and there have been considerable efforts at creating sensor ontologies. Eastman, Schlenoff, Balakirsky, and Hong. [22] present a good overview of currently available sensor ontologies.

The work we propose does not focus on ontologies. Nevertheless, in order to create reusable models and as an effort to avoid reinventing the wheel during the modeling efforts, ontologies could prove to be a useful reference, and our approach will be to review these existing ontologies, in particular with regard to robotics, sensors, and environment, when creating our models.

3.4 Industry 4.0 Framework

Another interesting modeling effort that has recently begun is the Industry 4.0 reference architecture [23]. It seeks to create a unified model of industrial components that are parts of "Industry 4.0" to allow for better design, interoperability, and usage. As a means to describe the parts completely, it combines a number of established standards including the IEC 62890 for defining the life cycle and value stream and the IEC 62264 and IEC 61512 for defining hierarchy levels. It is more engineering oriented than an ontology, and the aim is industrial standardization. Just like with ontologies, it would make sense to consider the framework when creating the models we propose. Any model which fully conforms with Reference Architectural Model Industrie RAMI 4.0 will be very rich with semantic information and will contain much more information than needed for modeling HRC applications. However in view of the fact that HRC applications are at the core of Industry 4.0, and with a forward view towards compatibility and reusability, it would at least be interesting to see how RAMI 4.0 can be taken into account in the modeling effort. A goal would be to have models that are RAMI 4.0 compatible, which only contain the information that necessary from the designer point of view.

4. DISCUSSION

Based on the requirements derived in Section IIc, and on the state of the art, we see that there is a gap between existing systems engineering approaches for robotics safety and current engineering best practice. Any new engineering tools for supporting engineers during the conception and development of industrial HRC applications need to interface with existing CAD/CAE tools to ensure a smooth workflow and should feature requirements engineering based on the three most relevant standards for collaborative robots, namely the ISO-10218-1 and -2, as well as the ISO-TS 15066. Finally, from the viewpoint of a system integrator, any engineering tools will also support the real world implementation, e.g. by verifying that the real set-up corresponds to the modeled system. This will ensure that the requirements fulfilled in the modeled system are also satisfied in the real world and help reduce transcription errors, and will include further support, e.g. for simplifying the parametrization process of safety sensors.

5. CONCLUSION

The design and development of industrial applications featuring HRC currently present engineering teams with a particular challenge. HRC applications are highly complex systems with a large number of individual components that have a large range of interdependencies. Existing engineering tools are fragmented and it is difficult to understand the trade-offs made in the concept and development phases of a new system.

In this paper, we have first described the current methodology for designing an industrial application featuring HRC and presented a few examples of how these applications currently look. We then made the case for using a model based systems engineering approach to better support the design process and defined initial requirements for new engineering tools.

We then presented an overview of the state of the art of methods using Model-Based Systems Engineering (MBSE) in robotics, and described how these relate to our proposed method. Finally, with an eye towards reusability and compatibility with other modeling efforts, we also identified specific ontologies and the RAMI 4.0 as other supporting efforts which are related to the systems engineering approach and which should be considered for future developments in this field.

The authors have begun modeling HRC systems with the aim of developing the engineering tools as specified here. Future publications will describe these efforts towards application of MBSE methodology to the task of designing industrial applications featuring HRC.

6. ACKNOWLEDGMENTS

This work has been partially supported by the ColRobot project, which has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 688807.

7. REFERENCES

[1] RobotEnomics (2016): The facts about Co-Bot Robot sales. Online under https://robotenomics.com/2016/01/11/the-facts-about-co-bot-robot-sales/, cited on 28.08.2017.

[2] International Federation of Robotics (2016): Executive Summary World Robotics 2016 Industrial Robots. Online under https://ifr.org/img/uploads/Executive_Summary _WR_ Industrial_Robots_20161.pdf, cited on 28.08.2017.

[3] Dersten, S., Axelsson, J., Froberg, J., 2011. Effect analysis of the introduction of AUTOSAR: a systematic literature review. In: 2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA). Oulu, Finland, S. 239–246.

[4] R. Behrens, J. Saenz, C. Vogel, N. Elkmann.: "Upcoming technologies and fundamentals for safeguarding all forms of human-robot collaboration", 8th International Conference Safety of Industrial Automated Systems (SIAS 2015), Königswinter, Germany 18-20 November, 2015. ISBN 987-3-86423-163-6, S.18-23.

[5] https://www.extremnews.com/berichte/wirtschaft/9fb01669a439db8 Cited on 04.09.2017.

[6] http://www.computer-automation.de/feldebene/robotik/artikel/119127/ Cited on 04.09.2017.

[7] https://www.press.bmwgroup.com/deutschland/article/detail/T0209722DE/neuartige-mensch-roboter-zusammenarbeit-in-der-bmw-group-produktion?language=de Cited on 04.09.2017.

[8] https://opelpost.com/03/2016/kollege-roboter/ Cited on 04.09.2017.

[9] Wortmann, U., 2017. Sichere MRK bei Procter & Gamble – Entwicklung und Realisierung des CoPal. Presentation, Magdeburg, Germany.

[10] Walden, D., Roedler, G., Forsberg, K., Hamelin, D., Shortell, T. Systems engineering handbook. A guide for system life cycle processes and activities. 4th edition.

[11] Bischoff, R., Guhl, T., Prassler, E., Novak, W., Kraetschmar, G., Bruyninckx, H., Siciliano, B., Pegman, G., Hägele, M., Prassler, E., Zimmermann, T., Agirre, J., Leroux, C., Tranchero, B., Labruto, R., Knoll, A., Lafrenz, R., 2010. BRICS - Best practice in robotics. International Symposium on Robotics. Munchen, Germany, 7-9 June.

[12] Schlegel, C., Hassler, T., Lotz, A., Steck, A., 2009. Robotic software systems: From code-driven to model-driven designs. In: Advanced Robotics, 2009. ICAR 2009. International Conference on, S. 1–8.

[13] Stampfer, D., Lotz, A., Lutz, M., Schlegel, C., 2016. The SmartMDSD Toolchain: An Integrated MDSD Workflow and Integrated Development Environment (IDE) for Robotics Software 7(1), S. 3–19.

[14] http://robmosys.eu/approach/ Cited on 04.09.2017.

[15] Guiochet, J., Motet, G., Baron, C., Boy, G., 2004. Toward a human-centered UML for risk analysis. In: Chris W. Johnson und Philippe Palanque (Hg.): Human Error, Safety and Systems Development, Bd. 152. Boston: Kluwer Academic Publishers (IFIP International Federation for Information Processing), S. 177–191.

[16] Guiochet, J., 2016. Hazard analysis of human–robot interactions with HAZOP–UML. In: Safety Science 84, S. 225–237. DOI: 10.1016/j.ssci.2015.12.017.

[17] Yakymets, N., Dhouib, S., Jaber, H., Lanusse, A., 2013. Model-driven safety assessment of robotic systems. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, S. 1137–1142.

[18] Gribov, V., Voos, H., 2013. Safety oriented software engineering process for autonomous robots. In: 2013 IEEE

18th Conference on Emerging Technologies Factory Automation (ETFA), S. 1–8.

[19] Tenorth, M., Perzylo, A., Lafrenz, R., Beetz, M., 2012. The RoboEarth language: representing and exchanging knowledge about actions, objects, and environments. In: 2012 IEEE International Conference on Robotics and Automation (ICRA). St Paul, MN, USA, S. 1284–1289.

[20] Tenorth, M., Bartels, G., Beetz, M., 2014. Knowledge-based Specification of Robot Motions. In: Proc. of the European Conference on Artificial Intelligence (ECAI).

[21] 2015: IEEE Standard Ontologies for Robotics and Automation.

[22] Eastman, R., Schlenoff, C., Balakirsky, S., Hong, T., 2013. A Sensor Ontology Literature Review: National Institute of Standards and Technology.

[23] Bitkom e.V. Bunderverband Bitkom e.V. Bunderverband Informationswirtschaft, Telekommunikation und neue Medien e.V. (2015): Plattform Industrie 4.0.