Managing Safety and Adaptability in Mobile Multi-Robot Systems

Darko Bozhinoski
Gran Sasso Science Institute
viale Francesco Crispi, 7
67100 L'Aquila (AQ), Italy
darko.bozhinoski@gssi.infn.it

ABSTRACT

In order to reduce human involvement in repetitive and dangerous tasks, innovative approaches are increasingly sought and explored. One of these approaches is represented by the Mobile Multi-Robot systems (MMRSs). The introduction of this kind of systems opens a collection of new business and societal opportunities, but also raises many new challenges. Such systems are exposed to various spheres of uncertainty, spanning from software and hardware variability of a single robot to the one associated to mission planning and execution in possibly unforeseeable environments. In this proposal, we aim to identify how to preserve safety while enabling adaptability at run-time in MMRSs. The objective is to provide a modelling framework based on a methodology which explicitly takes into account safety and adaptability properties at run-time. In order to reach our goal, we plan to use Model-driven engineering(MDE) methods and techniques. Motivated from positive results from the application of MDE in other domains (e.g. avionics, automotive and telecommunications) we postulate that many methods used in this methodology are also relevant for MMRSs. With our work, we expect to give a contribution towards the assurance of safety and adaptability properties for MMRSs in a dynamic run-time context.

Categories and Subject Descriptors

D.2.4 [**Software Engineering**]: Software/Program Verification; D.2.9 [**Software Engineering**]: Management; D.2.10 [**Software Engineering**]: Design; D.2.11 [**Software Engineering**]: Software Architectures

General Terms

Verification, Management, Standardization

Keywords

Mobile Multi-Robot systems, software engineering, Model-driven engineering, safety, adaptability

QoSA'15, May 4–8, 2015, Montréal, QC, Canada.

http://dx.doi.org/10.1145/2737182.2737195.

1. INTRODUCTION

In order to reduce the human involvement in scenarios which are characterized by repetitive and dangerous tasks (ex. natural catastrophes, delivery services, surveillance, environmental monitoring), innovative approaches represented by mobile robotics are seen as particularly suitable for aiding in the process of replacement of the human beings with robotic systems.

A **mobile robot** consists of a SW/HW platform carried around by locomotive elements and able to perform tasks in different contexts. The kind of locomotion that the robot is able to perform is primarily decided upon the environment (aquatic, aerial or terrestrial) in which the robot will be operating [10]. Mobility gives robots enhanced operative capabilities, but at the same time increase complexity and it brings additional challenges that need to be addressed. Some of these challenges are common to all mobile robots (e.g. the navigation problem), while others (e.g. walking gait) are peculiar for a specific locomotion type. The set of mobile robots operating as a team (even together with humans) in a shared environment are defined as Mobile Multi-Robot Systems (MMRSs).

Over the last decades, research in robotics has made huge progress in the fields of image recognition and processing, planning, control, and collaboration. However, we currently have at our disposal a myriad of isolated solutions that are hard to reuse and combine. Software engineering is called to play a key role in securing this new technology's affirmation by making it pervasive and ubiquitous. Powerful methodologies are required to assist the development of robotic software systems, which are expected to be able to: (i) operate in dynamic environments; (ii) collaborate with other systems for solving problems that could not be solved otherwise neither by one single robot nor by a team of robots belonging to the same category; and (iii) automatically deal with unexpected emergent behaviours that might potentially cause severe misshapes.

In the industry, robots have been widely deployed to improve productivity and perform dangerous or tedious and repetitive tasks. These preprogrammed robots have been very successful in industrial applications due to the assurance of appropriate level of safety [18]. In a world that is undergoing significant environmental change, there will be an increasing demand for robots to move from safe controlled environments to operate in dynamic, unpredictable ones. Hence, it is fundamental that safety aspects should be reconsidered and greatly enhanced at this point of time.

To address the increasing complexity of mobile robots, the robotics and automation industry is working towards the establishment of new international safety standards through the International Organization for Standardization (ISO) for robots and robot systems integration [1]. Establishing the guidelines to regulate a safe use of these innovative technologies is, as a matter of fact, the means to increase their trustworthiness and thereby their appreciation, not only in the research and business sectors, but even in the private social sphere. Commercialisation and adoption of mobile robots in dynamic environments will only occur if the safety aspects are considered and incorporated as first class elements in the design of the system. This is the reason why the necessity for formal verification and validation of safety properties for complex robotic software systems, especially MMRSs, has largely intensified. Certification bodies should assure some type of safety certification that relies on a complete understanding of the system. However, for MMRSs that operate in dynamic environments it's quite challenging to consider all variants of the overall system due to their adaptive behaviour [22]. Developers, researchers, scientists and practitioners are facing with a number of challenges in various different domains. From a software engineering point of view, the main challenge is to maximize maintainability, interoperability, and reusability of the robotic system while preserving efficiency, robustness, safety, and reliability [7]. Furthermore, it is very hard to tame the ever-increasing heterogeneity and complexity of multi-robot systems in terms of communication infrastructure, device capabilities and involved stakeholders, while preserving reasonable costs and time to market. In this proposal, we focus into identifying how to preserve safety properties while enabling adaptability at run-time as one of the biggest challenges of Mobile Multi-Robot Systems (MMRSs) within the software engineering research area.

2. BACKGROUND

MMRSs have been proposed in the last twenty years in various settings and for different goals, both in research and in practice [5]. They are composed of set of subsystems which behave as a team, thus accomplishing a global mission. These subsystems might vary both in type and in number. During the MMRSs' life-time new robots might come into the picture or current ones might be reconfigured (or self-configured) at run-time as countermeasure to possible faults, malicious attacks or simply due to additional knowledge acquired during the mission execution.

In order to better understand the complexity of MMRSs we can analyse these systems through the aspect of multidimensional variability since changes can occur both vertically (within domain and specification of a single robot) as well as horizontally (among the MMRSs' parts). In this proposal the focus is on the horizontal variability and the challenges that emerge out of it. In this context, we identify two dimensions of horizontal variability: *software and hardware variability*, and *mission planning and execution variability*, as discussed in the following.

Software and hardware variability. Currently robotic software researchers and engineers are mainly focusing on delivering highly efficient implementations of control applications for specific platforms in well-defined operational environments [7]. However, the strive to tackle performance issues has resulted in neglecting other relevant quality attributes of a software system, such as reusability, interoperability, and maintainability [7]. As a consequence, available solutions, both software and hardware, are incredibly heterogeneous. Additionally, neither software nor hardware has been truly standardized yet, thus making the development of MMRSs involving heterogeneous robots very intricate and somewhat frustrating for developers.

Mission planning and execution variability. MMRSs are called to operate in diverse usage scenarios and civilian missions. These scenarios might require specific quality of services, such as safety, real-time and resource constraints, timing requirements, etc for successful completion of the mission. Moreover, these systems operate in highly dynamic environments: (i) resources that these systems use (energy, memory, connection capabilities, etc.) are limited in capacity, (ii) resources might degrade or even disappear (e.g., wireless connection), (iii) they usually are not extensible during the system's lifetime, (iv) new resources might become available and contextual information typically vary. Quality criteria need to be preserved despite adaptation, which for these systems is the norm rather than the exception. Finally, the variety of missions discussed above implies an extensive assortment of interaction scenarios and involves variegated categories of final users, each with particular characteristics and needs. This calls for diversification in human-robot interfaces, information visualization, suitable abstraction, as well as proper information hiding. In this section we elucidate the core challenges that we identified as most relevant implications of the two dimensions of horizontal variability.

2.1 Software and hardware variability

We have identified three main software and hardware variability dimensions that occur when managing the development and adoption of MMRSs.

2.1.1 Platform neutrality

Nowadays, heterogeneity has become a common trait that characterizes robotic software systems, especially when it comes to the hardware composition on which the software system is meant to run. This heterogeneity, together with the lack of a standardized platform-agnosticity of robotic software solutions, makes cross-platform development intractable at the moment. That is the reason why a lift in the level of abstraction at which robotic software is developed towards platform neutrality. Such a lift cannot disregard an investigation of a basic common infrastructure to assist the development of robotic software, user interfaces and management of diverse application domains is strongly needed. The ideal goal would be the standardization of basic systems functions, scalability to different robots and platform variants, transferability throughout the network, integration from multiple suppliers, maintainability throughout the robot life-cycle and software updates and upgrades over the robot's lifetime.

2.1.2 Systematic reusability

Robotic software developers often experience a sense of frustration when they have to develop from scratch a "new" application, even if very similar to previous releases for different projects, because they do not have at their disposal appropriate support for capturing and exploiting commonalities [7]. The implementation of a specific functionality is initiated by domain experts that design the system and take key decisions on algorithms that then the software developer implements in terms of code. The lack of systematic, disciplined, and quantifiable robotic software engineering methodology, as well as comprehensive abstraction mechanisms for handling the increasing complexity of robotic software systems, lead to countless, similar, but not congruent, isolated solutions which cannot be easily reused and combined. For this reason, tackling this challenge is of paramount importance in order to make development of robotic software sustainable on the market. Systematic reusability calls also for appropriate integration means (e.g., architectures, connectors, integration patterns) that ease collaboration and integration of existing solutions while still satisfying the system's purpose under a controlled degree of uncertainty. These approaches should be efficiently performed at run-time and should be able to guarantee different quality attributes both in isolation and, more importantly, in combination since they usually heavily affect each other. Therefore, specific effort shall be devoted to trade-off analysis of quality attributes for enabling optimization with guarantees.

2.1.3 Orchestrating concurrency

MMRSs are meant to be loosely coupled and concurrent. This means that the global state of the system might be suddenly changed by actions of the individual robots that operate independently of each other. The inherent concurrence typical of MMRSs makes their development and execution extremely challenging. The number of robots collaborating in the MMRSs can be arbitrary, and each of them can be programmed, re-programmed, or can even automatically adapt to contextual information in an independent manner with respect to the other robots composing the MMRSs. Moreover, MMRSs may expose emerging properties that represent unexpected behaviours that stem from interactions between the system parts and the context [15]. Emergent properties might be beneficial, but they can be also harmful, e.g., if they compromise system's safety. For these reasons, suitable orchestration mechanisms shall be provided to suitably control concurrency of MMRSs.

2.2 Mission planning and execution variability

When planning and executing missions involving multi-robot systems there are different variability aspects that have to be taken into account as discussed in the remaining of this section.

2.2.1 Context awareness

The development of context-aware applications is always a challenging and complex task. These applications adapt to changing context information, where context might be physical, computational, and user-related. Modelling real scenarios require processing of context facts and reasoning upon them to attain a form of context information that is appropriate for the different missions. Moreover, context information is usually collected from different sources that differ in quality and that are often failure prone [4]. To minimize the possibility to run into failing situations at run-time derived by faulty context information, development and representation of the context for each mission should be supported by adequate modelling and reasoning techniques. These techniques need to address the variegation of context information types and the relationships among them, but also to cope with the uncertainty that comes along with context information.

2.2.2 Cope with uncertainty

On the one hand MMRSs are required to live under uncertainty: they are intrinsically dynamic and might evolve according to available resources, dynamic contextual information, etc. On the other hand, they are subject to rigid development practices imposed by certifications and compliance to standards that are inescapable. Therefore, in order to ensure the quality users expect, the evolution of MMRSs should be controlled by suitable mechanisms able to ensure that defective and malicious adaptations do not affect the entire MMRSs, especially when it comes to safety and reliability.

2.2.3 Dynamic discoverability of available resources

MMRSs operate in highly dynamic environments where uncertainty and unforeseen changes can unanticipatedly occur at any time. Matching the needs of the mission with the available resources right away is one of the basic and key aspects when designing MMRSs. In order for this to happen, we have to look for suitable robots satisfying a given set of constraints, and for which we have the access permission in the particular moment. In highly dynamic environments, new or recovered robots can show up in the environment at any time and, in order for the MMRSs to exploit them, a mechanism that handles dynamic discoverability of available resources and constraints is pivotal and it is particularly crucial in the case of new, unknown robots entering the MMRSs. Thereby, the system should be able to communicate and recognize the robots' characteristics and to do the necessary adjustments to mission plans and strategies accordingly.

2.2.4 Subsystems interoperability

MMRSs are characterized by the integration of heterogeneous subsystems that collaborate towards a common goal. Managing heterogeneity of the individuals forming the MMRSs is a challenging task due to the nature of the subsystems that differ in resources, protocols, platforms etc. In order for the MMRSs to provide appropriate coordination between heterogeneous subsystems, flexibility has to be enforced in many different aspects such the ability to make individuals using different communication protocols to communicate, the possibility to easily modify a robot's task, as well as the capability of plug and play both applications and individuals in the running MMRSs.

2.2.5 Human-robot synergy

Human-robot synergy has been delineated by Goodrich et al. [11] as the problem of understanding and shaping the interactions between one or more humans and one or more robots. Since robotic software systems are created and used to do work for and with humans, these interactions permeate in all of robotics, from design to deployment, from execution to maintenance, and even in the autonomous branch. For this reason it is of cardinal importance to invest effort in evaluating capabilities of both humans and robots, and designing proper means of interaction. We share Goodrich and Schultz's claim that, to properly address this task, the robotic software designer should take into account five main characteristics that affect interactions: (i) level of robot autonomy, (ii) type of information exchange, (iii) MMRSs' composition (in terms of individual sub-systems), (iv) learning of humans and robots in the MMRSs, and (v) mission (or task) definition and configuration.

3. PROPOSAL

Considering MMRSs through the aspect of horizontal variability, we had identified the challenges that emerge out of it. In this section we will introduce our research problem, define our objectives and explain our methodology on how we plan to tackle the problem.

3.1 Research problem

MMRSs are critical systems intensively validated at design-time which give them a predictable behaviour at run-time. However, there is a growing need for more flexible adaptive systems which will be able to operate in dynamic environments, coping with unanticipated situations, while still being able to ensure safety properties. To ensure safety, variants of the system have to be checked against the possible threats. In highly adaptive systems, the number of system variants usually grows exponentially and thus, prolongs the safety check to an infeasible degree. To address this issue, safety checks can be postponed to run-time of the system [21]. Realizing safety assurance at run-time still remains one of the biggest challenges in MMRSs [6]. Furthermore, the integration at run-time and the adaptation to dynamically changing environment contexts greatly complicates the assurance of all functional and quality properties. A particular challenge is the management of multiple quality properties from different domains. To consider multiple qualities simultaneously, their interdependences and interferences needs to be determined and considered in all phases of the system. In this context, it is of utmost importance to consider the interdependence and interference of safety and adaptability at run-time. Thus, designing MMRSs that will be safe and adaptable at the same time is still a concern that remains to be addressed [6].

For example, we can take in consideration a system that is represented by a swarm of quadrotors which aim to safely and efficiently perform a monitoring mission [9]. In case of a fault of one quadrotor, two problems need to be resolved: (i) managing the quadrotor with the fault and (ii) reconfiguring the other quadrotors, so they will be able to accomplish the mission. The reconfiguration takes into account the status of each quadrotor of the swarm and, depending on the mission, it performs the necessary adaptation actions that will not jeopardise the safety properties of the system. Let us assume that during the mission a failure happens on one of the engines of a quadrotor. In order to accomplish the mission, the swarm need to be reconfigured. The quadrotor that had problems can be reconfigured to return to the base station, while the reconfiguration of the other drones can be done by recomputing and reassigning the adapted monitoring areas.

3.2 Research Objective

The problem described before is the result of the early stage of development in which this area of research currently is. For this reason, we claim that further effort is needed to investigate this area. Given the problem description and the example from the previous section, we raised the following research question:

How to preserve safety while enabling adaptability at run-time in MMRSs?

In order to approach this question we identified several sub-research questions that we plan to address:

1. *How to model safety and adaptability properties for MMRSs in a dynamic run-time context?*

2. *How to enable safety analysis in relation to the system context changes and system adaptability in MMRSs?*

Safety is defined as the absence of catastrophic consequences on the user(s) and the environment [3], while adaptability is the extent to which a software system adapts to changes in its environment [23]. An adaptable system can tolerate changes in its environment without external intervention. Whenever the system's context changes the system has to decide whether it needs to adapt. In this proposal we aim to address the problem of proper handling safety and adaptability properties at run-time as one of the very important challenges in MMRSs [7].

In order to address these issues we will be using the very promising Model-driven engineering methodology (MDE). MDE is successfully used and showed positive results in other domains (e.g. avionics, automotive and telecommunications). Under this perspective, MDE can be a viable direction for successfully supporting future MMRSs. In particular, by means of MDE it is possible to systematically concentrate on different levels of MMRSs abstractions at which all involved stakeholders can operate [14] for (i) improving the quality of MMRSs in terms of, e.g., safety, adaptability, reliability and reusability, (ii) reducing the intrinsic variability and complexity of today's MMRSs, and (iii) promoting the reuse of software and hardware components across MMRSs. As a result we expect to significantly improve the state of the art in the field, and make contribution towards a standard methodology which will guarantee safety and adaptability properties within the domain of MMRSs.

The final objective of the proposal is to design, develop and validate a modelling framework based on a methodology which explicitly takes into account safety and adaptability at run-time and provides guidelines and principles for researchers and developers to design MMRSs. The work proposed here aims at providing important contributions towards assurance of safety and adaptability properties for MMRSs in a dynamic run-time context.

3.3 Research Methods

In this section we will illustrate some of the methodologies, tools and concepts that we will be using throughout this work. In order to identify and evaluate the relevant literature related to our research questions, currently we are doing a systematic mapping study. A systematic approach will give us an objective, thorough summary and critical analysis of the relevant high-quality literature available on this topic [8, 16]. In the systematic mapping study we aim to identify, classify, and understand existing research on safety in mobile robots rather than just focusing on safety in MMRSs, in order to understand the gaps of the current research on a more general and broader topic.

In order to build our modelling framework, we are planning to use Model-driven engineering (MDE). Many MDE technologies and concepts (like MOF, OCL, UML, model transformations) are widely known in both research and practice, and have been standardized by international consortia like the Object Management Group (OMG). Success stories on the adoption of MDE technologies in industrial contexts might be found in [17]. In the robotics field, having standard technologies is extremely valuable since it will make less painful to develop, maintain, and reuse software tools and components across different projects and organizations.

In the following we will discuss how MDE can be used to deal with the run-time adaptation. A promising approach for managing complexity in run-time environments is to develop adaptation mechanisms that leverage software models, referred to as models@run-time. As we noticed in the previous sections, MMRSs are required to adapt dynamically at run-time, often to situations unforeseeable at design time. Application code generated from design time models fails to provide the required flexibility, as the design rationale held in the models is not available at run-time. To tackle this issue the use of run-time models (or models@run.time) has been proposed. In contrast to development models, run-time models are used to reason about the operating environment and run-time behaviour, where different dimensions need to be balanced, including resource-efficiency (time, memory, energy), context-dependency (time, location, platform), as well as personalization (quality-of-service specifications, profiles) which are all factors that influence on the safety.

Furthermore, in order to ensure safety in MMRSs we will define precise and cogent models that describe systems' guarantees and assumptions. To guarantee the correct behaviour of the system i.e. to ensure the safety properties, formal modelling and formal verification analysis techniques should also be considered. In [29] is given the state of the art in verification and validation in Cyber-Physical Systems. A lot of the approaches and tools presented there can be used and adopted for verification of our defined properties in MMRSs. The challenge will be to find and adopt appropriate safety verification methods for adaptable systems performing in a dynamic environment.

As a last part of our work, we are planning to evaluate the quality of our proposed contributions both from an experimental and theoretical point of view. In order to validate our contributions in practice we are planning to perform a quasi-experiment [28]. In addition to it, we will try to arrange a case study [28] with a company or a research partner. In this direction, we are planning to use the experience acquired from the VASA project and FLYAQ and use the collaboration between the projects to validate our approach. VASA[1] is an autonomous underwater vehicle developed at Mälardalen University, Sweden, while FLYAQ[2] is a platform for mission planning of autonomous quadrotors.

4. RELATED WORK

In this section we will give the scope and analysis on studies which completely or partially are addressing the topic of safety and adaptability in mobile multi-robot systems.

The authors of [24] present a general survey of various publications that focus on mechanical design and actuation, controller design, and safety criteria and metrics used to validate safety of domestic robots during unexpected collision between a robot and a human user. The focus on the survey is on the mechanical and controller design, while not taking in consideration safety from a software engineering point of view.

A review about Human-Robot Interaction (HRI) is presented in [12]. It attempts to identify the key themes and challenges from multiple perspectives, as HRI requires understanding and comprehension of multiple domains related to people, robotics, design, cognitive psychology etc.

In [25] a survey investigating safety issues in human-robot interactions is proposed. It starts with a review of safety issues in industrial settings, then shifting focus on safety issues related to mobile robots that operate in dynamic environments. It gives general ideas and directions of possible hazards and methods used for risk reduction, pointing out risks which have been introduced with the development of modern robotic systems.

[18] is a PhD work that proposes a novel methodology for ensuring safety for human-robot interaction during task planning. The outcome of the thesis is a safety monitoring system that tries to ensure the required level of safety.

Another PhD thesis [13] proposes and discusses real-time collision avoidance methods, i.e. how to design pre-collision strategies to prevent unintended contact in human-robot interaction. In this thesis, the human is placed as the central entity for evaluation of the safety.

Moreover, the authors of [2] present the state of the art in the field of safe and dependable physical human-robot interaction undertaken within two projects: PHRIDOM (Physical Human-Robot Interaction in Anthropic Domains) and PHRIENDS (Physical Human-Robot Interaction: dependability and safety). Results from different research groups about possible metrics for the evaluation of safety, dependability and performance in physical human-robot interaction are presented.

Furthermore, in [27] are presented methods, tools and theories developed within the ASCENS project (integrated project funded by FP7) which aims to ensure correct behavior of ensembles - collective autonomic systems able to adapt to the environment at runtime. To validate their contributions they conduct a swarm-robotics case study that takes into account ensembles of cooperative self-aware robots.

Moreover, in [26] are presented state-of-the-art results of workshops of the InterLink working group which cover topics for methods, languages and tools for ensemble engineering.

In [19] are represented several MDE software development approaches implemented in the robotics domain, but none of them provides explicit models for quality properties.

Finally, in [20] there is a discussion about an attempt for a systematic framework design aiming to support some quality attributes that are relevant in the robotics domain. This framework aims at improving the impact on software quality by addressing several quality attributes in single robotic systems, not taking in consideration cooperative multi robotic systems. We consider it is a good starting point for analysis of these types of frameworks in order to check to which level safety and adaptability properties are addressed.

The work we are addressing in this proposal is different from the one considered previously in terms of: (i)analysis of the interpendence and interference between safety and adaptability; (ii) focus on MMRSs at run-time and only from software engineering perspective and (iii) inclusion of different types of robots, rather than focusing on one.

[1]http://www.mrtc.mdh.se/projects/ralf3/robosub/index.php
[2]http://www.flyaq.it/

5. CONCLUSIONS

This paper presented an approach towards the assurance of safety and adaptability properties for MMRSs in a dynamic run-time context. The approach presented here is using the very promising MDE methodology. The next steps towards the dissertation are: (i) completing the systematic mapping study on existing research on safety in mobile robots, (ii) designing a methodology and modelling framework which explicitly takes into account safety and adaptability properties at run-time and (iii) validating our contributions by performing a quasi-experiment and case study.

6. REFERENCES

[1] Safety standards: International organization for standardization (iso) for robots and robot systems integration. http://goo.gl/IvpraB, October 2014.

[2] R. Alami, A. Albu-Schaeffer, A. Bicchi, R. Bischoff, R. Chatila, A. De Luca, A. De Santis, G. Giralt, J. Guiochet, G. Hirzinger, et al. Safe and dependable physical human-robot interaction in anthropic domains: State of the art and challenges. In *Proc. IROS*, volume 6. Citeseer, 2006.

[3] A. Avižienis, J.-C. Laprie, and B. Randell. Dependability and its threats: a taxonomy. In *Building the Information Society*, pages 91–120. Springer, 2004.

[4] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161–180, 2010.

[5] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, 2013.

[6] D. Brugali. *Software engineering for experimental robotics*, volume 30. Springer Science & Business Media, 2007.

[7] D. Brugali and E. Prassler. Software engineering for robotics. *IEEE Robotics and Automation Magazine*, 16(1):9–15, 2009.

[8] P. Cronin, F. Ryan, and M. Coughlan. Undertaking a literature review: a step-by-step approach. *British Journal of Nursing*, 17(1):38, 2008.

[9] D. Di Ruscio, I. Malavolta, and P. Pelliccione. Engineering a platform for mission planning of autonomous and resilient quadrotors. In *Software Engineering for Resilient Systems*, pages 33–47. Springer, 2013.

[10] E. Garcia, M. A. Jimenez, P. G. De Santos, and M. Armada. The evolution of robotics research. *Robotics & Automation Magazine, IEEE*, 14(1):90–103, 2007.

[11] M. A. Goodrich and A. C. Schultz. Human-robot interaction: A survey. *Found. Trends Hum.-Comput. Interact.*, 1(3):203–275, Jan. 2007.

[12] M. A. Goodrich and A. C. Schultz. Human-robot interaction: a survey. *Foundations and trends in human-computer interaction*, 1(3):203–275, 2007.

[13] S. Haddadin. *Towards Safe Robots: Approaching Asimov's 1st Law*. Springer Publishing Company, Incorporated, 2013.

[14] B. Hailpern and P. Tarr. Model-driven development: The good, the bad, and the ugly. *IBM Systems Journal*, 45(3):451–461, 2006.

[15] C. W. Johnson. What are emergent properties and how do they affect the engineering of complex systems? *Reliability Engineering & System Safety*, 91(12):1475 – 1481, 2006. Complexity in Design and Engineering.

[16] B. Kitchenham. Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33:2004, 2004.

[17] G. Liebel, N. Marko, M. Tichy, A. Leitner, and J. Hansson. Assessing the state-of-practice of model-based engineering in the embedded systems domain. In *Model-Driven Engineering Languages and Systems*, pages 166–182. Springer, 2014.

[18] O. Ogorodnikova. *Human Robot Interaction: The Safety Challenge*. PhD thesis, Budapest University of Technology and Economics, 2010.

[19] A. Ramaswamy, B. Monsuez, and A. Tapus. Model-driven software development approaches in robotics research. In *Proceedings of the 6th International Workshop on Modeling in Software Engineering*, pages 43–48. ACM, 2014.

[20] M. Reichardt, T. Föhst, and K. Berns. On software quality-motivated design of a real-time framework for complex robot control systems. In *Proceedings of the 7th International Workshop on Software Quality and Maintainability (SQM), in conjunction with the 17th European Conference on Software Maintenance and Reengineering (CSMR)*, 2013.

[21] D. Schneider and M. Trapp. A safety engineering framework for open adaptive systems. In *Self-Adaptive and Self-Organizing Systems (SASO), 2011 Fifth IEEE International Conference on*, pages 89–98. IEEE, 2011.

[22] T. Skrzypietz. *Unmanned Aircraft Systems for Civilian Missions*. BIGS, 2012.

[23] N. Subramanian and L. Chung. Metrics for software adaptability. *Proc. Software Quality Management (SQM 2001), April*, 2001.

[24] T. S. Tadele, T. J. Vries, and S. Stramigioli. The safety of domestic robotics: A survey of various safety-related publications. *IEEE Robotics & Automation Magazine*, 21(3):134–142, 2014.

[25] M. Vasic and A. Billard. Safety issues in human-robot interactions. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 197–204. IEEE, 2013.

[26] M. Wirsing, J.-P. Banâtre, M. Hölzl, and A. Rauschmayer. Software-intensive systems and new computing paradigms. 2008.

[27] M. Wirsing, M. Hölzl, N. Koch, and P. Mayer. *Software Engineering for Collective Autonomic Systems*. Springer, 2015.

[28] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in software engineering*. Springer, 2012.

[29] X. Zheng, C. Julien, M. Kim, and S. Khurshid. On the state of the art in verification and validation in cyber physical systems. 2014.