

# RoboMAX: Robotic Mission Adaptation eXemplars

Mehrnoosh Askarpour  
McMaster University  
Hamilton, Canada  
askarpom@mcmaster.ca

Christos Tsigkanos  
Technische Universität Wien  
Vienna, AVienna University of Technologyustria  
christos.tsigkanos@dsg.tuwien.ac.at

Claudio Menghi  
University of Luxembourg  
Luxembourg, Luxembourg  
claudio.menghi@uni.lu

Radu Calinescu  
University of York  
York, UK  
radu.calinescu@york.ac.uk

Patrizio Pelliccione  
Gran Sasso Science Institute, Italy  
Chalmers | University of Gothenburg  
Gothenburg, Sweden  
patrizio.pelliccione@gssi.it

Sergio García, Ricardo Caldas  
Chalmers | University of Gothenburg  
Gothenburg, Sweden  
{gsergio, ricardo.caldas}@chalmers.se

Tim J von Oertzen  
Johannes Kepler University Linz  
Kepler University Hospital  
Linz, Austria  
tim.vonoertzen@kepleruniklinikum.at

Manuel Wimmer, Luca Berardinelli  
Johannes Kepler University Linz  
Linz, Austria  
{manuel.wimmer, luca.berardinelli}@jku.at

Matteo Rossi, Marcello M. Bersani  
Politecnico di Milano  
Milan, Italy  
{matteo.rossi, marcellomaria.bersani}@polimi.it

Gabriel S. Rodrigues  
University of Brasilia  
Brasília, Brazil  
siqueira.rodrigues@aluno.unb.br

**Abstract**—Emerging and future applications of robotic systems pose unique self-adaptation challenges. To support the research needed to address these challenges, we provide an extensible repository of robotic mission adaptation exemplars. Co-designed with robotic application stakeholders including researchers, developers, operators, and end-users, our repository captures key sources of uncertainty, adaptation concerns, and other distinguishing characteristics of such applications. An online form enables external parties to supply new exemplars for curation and inclusion into the repository. We envisage that our RoboMAX repository will enable the development, evaluation, and comparison of self-adaptation approaches for the robotic systems domain.

## I. INTRODUCTION

In early applications, robots were deployed within controlled industrial facilities, and dedicated to the execution of a single, predefined task (i.e., *mission*). However, with 173,000 professional service robots sold in 2019 alone [1], robotic systems are now ubiquitous, and are increasingly used in domains ranging from healthcare and logistics to infrastructure inspection and industrial manufacturing. These systems are deployed in uncontrolled real-world environments, where the robots are exposed to multiple sources of uncertainty that can only be handled through *self-adaptation*. In many applications, the robots are additionally required to perform complex missions, increasingly in collaboration with other robots [2, 3] or humans [4, 5]. As such, their control software needs to perform monitoring, analysis, planning and execution activities that are the hallmark of self-adaptive systems.

The development of the control software for robotic systems is a recognized challenge that demands new software engineering methods and techniques [6], not least to achieve self-adaptation. This demand has increased the software engineering community’s interest in understanding the needs of the robotics domain [7], and in developing solutions that address these needs [8, 4]. In this context, a major difficulty faced by researchers and practitioners has been the scarcity of model problems [9] that can be readily used to guide research into, and to support the development of, such solutions.

We address this gap by providing a repository of robotic mission adaptation *exemplars* that can be used to develop, evaluate, and compare self-adaptation approaches for robotic applications. Specifically, our contributions are as follows:

- We define a *robotic mission adaptation exemplar* as a semi-structured model problem that captures key aspects associated with the self-adaptation needs of a robotic mission. These aspects include, for instance, sources of uncertainty and adaptation concerns for the mission.
- We systematically classify the exemplars by recording extensive meta-information highlighting the characteristics of the exemplars, such as the context in which each robotic mission takes place, the characteristics of the involved robot(s), and adaptation-relevant specifics.
- We set up an exemplar repository (RoboMAX) [10, 11], and a method for collecting further exemplars from the community; once curated by our RoboMAX team, these additional exemplars will be used to extend the repository.

- We bootstrap the repository with a set of 11 exemplars co-designed with end-users and domain experts (i.e., researchers, developers and operators of robotic systems).
- We analyze the meta-information of the 11 exemplars of the RoboMAX repository and provide an outlook of potential uses of the robotic mission adaptation exemplars.

Similar to the ‘*Feed Me, Feed Me*’ IoT exemplar [12], our RoboMAX exemplars provide the high-level requirements and key contextual information associated with the considered systems and their adaptation concerns. In this way, our repository serves a different purpose than, and complements, existing SEAMS exemplars, which provide simulators for simple robotics applications (UNDERSEA [13], Dragonfly [14] and DARTSim [15]) or generic frameworks for developing self-adaptive cyber-physical applications (DEECo [16], Intelligent Ensembles [17]), or datasets (AMELIA [18]).

The rest of the paper is structured as follows. Sec. II presents the structure of the exemplars. Sec. III describes our exemplar collecting methodology. Sec. IV showcases a characteristic exemplar and describes the repository. Sec. V provides a meta-analysis, outlines lessons learned and highlights potential uses of robotic mission adaptation exemplars. Sec. VI concludes.

## II. STRUCTURE OF ROBOMAX EXEMPLARS

In this section, we describe the structure adopted for the exemplar definition. Each exemplar revolves around a model problem; besides a description of the underlying scenario, we associate with each exemplar some meta-information to explicitly record its specific features. The classification we adopt intends to i) precisely define the context where the scenario takes place, ii) describe robotic system peculiarities, and iii) highlight adaptation specifics inherent in the scenario. The aspects we describe for exemplars are explained below.

**Description.** A high-level description of the robotic mission and its environment, as elicited from stakeholders and domain experts. This is a natural-language presentation, akin to a user story [20]. It intends to capture essential behavior that the robotic system should exhibit.

**Source.** A description of whether the exemplar comes from the academic or the industrial domain.

**Domain.** The identification of the application domain where the robotic adaptation scenario takes place (e.g., healthcare, logistics or manufacturing).

**Multirobot.** A description of whether the exemplar represents a single or multi-robot application. If the latter [6], we further characterise it as *Swarms* (i.e., robots with the same behavior) or *Teams* (i.e., robots with distinct behaviors).

**Robot type.** The definition of the robot(s) employed in the exemplar including any specialized capabilities they possess. Robots have individual features, capabilities, and corresponding application requirements; a crucial step in robotics system design and development is defining a mapping between application requirements and robot capabilities. We loosely follow the taxonomy of [21], characterizing robots in terms of:

- **Robot Features.** A description of the core hardware and software robot components and their specification, including

sensors, actuators, type of locomotion, electronics and materials. In this work, we consider three types of robot features: (i) parameter adaptability (whether the robots can adapt their behaviour by changing some parameters, e.g., their speed), (ii) component adaptability (whether they can add or remove components), and (iii) mechatronics configurability (whether they can change their mechatronics configuration).

- **Technical Capabilities.** The higher-order capabilities enabled by the robot’s hardware and software features, interaction capabilities and skills and behaviour that emerge from robot interactions. For example, the “interaction with a human” is a technical capability of a robot. We consider three types of technical capabilities: (i) robot capabilities (e.g., perception and interpretive, robot task abilities, actions and envisioning capabilities), (ii) interaction capabilities (e.g., social or physical interaction with the environment, cognitive interaction with other information systems), and (iii) intelligence capabilities including physical morphological (e.g., visual-spatial skills), cognitive (e.g., learning or mathematical logic), social (e.g., emotional behaviour), and collective intelligence (e.g., collaboration).
- **Operational Capabilities.** The robot profile in terms of conceptual concerns such as cost (e.g, the energy consumed by robots), duration (e.g., timing requirements such as completing a task in two minutes), safety (e.g., preventing collisions, crushing, and injuries by mechanical parts), security (e.g., dealing with proprietary and sensible assets), testing (e.g., employing robots to test other devices), training (e.g., required training during or before operation), acceptance (e.g., treating user acceptance as a primary requirement), or usability (e.g., necessary usability requirements).

**Sources of uncertainty.** A description of a variety of circumstances from which the uncertainty originates. Uncertainty, defined as ‘*any deviation from the unachievable ideal of completely deterministic knowledge of the relevant system*’ [22], is the key driver in engineering self-adaptation in systems. Self-adaptation enables a system to accommodate its execution within different contexts, achieving continuous satisfaction of its requirements. To characterize uncertainties within the robotic model problems, we follow the taxonomy of [19]. We adapt the taxonomy to account for the peculiarities of the robotics domain; we especially note that sources of uncertainty described in [19] may be still applicable to robotic adaptive systems. Sources of uncertainty are classified as:

- **Model or State Uncertainties.** The representation of information or knowledge that the robot employs may be incomplete, contradictory, overly complex or incorrect; an example is a discrepancy between the cyber-physical environment and its perceived model by the robot.
- **Adaptation Functions Uncertainties.** The uncertainties inherent in the variability space, decision making and coordination functions that adaptation must handle.
- **Mission Uncertainties.** The intended behavior of the application may be poorly specified, change, or become outdated.
- **Environment Uncertainties.** The variability of the overall context where the robot operates at runtime, including

TABLE I  
SOURCES OF UNCERTAINTY – ADAPTED FROM [19].

Class	Source	Causes
Model or state uncertainty	Abstraction	Caused by omitting certain details and information from models.
	Incompleteness	Caused by lack of knowledge about parts of the internal robot or external environment state.
	Model drift	Caused by discrepancy between the state of models maintained by the robot and the actual state of the robotic application.
	Different sources of information	Caused by differences between representations of information provided by different sources (e.g., camera vs presence sensor).
Adaptation functions uncertainty	Complex models	Caused by complexity of runtime models representing the managed robotic application.
	Sensing	Caused by robotic sensors which are inherently imperfect.
	Actuation	Caused by robotic actuators of which the effects may not be completely deterministic.
Mission uncertainty	Variability space	Caused by the size of the variability space that adaption functions need to handle.
	Automatic learning	Caused by machine learning techniques employed of which the effects may not be fully predictable.
	Decentralization & Coordination	Caused by decision making by robots in teams or swarms, of which the effects may not be predictable.
Environment uncertainty	Future mission changes	Caused by potential changes in the mission that could not be completely anticipated.
	Mission specification	Caused by lack of deterministic and precise specification of mission.
	Outdated mission	Caused by overlooking that the mission is outdated.
Capabilities uncertainty	Execution context	Caused by the inherent unpredictability of execution contexts, exacerbated by the complexity of the environment of the robotic application.
	Human in the loop	Caused by the inherent unpredictability of human behavior interacting with the robot.
Capabilities uncertainty	New or defunct capabilities	Caused by new availability or no longer existing or functioning robotic capabilities.
	Changing capabilities	Caused by dynamicity of capabilities in the robotic system.

unpredictable interaction with humans.

- **Capabilities Uncertainties.** A description of uncertainty related to the robot capabilities. Robots have individual features that they employ to achieve their mission, including technical and operational capabilities. However, those are not static; they may be changed, removed or new ones may be available to the robot.

**Types of adaptation.** The identification of self-\* properties that the robotic system may require to handle uncertainties previously identified. Based on the classification of [23], those can include self-management (i.e., the system has at least one self-\* property), self-stabilization (i.e., starting from an arbitrary initial configuration, it recovers to a legal configuration and then remains in that configuration), self-healing (i.e., given a set of actions, the occurrence of one of these actions causes at most a temporary violation of a property), self-organization (i.e., it maintains, improves or restores a safety property following certain actions), self-protection (i.e., it continuously maintains a safety property), self-optimization (i.e., starting from an initial configuration, it improves the value of an objective function), self-configuration (i.e., it can change its configuration to restore or to improve some safety property) and self-scaling (i.e., it maintains or improves a property during the occurrence of a set of actions).

**Adaptation concerns, constraints and other factors.** The description of adaptation concerns, targeting the impact of different sources of uncertainty on the system—e.g., variability in the sensing of the cyber-physical environment may lead to failures (i.e., safety and/or security concerns) that can be mitigated by self-healing functions. Adaptation factors may include performance factors, timing, utility, cost, or trade-offs.

**Authors, Affiliations and Contact Details.** Names and affiliations of the exemplar authors—i.e., the scenario providers.

**Source and further resources.** The description of the model problem origins and an indication of any further material. It aims to provide the basis for investigating the particular problem context, auxiliary resources or existing artifacts.

### III. METHODOLOGY

To collect robotic mission exemplars from domain experts and practitioners, we adopted Google Surveys. We defined a Google Form for the collection of the meta-information described in Sec. II and made it available online [24]. The form contains 36 questions—28 open questions, and 8 multiple-choice questions—organized in eight sections: *exemplar*, *usage context*, *model or state uncertainties*, *adaptation functions uncertainties*, *mission uncertainties*, *environment uncertainties*, *capabilities uncertainties*, and *adaptation*. The *exemplar* section collects general information related to the exemplar—i.e., the name, the description, and the authors’ contact details. The *usage context* section collects information of the source, the domain, multi-robot, robot features, robot operational and technical capabilities. The *model or state uncertainties*, *adaptation functions uncertainties*, *mission uncertainties*, *environment uncertainties*, *capabilities uncertainties* sections collect the information presented in Table I. Finally, the *adaptation* section collects adaptation concerns, constraints and other factors, and the source and further resources.

To maximize the usability of RoboMAX, users can select the details of the meta-information associated with the exemplars. Our form enables adding an exemplar with a minimal amount of meta-information or carefully choosing its meta-information. Therefore, excluding the author’s information and the scenario description, the rest is optional. For multiple-choice questions, users can provide additional answers, that are not part of the one the form provides by default. Note that,

the more detailed is the meta-information added by the user, the more likely is the exemplar to be used and discussed in the community. Therefore, we expect users to carefully select the meta-information associated with each exemplar. Authors will review and curate the repository, by ensuring that new exemplars adhere to the classification.

We added 11 exemplars to RoboMAX, each obtained by:

- 1) *Collecting robotic mission requirements.* Some of our robotic mission requirements come from a recent survey that analyzes the state of the art and practice of robotics software engineering [7] (i.e., six mission requirements) and the rest are defined in collaboration with industrial partners (i.e., five mission requirements). For the latter, we provided an initial example of a mission to show to practitioners what we mean by mission specification and the level of details we need; then, they provided a first version of the exemplars; we checked with them a refined version, and finalised them. We inserted all the exemplars via our survey form to evaluate its correct functionality and provided feedback to its designer (i.e., one of the authors not involved in the insertion process). After a few refinements, our form is now ready to be used directly by practitioners to insert new exemplars in RoboMAX.
- 2) *Analyzing the mission requirements and defining exemplars.* We analyzed each mission requirement and checked whether it involved any self-adaptation concern. We added the mission requirement to our dataset of robotic mission adaptation exemplars via our form.

#### IV. DATASET

We collected 11 exemplars in our repository [10], each containing at least one self-adaptation concern. Figure 1 presents an exemplar instance, “Vital Signs Monitoring”, which refers to a robotic challenge in the medical/healthcare domain.

The exemplar revolves around monitoring patients’ vital signs in a hospital setting. Typically, the process is performed by medical staff and involves checking various metrics such as heart rate, body temperature, and blood oxygenation. Employing robotics in such a setting would be highly desirable, as it would lessen the burden on medical staff, who would be called only when necessary, in case of emergency, medical treatment or robot failure. The scenario presents several challenges, which are specific to healthcare aspects but also caused by the uncertainty and the variability inherent in the hospital setting.

The robotic system described in the exemplar involves a single or multiple moving robots, each equipped with specialized medical sensor packages and human interaction features. Several sources of uncertainty exist, mainly in the adaptation functions, the mission and the environment. Human interaction, for instance, is recognized as highly variable and challenging to implement in a deterministic manner—a typical example is the robot failing to comprehend human responses. Non-determinism in the mission also poses challenges—the robot may be required to make decisions to fulfill its mission, especially in cases where behavior cannot be specified completely at design time. For instance, failure or

delay in contacting medical staff in an emergency. Those are exacerbated by the overall criticality of the healthcare domain, rendering environment uncertainties challenging. Robotics in healthcare operate within cyber-physical spaces, populated both by humans, in highly dynamic environments [25]. Overall applications need to avoid injury or hindrance to patients and medical staff, and also meet strict timing requirements. Finally, self-organization is required for multi-robot solutions—e.g., to avoid collisions and tasks distribution.

#### V. META-ANALYSIS

In this section, we discuss the results of our meta-analysis, we outline the lessons learned, and we highlight an outlook of potential uses of the exemplars. Our analysis aims at providing an overview of the features of RoboMAX exemplars. Considering the limited number of exemplars, we do not make any claim on the generalizability of our observations.

**Source.** As mentioned in Sec. III, our exemplars were extracted from a survey on robotics software engineering [7] and a research project proposal. We marked these exemplars as industry-sourced since they were defined in collaboration with industrial partners. Even though industrial exemplars are realistic and practical, we intend also to include academic exemplars, representative of future applications which are currently under lab testing/exploration. Having both exemplar types allows comparing academic and industrial exemplars that could lead to further research directions in this area.

**Domain.** The tags used to identify the domain of the exemplars are the following: Medical (9), Emergency (2), Logistics (2), Household (2), Food supply (1), Educational (1).<sup>1</sup> The domain analysis shows that robotic applications concern a variety of domains. In most of them, a robotic application has to interact with humans. Hence, self-adaptation is needed to handle the unpredictability of humans.

**Multirobot.** Ten of the exemplars were associated with the single-robot and four with the multi-robot application tag. Note that, when the exemplar mission can be executed by either a single-robot and multi-robot application, both tags are associated with it. For all the cases in which the multi-robot tag was selected, the exemplar contained a team of robots with distinct behaviors. In one of the cases, a robot of the team could also be replaced by a human. The results show that, while single-robot applications are still the majority, there is a growing interest in multi-robot applications. Indeed, single robot applications are no longer sufficient for the ever-growing complexity of robotic missions. Moreover, there is an increasing need for robots with collaborative capabilities. The presence of multiple robots with different capabilities makes self-adaptation critical for these types of applications.

**Robot Type.** The results of the tag analysis are as follows.

**Robot Features.** The tags used to identify the robot features are: parameter adaptability (8), component adaptability (2), and mechatronics configurability (1). Parameter adaptability is, by far, the most used in our exemplars. Indeed, robots usually

<sup>1</sup>Our form allows tagging a single exemplar with several domains.

<b>Name:</b> Vital Signs Monitoring	<b>Authors &amp; Affiliations:</b> Chalmers   University of Goteborg	<b>Domain:</b> Medical/Healthcare
<b>Description:</b> <i>Every two hours</i> , all patient's vital signs should be checked. Therefore, the assigned robot should visit every <i>occupied room</i> . Before entering the room, the robot must check <i>if the patient is available</i> for vital signs checking, and if there is <i>enough battery</i> to perform all the tasks inside the room. If the patient is not available, the robot should come back in <i>five minutes</i> . If there is not enough battery, the robot should recharge and the <i>tasks need to be reassigned</i> . Once inside the room, the robot should approach the patient, announce the procedure to be performed, and provide instructions to the patient. The failure of collecting any necessary vital signs should trigger a set of questions to assess the patient status for those signs that were failed to be collected. In case of non-response, an alert should be sent to the responsible nurse/doctor. Once the patient is checked, the robot should mark it as done and proceed to the next room. Patients subject to infectious diseases must be checked for heart rate, body temperature, and blood oxygenation. Patients subject to post-surgical conditions must be checked for heart rate and blood oxygenation. Patients with a history of diabetes must be checked for glucose levels in the blood. When a robot comes out of a room where a patient subject to infectious disease is being treated, it must be sterilized. All the patients in high risk must be checked in the first 15 minutes of the mission. Additional robots can be assigned to help in case of impossibility of performing the mission in the required time. The responsible manager for the site must be alerted if no combination of robots can complete the mission.		
<b>Robot Features:</b> Parameter adaptability (whether the robots can adapt their behaviour by changing some parameters) is required to adapt the robot behavior based on the battery level.		
<b>Technical Capabilities:</b> Robot capabilities (e.g., perception and interpretive, robot task abilities, actions and envisioning capabilities), and interaction capabilities (e.g., physical interaction with the environment, social interaction, cognitive interaction with other information systems).		
<b>Operational Capabilities:</b> Duration (the robot should be able to check whether the mission can be accomplished in 15 minutes).		
<b>Model or State Uncertainties:</b> Model drift (a discrepancy between the state of the model of the environments in which the robots are deployed may be caused by sensing inaccuracy).		
<b>Adaptation Functions Uncertainties:</b> Unclear how sterilization is performed. Unclear what behavior is intended when operation takes longer than expected and the robot fails to meet timing requirements. Unclear what behavior should be performed when human interaction fails—e.g., because patient's responses are unintelligible.		
<b>Mission Uncertainties:</b> Unclear what behavior is intended if the patient is not available for vital signs checking due to an emergency situation. Unclear what should the robot do when not having enough charge to reach the recharge station. Non-determinism in edge cases—e.g., not knowing to whom send alert if responsible medical staff is busy or unreachable.		
<b>Environment Uncertainties:</b> Operation within critical cyber-physical system, which may be highly variable and unpredictable. Timing requirements may fail to be satisfied for multiple reasons—e.g., the robot may fail to check patients within the defined 15 minutes because the ward is crowded and avoiding collisions with humans makes it stop.		
<b>Capabilities Uncertainties:</b> —	<b>Multirobot:</b> Single or Team.	<b>Types of adaptation:</b> Self-organization
<b>Adaptation concerns and other factors:</b> Timing constraints		<b>Source and resources:</b> —

Fig. 1. The Vital Signs Monitoring exemplar.

need to change some parameter value (e.g., speed) depending on their environment. Mechatronics configurability is present in one exemplar requiring the robot to change its configuration occasionally. Although only two exemplars are tagged with component adaptability, we believe it is intrinsically present in other exemplars since alternative robot behaviors can be encoded in different software components that should be plugged and played according to the environmental context.

*Technical Capabilities.* The following tags are used to identify the technical capabilities: robot capabilities (12), interaction capabilities (9), and intelligence capabilities (4). Robot capabilities include, e.g., the ability of a robot to sense and act on its environment. Therefore, as expected, they are used in all of our exemplars. Interaction capabilities are present in many exemplars that require robot-human interaction. Finally, three exemplars are tagged with intelligent capabilities that require cognitive intelligence—e.g., learning or logical-mathematical skills—to make robots understanding/replicating human behaviors. We believe that robot capabilities and interaction capabilities are inherently part of most robotic applications. We also believe that, as robotic applications become more complex, the need for intelligent capabilities increases. Therefore, we expect more exemplars with increasingly intelligent capabilities to be inserted in the future.

*Operational Capabilities.* The occurrence of the operational capabilities tags is the following: duration (5), safety (3), usability (3), security (2), re-usability (1), acceptance (1), cost (1), Versatility (1), and reliability (1). The results show

that a considerable number of exemplars (5) involve timing requirements and duration operational capabilities. Safety, usability, and security also appear frequently (in three, three, and two exemplars, respectively). Finally, re-usability, and reliability are required for one exemplar each. As robotic applications become more ubiquitous and are deployed in less controlled environments, the importance of operational capabilities such as safety and security will increase even more. In addition, responsibility for the specification of robotic missions is slowly passing from highly qualified developers, with considerable programming experience, to final users. Therefore, we expect the importance of usability, re-usability, and reliability to increase.

**Sources of uncertainty.** Figure 2 shows the number of exemplars that refer to each source of each uncertainty class of Table I. An analysis of the tags shows the following.

*Model or state uncertainty.* Sensing and model drift are the most common sources of model/state uncertainty in our exemplars. They are present in ten and eight exemplars, respectively. For most of the robotic exemplars, there is an intrinsic uncertainty in the sensor measures. This uncertainty may cause differences between the state of the models maintained by the robot and the actual state of the robotic application—i.e., it results in a model drift. One exemplar is tagged with abstraction due to the omission of certain details and information about the environment map. Incompleteness is present in three exemplars due to the inherent lack of knowledge of some parts of the model environment—i.e., the

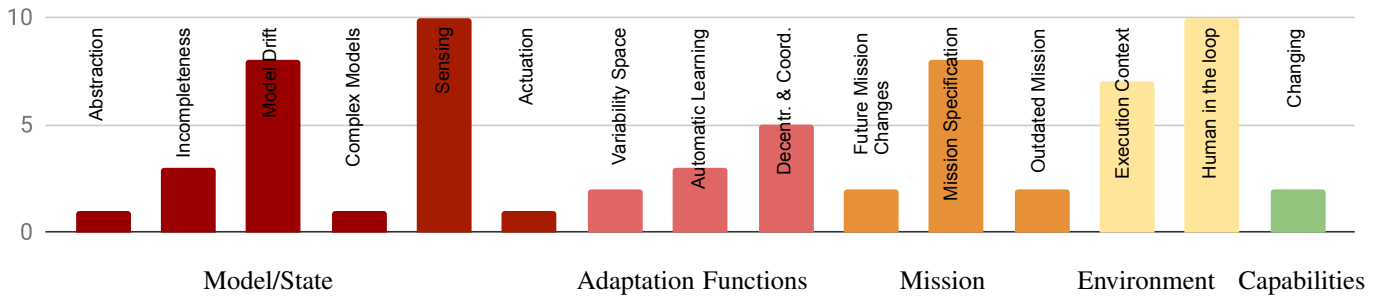


Fig. 2. Number of exemplars for each uncertainty sources described in Table I.

one representing the children’s behavior. Uncertainties due to complex models and actuation are present in one model each. None of the models was tagged with uncertainties caused by different sources of information. We believe uncertainties due to complex models, actuation, and different sources of information are not usually identified during the definition of robotic missions, since they are more related to the actual mission execution. Therefore, they are considered later, when the actual software of the robotic application is developed.

*Adaptation functions uncertainty.* Two exemplars have variability space uncertainties, because the size of the variability space that the adaptation functions need to handle is as large as the number of agents and of the actions they can perform. three exemplars have automatic learning uncertainties stemming from machine learning components that can be used to learn users’—e.g., children’s—behaviors, and that are subject to uncertainties. Five exemplars have decentralization and coordination uncertainties arising from collaboration activities among robots, which may unpredictably fail. We believe that adaptation function uncertainties will grow over time with the number/types of robots needed to accomplish robotic missions.

*Mission uncertainty.* Future mission uncertainties are present in two of the exemplars, where robots have to entertain children. The mission changes depending on the age, preferences, and hobby of the children the robot has to entertain. Mission specification uncertainties are present in eight exemplars, where a deterministic and precise specification of the mission is missing for some unpredictable conditions or events that can occur in the robotic application environment. In two exemplars, robots must perform tasks within 30 minutes, highlighting the time-sensitivity of the respective missions. However, the tasks to be performed by the robot when the mission is outdated might lead to uncertainty. We believe that the robotic mission adaptation exemplars are usually subject to mission uncertainties. Indeed, it is impractical, if not even impossible, for robotic mission designers to fully document the behavior of the robotic applications for all the possible changes that can occur in their environment.

*Environment uncertainty.* Seven exemplars have execution context uncertainties, manifesting every time there is uncertainty caused by the unpredictability of the environment. For example, a receptionist robot that welcomes people to a hospital is subject to uncertainties dictated by moving items and

devices. Ten exemplars have human-in-the-loop uncertainties. For instance, the receptionist robot is subject to uncertainties dictated by unexpected user behavior.

*Capabilities uncertainty.* None of our exemplars contains new or defunct capability uncertainties. Specifically, in none of them robots were discovering new capabilities—e.g., discovering new actuators—over time. Changing capabilities uncertainties were present in two scenarios that required the robots to use different actuators or change them over time.

**Types of Adaptation.** Self-management, self-optimization, self-protection, and self-configuration adaptations are present in ten, seven, three, and three exemplars, respectively. Self-optimization ensures that the robotic application satisfies its requirements by improving the value of an objective function. Self-protection aims at maintaining a safety property. Finally, self-configuration changes the configuration of the system to restore or improve some property.

**Adaptation concerns, constraints, and other factors.** Timing constraints, utility, and cost are present in six, three, and one exemplar, respectively. Timing constraints specify time-related concerns, such as patients must be checked in the first 15 minutes of the mission. Utility captures the goal of the robotic application through utility functions. Costs represent other concerns that negatively impact the satisfaction of the mission, such as energy consumption.

## VI. CONCLUSION

Motivated by the wider research community’s need for exemplars to engineer self-adaptive robotic systems, we presented RoboMAX, an exemplar repository that aids design and development. Each exemplar has a high-level description of the mission and is classified with extensive meta-information highlighting its context, robot particularities, and adaptation-relevant specifics. The repository and the adopted method allow researchers and practitioners to consult, enrich, and adopt the collected exemplars to establish common ground, and enable approach comparison and cross-evaluation.

The RoboMAX exemplars include both functional and nonfunctional robotic mission requirements. This differs from the “academic examples” of robotic missions used in many research papers, which focus primarily on functional requirements. As such, we envisage that RoboMAX will support the

evaluation of self-adaptation solutions within more realistic scenarios than currently available to the research community.

#### ACKNOWLEDGEMENT

We acknowledge the following contributions for the exemplar definitions: Kompai robotics for the food logistics, PAL Robotics, Clinica Humana, and Badalona Hospital. This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 694277), the Austrian Science Fund (FWF) project M 2778-N “EDENSPACE”, the UKRI project EP/V026747/1 “Trustworthy Autonomous Systems Node in Resilience”, NSERC, and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

#### REFERENCES

- [1] International Federation of Robotics, “World Robotics 2020: Service Robots,” 2020. [Online]. Available: [https://ifr.org/img/worldrobotics/Executive\\_Summary\\_WR\\_2020\\_Service\\_Robots.pdf](https://ifr.org/img/worldrobotics/Executive_Summary_WR_2020_Service_Robots.pdf)
- [2] D. Bozhinoski, D. Di Ruscio, I. Malavolta, P. Pelliccione, and I. Crnkovic, “Safety for mobile robotic systems: A systematic mapping study from a software engineering perspective,” *Journal of Systems and Software*, vol. 151, pp. 150–179, 2019.
- [3] M. Gleirscher, N. Johnson, P. Karachristou, R. Calinescu, J. Law, and J. A. Clark, “Challenges in the safety-security co-assurance of collaborative industrial robots,” 2020. [Online]. Available: <https://arxiv.org/abs/2007.11099>
- [4] L. Lestingi, M. Askarpour, M. M. Bersani, and M. Rossi, “Formal verification of human-robot interaction in healthcare scenarios,” in *Software Engineering and Formal Methods*. Springer, 2020, pp. 303–324.
- [5] F. Vicentini, M. Askarpour, M. G. Rossi, and D. Mandrioli, “Safety assessment of collaborative robotics through automated formal verification,” *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 42–61, 2020.
- [6] M. Luckcuck, M. Farrell, L. A. Dennis, C. Dixon, and M. Fisher, “Formal specification and verification of autonomous robotic systems: A survey,” *ACM Comput. Surv.*, vol. 52, no. 5, 2019.
- [7] S. García, D. Strüber, D. Brugali, T. Berger, and P. Pelliccione, “Robotics software engineering: a perspective from the service robotics domain,” in *European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*, 2020, pp. 593–604.
- [8] C. Menghi, C. Tsigkanos, P. Pelliccione, C. Ghezzi, and T. Berger, “Specification patterns for robotic missions,” *IEEE Transactions on Software Engineering*, 2019.
- [9] “What is a model problem?” [www.hpi.uni-potsdam.de/giese/public/selfadapt/what-is-a-model-problem/](http://www.hpi.uni-potsdam.de/giese/public/selfadapt/what-is-a-model-problem/), 2020.
- [10] “Robotic Mission Adaptation Exemplars,” 2021, <https://doi.org/10.5281/zenodo.4613946>.
- [11] “RoboMAX,” 2021, <https://github.com/Askarpour/RoboMAX>.
- [12] A. Bennaceur, C. McCormick, J. García-Galán, C. Perera, A. Smith, A. Zisman, and B. Nuseibeh, “Feed me, feed me: an exemplar for engineering adaptive software,” in *Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. IEEE/ACM, 2016, pp. 89–95.
- [13] S. Gerasimou, R. Calinescu, S. Shevtsov, and D. Weyns, “UNDERSEA: An exemplar for engineering self-adaptive unmanned underwater vehicles,” in *Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. IEEE/ACM, 2017, pp. 83–89.
- [14] P. H. Maia, L. Vieira, M. Chagas, Y. Yu, A. Zisman, and B. Nuseibeh, “Dragonfly: a tool for simulating self-adaptive drone behaviours,” in *Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. IEEE/ACM, 2019, pp. 107–113.
- [15] G. Moreno, C. Kinneer, A. Pandey, and D. Garlan, “DARTSim: An exemplar for evaluation and comparison of self-adaptation approaches for smart cyber-physical systems,” in *Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. IEEE/ACM, 2019, pp. 181–187.
- [16] M. Kit, I. Gerostathopoulos, T. Bures, P. Hnetyinka, and F. Plasil, “An architecture framework for experimentations with self-adaptive cyber-physical systems,” in *Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. IEEE, 2015, p. 93–96.
- [17] F. Krijt, Z. Jiracek, T. Bures, P. Hnetyinka, and I. Gerostathopoulos, “Intelligent ensembles—a declarative group description language and java framework,” in *Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. IEEE/ACM, 2017, pp. 116–122.
- [18] C. Tsigkanos, L. Nenzi, M. Loreti, M. Garriga, S. Dustdar, and C. Ghezzi, “Inferring analyzable models from trajectories of spatially-distributed Internet of Things,” in *Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, 2019, pp. 100–106.
- [19] S. Mahdavi-Hezavehi, P. Avgeriou, and D. Weyns, “A classification framework of uncertainty in architecture-based self-adaptive systems with multiple quality requirements,” in *Managing Trade-Offs in Adaptable Software Architectures*. Elsevier, 2017, pp. 45–77.
- [20] G. Lucassen, F. Dalpiaz, J. M. E. van der Werf, and S. Brinkkemper, “Improving agile requirements: the quality user story framework and tool,” *Requirements Engineering*, vol. 21, no. 3, pp. 383–403, 2016.
- [21] M. Linjawi and R. K. Moore, “Towards a comprehensive taxonomy for characterizing robots,” in *Annual Conference Towards Autonomous Robotic Systems*. Springer, 2018, pp. 381–392.
- [22] W. E. Walker, P. Harremoës, J. Rotmans, J. P. Van

Der Sluijs, M. B. Van Asselt, P. Janssen, and M. P. Kraye von Krauss, "Defining uncertainty: a conceptual basis for uncertainty management in model-based decision support," *Integrated assessment*, vol. 4, no. 1, pp. 5–17, 2003.

- [23] A. Berns and S. Ghosh, "Dissecting self-\* properties," in *International Conference on Self-Adaptive and Self-Organizing Systems*. IEEE, 2009, pp. 10–19.
- [24] "RoboMAX Exemplar Submission Form," Google-Form/RoboMAX, 2020.
- [25] C. Tsigkanos, T. Kehrer, C. Ghezzi, L. Pasquale, and B. Nuseibeh, "Adding static and dynamic semantics to building information models," in *International Workshop on Software Engineering for Smart Cyber-Physical Systems*. ACM, 2016, pp. 1–7.