Model-Based Testing for System-Level Safety of Autonomous Underwater Robots

Sergio Quijano
Advisor: Mahsa Varshosaz
Department of Computer Science
IT University of Copenhagen
Copenhagen, Denmark
sequ@itu.dk

***Abstract*—For the deployment of autonomous robotic systems in mission- and safety-critical underwater environments, aspects such as reasoning and planning need to be designed to operate in highly dynamic, uncertain environments while assuring a safe and reliable operation. Systems are often deployed without a prior safety assessment or developed with safety analysis as a separate engineering process. In this paper, to tackle these challenges, we propose an initial research vision and plan with the envisioned contributions towards designing an approach for system-wide modeling and Model-Based Testing to support safety assessments of autonomous underwater robots.**

***Index Terms*—Model-Based, testing, autonomous, underwater, robots, safety**

I. Motivation

The deployment of robotic systems in mission- and safety-critical underwater environments raises new design and operation challenges. Aspects such as reasoning and planning need to be designed to operate in highly dynamic, uncertain environments while assuring a safe and reliable operation. Because of the natural complexity of robotic systems [1], and the requirement of a safe interaction in its operational context [2], it is essential to conduct thorough testing of these systems before their deployment. However, the increase of autonomous, sensing-based functionality in robotic systems has made testing increasingly challenging.

Model-Based Testing (MBT) is a technique that may bring proficiency into the testing process. MBT allows for generating and executing test cases more efficiently through test automation. Test cases are derived using a system model that outlines the system's expected behavior. Once test cases are generated, they are executed against an operational version of the system; the goal of the test case execution is to check that the system's implementation complies with the specified behavior [3]. In MBT, the construction of behavioral models and test case generation is done through formal, mathematically-based semantics for the specification and verification of software systems to guarantee the correctness of the systems under test while also providing evidence for their certification.

Furthermore, safety is an important aspect to be considered during the development of autonomous underwater robotic systems. Safety is a system-level property and, therefore, needs to be analyzed on the system level [4]. Traditional safety analysis techniques such as Fault Tree Analysis (FTA) [5], or Failure Mode and Effects Analysis (FMEA) [6] focus only on component failures instead of system-level failures. However, the different system components need to be, to a large degree, checked against their safety requirements to ensure overall system safety. The complexity of underwater robotic systems makes complete system testing cumbersome. Nevertheless, we need to ensure that safety is considered in the testing process.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 956200

II. Research Goals and Methodology

Problem. Existing testing approaches and tools do not incorporate safety analysis. In case they do, they adhere to traditional safety analysis techniques that focus only on component failures instead of system-level failures. Furthermore, the external challenges like highly dynamic, uncertain environments and internal challenges stemming from the autonomous, agent-based, and multi-agent nature of underwater robots are not properly addressed.

Solution. A Model-Based Testing framework for system-level safety assessment of autonomous underwater robots.

Contribution. A novel approach for system-wide safety modeling and Model-Based Testing (MBT) for autonomous underwater robots. Specifically, to conduct efficient system-level safety testing, we expect to provide practitioners with a framework that combines system-wide modeling, MBT, and safety analysis.

Methodology. We designed the following research questions to guide the different phases of our research.

- **RQ1**. What formalisms are suitable for modeling the environment and behavior of autonomous underwater robots?
- **RQ2**. What are important properties to consider in testing for system-level safety assessments of underwater robots?
- **RQ3**. Given the identified formalism for behavioral modeling and safety properties, how can safety assessments be integrated and generalized for Model-Based Testing of underwater robots?

We expect these research questions to provide insightful findings towards designing and implementing the intended framework.

To accomplish the objectives described above, we will research this project using a two-phase, iterative methodology. The results of each phase, namely design and evaluation, will provide feedback for the subsequent phase.

Further, we will break down the design phase into two small, manageable steps. First, we will analyze the open-source Underwater Systems and Technology Laboratory (LSTS) toolchain [12]. The LSTS software toolchain implements a modular, adaptable control architecture supporting networked air and ocean vehicle systems [13]. This modular architecture allows for integrating information about the behavior of different components in an underwater robot and different operation scenarios. Through the analysis of this toolchain, we will investigate what kind of models are suitable to model underwater robots' behavior. Finite State Machines (FSM), Extended FSM (EFSM), and Communicating Extended FSM (CEFSM) [14] are commonly used formalisms in MBT for modeling the system behavior. Initially, we will analyze these formalisms to establish how they fit the existing toolchain and distinguish possible shortcomings.

Additionally, interactions between an autonomous system and its environment can significantly influence its behavior since the robot may be reacting to environmental conditions not considered at design time. Probabilistic models are a popular approach to model dynamic and uncertain environments. We will also investigate Probabilistic Finite State Machines to (partially) model the system's environment and behavioral models. The findings in this step will address **RQ1**.

In the second step, we will consider expert feedback about existing scenarios from our industrial partner OceanScan MST, to categorize the properties that should be considered for the evaluation of safety assurances. Furthermore, it is common to test autonomous systems by randomly selecting test cases from pre-recorded mission data [15]. Nevertheless, a significant amount of data is necessary to achieve sufficiently high coverage of all possible situations. This pre-recorded data can then be classified into scenario types. System-level tests can then be derived by searching for parameter settings in a scenario type that may violate a safety requirement. With the results in this step, it will be possible to address **RQ2**. With the findings in this phase, we will design and develop the necessary algorithms for test case selection and execution by integrating the suitable models and identified properties.

This PhD project is part of a large research and industry collaboration under the Reliable AI for Marine Robotics (REMARO) ETN project. REMARO focuses on reliable AI for autonomous robots performing inspection, surveying, and maintenance tasks underwater. During the evaluation phase, first, we will conduct experiments for model-based test-case generation and execution in a case study for an underwater robot from our industrial partner OceanScan MST. An objective of the evaluations is to refine and generalize the proposed methodology. To this end, and to guarantee the validity of this new methodology, we will perform several experiments on autonomous underwater robotic systems from industrial partners and third-party autonomous systems. The findings during this phase will address **RQ3**.

The proposed doctoral research is in the initial stage of its three-year period. Hence the visions and contributions for the fulfillment of the research objectives are still to be considered partial. In the first year, we will focus on modeling the OceanScan MST autonomous underwater vehicles' behavior as EFSM and identifying the system properties to consider for safety assessment. The findings will set the basis for the design of our MBT framework. In the second year, we will work on the design and implementation of the framework, followed by a first evaluation of the OceanScan MST case study. Finally, in the third year, we will generalize the developed framework and evaluate its use in diverse autonomous underwater systems from our industrial partners.

III. Related work

As explained in I, the model-based testing process begins by building the system models. Such models are extracted from requirements or specification documents. Furthermore, to test the software's safety properties, it is essential to create safety models from those requirements. In [7], the authors provide a safety model which is generated from an automaton model. Then, the test case and test script generation are performed based on the safety model. The proposed approach is validated using an industrial case from the railway domain.

Similarly, Yu et al. [8] propose a framework for generating test cases from a safety model. First, they model the system using finite state machines. Each test requirement is formulated as a temporal logic expression in the Promela language. In addition to these models, Markov chain models are used to describe the states of the system. The test case generation is performed through model checking techniques using the constructed models.

In [9], the authors propose a risk-based testing method using the information from Fault Tree Analysis (FTA). The test cases are created based on the risk given by the FTA. They use the event set notion and transform the event set into state machine as test models.

Gario et al. [10] propose a testing approach based on behavioral and fault models of the system. They build the behavioral model as a communicating EFSM (CEFSM) and the fault model as a fault tree. As the first step, they apply a so-called compatibility transformation using the behavioral model and fault tree and construct a transformed fault tree. Then the transformed fault tree is modeled as a Gate CEFSM (GCEFSM). Finally, they integrate the CEFSM and GCEFSM models into integrated CEFSM (ICEFSM). They developed a tool that uses these models while applying fail-safe testing on a case study from the aerospace domain. In [11], the authors extend the work from [10] to avoid the state space explosion induced by the safety analysis techniques; also, they evaluate the use of Model-Combinatorial based testing aiming to produce safety evidence for certification of safety-critical systems.

REFERENCES

[1] A. Afzal, C. L. Goues, M. Hilton, and C. S. Timperley, "A study on challenges of testing robotic systems," in *2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST)*. IEEE, oct 2020.

[2] A. Przemyslaw, L. Terrence, A. Song; Julie, and Shah, "A survey of methods for safe human-robot interaction," 2017.

[3] M. Utting, A. Pretschner, and B. Legeard, "A taxonomy of model-based testing," 2006.

[4] N. G. Leveson, "Engineering a safer world," 2011.

[5] W. Vesely, F. Goldberg, N. Roberts, and D. Hassl, *Fault Tree Handbook Nureg-0492*, U.S, 1981.

[6] F. Crawley, "Failure modes and effects analysis (fmea) and failure modes, effects and criticality analysis (fmeca)," 2020, pp. 103–109.

[7] G. Yu and Z. Xu, *Model-Based Safety Test Automation of Safety-Critical Software*. IEEE, 12 2010, no. 60674004, pp. 4–6.

[8] G. Yu, Z. Xu, and J. Du, *An Approach for Automated Safety Testing of Safety-Critical Software System Based on Safety Requirements*. IEEE, 5 2009, vol. 3, pp. 166–169.

[9] J. Kloos, T. Hussain, and R. Eschbach, *Risk-Based Testing of Safety-Critical Embedded Systems Driven by Fault Tree Analysis*. IEEE, 3 2011, vol. 2011, pp. 26–33.

[10] A. Gario, A. Andrews, and S. Hagerman, "Fail-safe testing of safety-critical systems: a case study and efficiency analysis," *Software Quality Journal*, vol. 26, no. 1, pp. 3–48, 7 2015.

[11] A. Gannous and A. Andrews, "Integrating safety certification into model-based testing of safety-critical systems." IEEE, 2021.

[12] Lsts group on github. [Online]. Available: https://github.com/LSTS

[13] J. Pinto, P. Dias, R. Martins, J. Fortuna, E. Marques, and J. Sousa, "The LSTS toolchain for networked vehicle systems," in *2013 MTS/IEEE OCEANS - Bergen*. IEEE, jun 2013.

[14] M. Fantinato and M. Jino, "Applying extended finite state machines in software testing of interactive systems," in *DSV-IS 2003*, ser. LNCS, J. Jorge, N. J. Nunes, and J. F. e Cunha, Eds., vol. 2844. Springer, 2003, pp. 34–45.

[15] Z. Saigol, F. Py, K. Rajan, C. McGann, J. Wyatt, and R. Dearden, "Randomized testing for robotic plan execution for autonomous systems," in *2010 IEEE/OES Autonomous Underwater Vehicles*. IEEE, sep 2010.