Model-Driven Development of Formally Verified Human-Robot Interactions

Livia Lestingi[1] [0000−0001−8724−1541]

Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Milan, 20133, Italy, livia.lestingi@polimi.it

Abstract. Service robots will operate in unconstrained environments due to the significant presence of humans. We present a model-driven framework based on formal methods to develop interactive robotic applications designed to handle the uncertainty of human behavior. Users formally model the human-robot interaction scenario, estimate the most likely outcome, and subsequently deploy the application. Collected traces constitute the data pool for an active automata learning algorithm to update the human model based on the accumulated knowledge. We validate the framework on realistic use cases from the healthcare setting.

Keywords: Service Robotics · Model-Driven Engineering · Human-Robot Interaction · Automata Learning.

1 Motivations and Goal

Service robots are growingly widespread in everyday settings, ranging from healthcare to education. In this scenario, robots will operate in highly **unconstrained** environments, where the primary source of uncertainty will be the people requesting services that require interaction. Several recovery strategies are available to tackle unexpected situations in *real-time*, such as collision avoidance and task-reallocation technologies. Nevertheless, this field's lack of well-established **software engineering** practices results in rigid and hardly human-centered applications [7].

This Ph.D. research project proposes a **model-driven** framework to develop service robots applications that can adjust to the **variability** of human behavior. The framework relies on formal modeling and verification techniques to guarantee **robustness**. Covered scenarios feature a **mobile robot** interacting with one (or multiple) **humans** in a closed environment, such as a hospital corridor. Given its human-oriented nature, the framework, although general, is validated on use cases from the healthcare setting.

We present the framework's structure and expected contributions in Section 2; Section 3 briefly reviews related works in literature; finally, Section 4 illustrates the current status of the research plan and future steps needed to complete it.

2 Contribution: Model-Driven Framework

The main elements of the tool-supported framework are shown in Fig. 1. The framework consists of three macro-phases: **PH1**. the **design-time** analysis phase to formally model

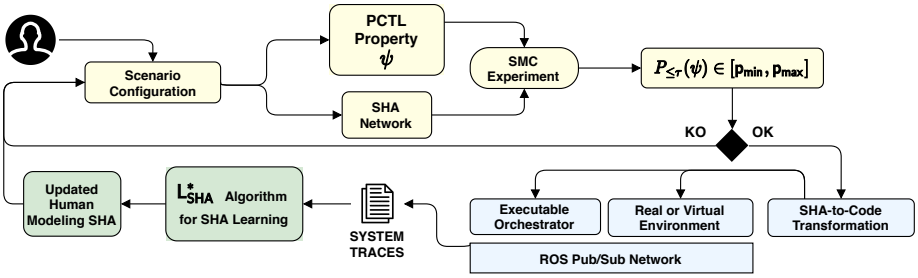

Fig. 1. Overview of the presented framework with its three macro-phases: design-time analysis in yellow, application deployment (or simulation) in blue, and human model adjustment in green.

the human-robot interaction scenario and estimate its probability of success; **PH2**. the **deployment** phase to run (or simulate) the application in a real (or virtual) environment; **PH3**. the **model adjustment** phase exploiting the traces collected during deployment to *learn* a model of human behavior up-to-date with the collected observations.

As the entry point to the framework, users **configure** the scenario under analysis specifying its main parameters: the floor layout, how many humans are involved, their physical characteristics, and what services they are requesting. The formalism of choice is **Stochastic Hybrid Automata** (SHA). In SHA, locations are endowed with differential equations constraining real-valued variables to model systems with complex dynamics [2]. In this specific case, we exploit this feature to model the evolution of human physical **fatigue** [9]. Stochastic behavior in SHA can be realized in two ways. Firstly, edges between two locations can be labeled with a *probability weight* that determines their likelihood of firing. Secondly, discrete variables can be realizations of *probability distributions* whose value is not entirely predictable a priori [6]. The stochastic features in our model capture the unpredictability of human behavior (e.g., due to haphazard choices) and the variability of fatigue rates [15]. The agents (the humans and the robots) and the robot controller (the *orchestrator*) are modeled as separate SHA and interact with each other through *channels* [10], constituting a **network**.

The stochastic nature of the model makes it eligible for **Statistical Model Checking** (SMC) [1]. The goal of SMC experiments is to estimate the probability of delivering all services successfully (captured by Boolean variable scs). Therefore, the property to be verified, expressed in **Probabilistic Computation Tree Logic** (PCTL), is the probability of scs *eventually* ($\diamond$) becoming true within time-bound τ: $P_{\leq\tau}(\diamond\, \mathsf{scs})$. Each SMC experiment yields a probability range $[\mathsf{p}_{\mathsf{min}}, \mathsf{p}_{\mathsf{max}}]$. The framework's user assesses the result and, if it is insufficient, modifies the scenario and repeats the verification experiment; otherwise, the application can move forward to deployment.

It is of paramount importance that the design-time results are maintained at run-time. To this end, for the second phase of the framework, a **transformation** principle maps SHA elements to executable code. The result is an **executable** orchestrator and a fully configured **environment** where the agents will operate. The orchestrator and the agents communicate over a network of ROS publishers and subscriber nodes. Sensors attached or worn by the agents periodically send data to the orchestrator, which examines it

and, if necessary, sends commands to the agents (e.g., *start* or *stop* walking). This deployment approach is compatible both with physical and *virtual* environments. In some cases, such as when the application involves patients in critical physical conditions, it is advisable to corroborate the design-time results by **simulating** the scenario with a realistic physics engine rather than repeatedly deploying it in reality.

At run-time, humans may do actions that their formal model does not cover.[1] These discrepancies impact the outcome of the robotic mission and invalidate the analysis carried out at design-time. The third and final phase of the framework exploits the traces collected through deployment to *learn* an updated model of human behavior. To this end, we introduce an **automata learning** algorithm named L^*_{SHA}, which extends the L^* architecture from Deterministic Finite Automata to SHA learning [3]. The algorithm is general and not exclusively applicable to human-robot interaction scenarios.

L^*_{SHA} relies on a **learner** that progressively refines the SHA hypothesis by asking queries to a **teacher**. While the L^* teacher has perfect knowledge about the system under learning, the L^*_{SHA} teacher relies on collected *samples* requesting new ones when necessary (as in **active** automata learning). The learner asks queries to investigate the system's **state** after a sequence of events occurs, that is to say, which *differential equations* constrain the evolution of physical variables and which *probability distributions* describe their parameters. When the SHA hypothesis is up-to-date with the accumulated knowledge, L^*_{SHA} terminates and returns the new model of human behavior. The user can then reiterate the design-time analysis to get more accurate results.

3 Related Work

Existing works investigate the application of formal methods to human-robot interaction. In some cases, the primary focus is ensuring that collaborative applications meet safety standards [19] or comply with social norms [17]. Some works specifically target the unpredictability of human behavior, for example, by modeling the system as a network of Timed Game Automata [4]. The hereby presented work, instead, adopts probabilistic formalizations of both human behavior and physical fatigue variability. Previous works present learning algorithms for Hybrid (HA) or Probabilistic Automata. Medhat et al. present a framework for HA mining based on clustering [16]. Works focusing on probabilistic systems adopt a frequentist approach [8] or a state merging method [5]. Tappler et al. also propose an extension of L^* to learn Markov Decision Processes based on collected samples [18]. The L^*_{SHA} algorithm developed within this research project is the first one covering both Hybrid and Stochastic features of the formalism.

4 Research Plan Status

Project development began in 2019 and will end in 2022. After an initial state-of-the-art review, the design-time phase has been developed and tested during the first year and presented in [13,11,12]. We will extend the first phase to cover multi-robot scenarios

[1] We remark that this stands also in case of simulation: virtual human agents act upon instructions received by real human users through keyboard.

and a more sophisticated model of human free will by formalizing cognitive models existing in the literature. The deployment module has been developed, tested in simulation, and presented in [14], and we will finalize it by carrying out experiments with real robotic devices. Finally, implementation and validation of L^*_{SHA} algorithm and the model adjustment phase have been completed during the second year. In the future, we plan on extending L^*_{SHA} to also learn weights of probabilistic transitions.

References

1. Agha, G., Palmskog, K.: A survey of statistical model checking. TOMACS **28**(1), 1–39 (2018)
2. Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T.A., Ho, P.H., Nicollin, X., Olivero, A., Sifakis, J., Yovine, S.: The algorithmic analysis of hybrid systems. Theoretical Computer Science **138**(1), 3–34 (1995)
3. Angluin, D.: Learning regular sets from queries and counterexamples. Information and computation **75**(2), 87–106 (1987)
4. Bersani, M.M., Soldo, M., Menghi, C., Pelliccione, P., Rossi, M.: Pursue-from specification of robotic environments to synthesis of controllers. Formal Aspects of Computing (2020)
5. Carrasco, R.C., Oncina, J.: Learning stochastic regular grammars by means of a state merging method. In: Intl. Colloquium on Grammatical Inference. pp. 139–152. Springer (1994)
6. David, A., Larsen, K.G., Legay, A., Mikučionis, M., Poulsen, D.B., Van Vliet, J., Wang, Z.: Statistical model checking for networks of priced timed automata. In: Intl. Conf. on Formal Modeling and Analisys of Timed Systems. pp. 80–96. Springer (2011)
7. García, S., Strüber, D., Brugali, D., Berger, T., Pelliccione, P.: Robotics software engineering: A perspective from the service robotics domain. In: ESEC/FSE. pp. 593–604 (2020)
8. Ghezzi, C., Pezzè, M., Sama, M., Tamburrelli, G.: Mining behavior models from user-intensive web applications. In: Intl. Conf. on Software Engineering. pp. 277–287 (2014)
9. Konz, S.: Work/rest: Part ii-the scientific basis (knowledge base) for the guide 1. EGPS **1**(401), 38 (2000)
10. Larsen, K.G., Pettersson, P., Yi, W.: UPPAAL in a nutshell. Int. J. on Softw. Tools for Tech. Transf. **1**(1-2), 134–152 (1997)
11. Lestingi, L., Askarpour, M., Bersani, M.M., Rossi, M.: Formal verification of human-robot interaction in healthcare scenarios. In: SEFM. pp. 303–324. Springer (2020)
12. Lestingi, L., Askarpour, M., Bersani, M.M., Rossi, M.: A model-driven approach for the formal analysis of human-robot interaction scenarios. In: IEEE SMC. pp. 1907–1914 (2020)
13. Lestingi, L., Askarpour, M., Bersani, M.M., Rossi, M.: Statistical model checking of human-robot interaction scenarios. arXiv preprint arXiv:2007.11738 (2020)
14. Lestingi, L., Askarpour, M., Bersani, M.M., Rossi, M.: A deployment framework for formally verified human-robot interactions. To appear in IEEE Access (2021)
15. Liu, B., Ma, L., Chen, C., Zhang, Z.: Experimental validation of a subject-specific maximum endurance time model. Ergonomics **61**(6), 806–817 (2018)
16. Medhat, R., Ramesh, S., Bonakdarpour, B., Fischmeister, S.: A framework for mining hybrid automata from input/output traces. In: EMSOFT. pp. 177–186. IEEE (2015)
17. Porfirio, D., Sauppé, A., Albarghouthi, A., Mutlu, B.: Authoring and verifying human-robot interactions. In: Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology. pp. 75–86 (2018)
18. Tappler, M., Aichernig, B.K., Bacci, G., Eichlseder, M., Larsen, K.G.: L*-based learning of markov decision processes. In: FM. pp. 651–669. Springer (2019)
19. Vicentini, F., Askarpour, M., Rossi, M.G., Mandrioli, D.: Safety assessment of collaborative robotics through automated formal verification. IEEE Transactions on Robotics (2019)