RoboTool: Modelling and Verification with RoboChart*

Alvaro Miyazawa† Ana Cavalcanti† Simon Foster†
Wei Li‡ Pedro Ribeiro† Jon Timmis‡ Jim Woodcock†

In previous work [3, 4, 7], we have proposed a domain-specific modelling language for robotic applications. It is called RoboChart, and its core modelling construct is a version of UML state machines, enriched with a precise action language, support for synchronous communication and abstraction, as well as facilities for modelling time and probabilities. RoboChart has multiple related semantics: an untimed semantics based on CSP [8], a timed semantics based on timed-CSP [8], and a probabilistic semantics based on Reactive Modules [2].

We propose to demonstrate the application of RoboChart and its associated tool, RoboTool, for the verification and validation of robotic applications. RoboTool supports the creation, editing, and validation of RoboChart models. Distinctively, as shown in Figure 1, it also automatically generates the mathematical models of RoboChart diagrams, as well reactive simulations for use with ARGoS [6]. Currently, RoboTool supports the following verification and simulation tools: FDR (untimed and timed), Prism, Storm, Isabelle/HOL and ARGoS. In our demonstration, we will consider a few small examples for illustration, and the larger example of a transporter. It is part of a swarm and cooperates with other identical robots to push an object to a target location.

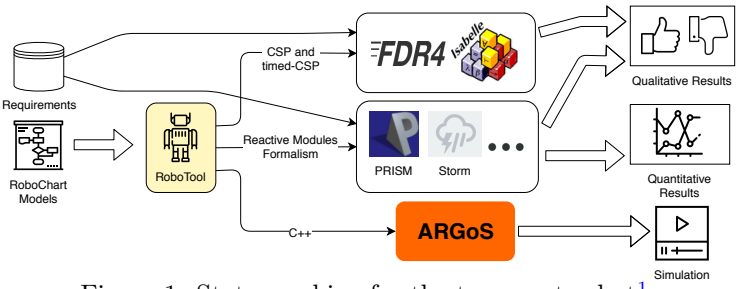

Figure 1: State machine for the transport robot[1].

We have identified a number of requirements that our transporter model must satisfy. These properties fall into three different groups: (1) refinement, (2) timed refinement, and (3) probabilistic. Here, we demonstrate the verification of properties of our example using six different techniques.

Automatic timed and untimed verification via model checking. Untimed properties can be automatically verified using our untimed semantics and the model checker FDR [1]. While a positive result is desirable, a negative result is equally interesting as it provides the user with a counterexample, which may

*This work is funded by the EPSRC grant EP/M025756/1. No new primary data created.
†Department of Computer Science, University of York.
‡Department of Electronic Engineering, University of York.
[1]Icons made by monkik, Freepik and Smashicons from www.flaticon.com.

lead to a revision of the model or of the requirement. Besides the verification of untimed properties, due to our timed semantics and the support for timed-CSP in FDR, we can also verify timed properties of our model.

Semi-automatic refinement verification via theorem proving. While model checking is performed automatically, it can be limited in the types of systems that it can check. While RoboChart allows the modelling of systems with infinite types, for example, their semantics is parameterised by finite instantiations of these types. This is a limitation of FDR, which is not present when using theorem provers. Model checkers that support verification of infinite state systems through integrations with SMT solvers and automatic theorem provers exist, but they are still limited in their verification power and modelling features.

The underlying theories developed in the theorem prover Isabelle/HOL [5] to support verification of RoboChart models can take advantage of both the structure of the models, and the type of properties to provide a high level of automation. In our demonstration, we use theorem proving to automatically check for deadlock freedom for a model with infinite data types, as well as a more interesting property with which FDR cannot cope due to state space explosion, even with finite approximations of data types.

Automatic verification via probabilistic and statistical model checking. Our third semantic model of RoboChart takes into account probabilistic choices. We consider the analysis of a collection of random walk algorithms relevant for our transporter and other applications like chemical detection. We compare the algorithms in terms of their use of battery, for instance. The specialisation afforded by RoboChart combined with statistical model checking enables analysis of models of significant realistic sizes.

Verification by simulation using ARGoS. While formal verification is important, roboticists make extensive use of simulation to explore the design and consider alternative platforms and environments. So, we complement the formal techniques used to analyse our examples with the automatic generation of simulations for the ARGoS simulator. Simulation provides the means to validate our model in varied and complex environments under semi-realistic conditions, and to produce clear and intuitive evidence of the suitability of the models. This is possible before any proof effort is invested, for instance.

References

[1] T. Gibson-Robinson, P. Armstrong, A. Boulgakov, and A. W. Roscoe. FDR3: A Modern Refinement Checker for CSP. In *TACAS'14*, pages 187–201, 2014.

[2] M. Kwiatkowska, G. Norman, and D. Parker. Probabilistic symbolic model checking with PRISM: a hybrid approach. *Int. J. Softw. Tools Te.*, 6(2):128–142, 2004.

[3] W. Li, A. Miyazawa, P. Ribeiro, A. L. C. Cavalcanti, J. Woodcock, and J. Timmis. From Formalised State Machines to Implementations of Robotic Controllers. In *DARS'16*, pages 517–529. Springer, 2018.

[4] A. Miyazawa, P. Ribeiro, W. Li, A. L. C. Cavalcanti, and J. Timmis. Automatic Property Checking of Robotic Applications. In *IROS'17*, pages 3869–3876, 2017.

[5] T. Nipkow, M. Wenzel, and L. C. Paulson. *Isabelle/HOL: a proof assistant for higher-order logic*. Springer, 2002.

[6] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, M. Birattari, L. M. Gambardella, and M. Dorigo. ARGoS: a Modular, Parallel, Multi-Engine Simulator for Multi-Robot Systems. *Swarm Intelligence*, 6(4):271–295, 2012.

[7] P. Ribeiro, A. Miyazawa, W. Li, A. L. C. Cavalcanti, and J. Timmis. Modelling and Verification of Timed Robotic Controllers. In *IFM'17*, pages 18–33. Springer, 2017.

[8] A. W. Roscoe. *The Theory and Practice of Concurrency*. Prentice-Hall Series in Computer Science. Prentice-Hall, 1998.