Formal Development of Supervision for Autonomous Systems*

Michael Butler, Toby Wilkinson
University of Southampton, UK

May 17, 2016

1 Abstract

We report on an on-going project aiming at a correct-by-construction method for autonomous systems that allows the discovery of design defects much earlier in the development flow than current practice. We use the Event-B formalism for precise capture of system requirements as formal models. Event-B supports a refinement based approach which allows features to be added incrementally and allows abstract models to be formally linked to software implementions. The Rodin tool for Event-B uses a range of back-end theorem provers to verify proof conditions assoicated with models to verify invariants preservation and refinement correctness of models.

Our project is focusing on methods and tools addressing safe physical movement of UAVs covering a range of requirements such as safe separation, terrain avoidance, segregated airspace and no-fly zones. This involves developing theories in continuous mathematics using the theory extension capabilities of Rodin. Results to date demonstrated the benefits of a refinement approach to development of a route validation both in terms of precision of requirements specification and management of complexity of the design. We made extensive use of the theory extension feature of Rodin to build and use physical theories of continuous paths in 3-D Euclidean space with safety invariants formulated for safe separation. The use of domain theories is leading to improved reusability of modelling and proof. We are currently working on modelling and reasoning about terrain avoidance, segregated airpsace and no-fly zone maintenance.

The architecture we are exploring is to have a separate route validator function that supervises the output of a nondeterministic intelligent planning func-

*Supported by the ASUR Programme project 1014_C6_PH1_104. This document is an overview of MOD sponsored research and is released to inform projects that include safety-critical or safety-related software. The information contained in this document should not be interpreted as representing the views of the MOD, nor should it be assumed that it reflects any current or future MOD policy. The information cannot supersede any statutory or contractual requirements or liabilities and is offered without prejudice or commitment.

tion that generates the routes. The idea being that the route validator can be made simpler than the route generator, and therefore its correctness more easily shown. Formal development is focused on the route validator.

We are refining the validator models towards code where, for example, continuous paths are reified to sequences of weypoints. In refining the models towards code level, we identified patterns of design that would benefit from automated support in order to increase productivity. Verification of physical properties is performed using mathematical reals whereas the implementation uses floating point arithmetic presenting a verification challenge. We are currently exploring the use of backend provers such as Isabelle to reason about continuous functions.