RoboCalc: A Calculus for Software Engineering of Mobile and Autonomous Robots

Overview
Modelling and development of robotic and autonomous systems is based on standard state machines, without formal semantics, to specify the robot controller only. This guides the development of a simulation, but no rigorous connection is established. Moreover, implementation requires ad-hoc adjustments to cater for the reality gap between simulation and the robot hardware and the environment.

In this scenario, evidence of safety is scarce; full verification is beyond the state of the art.

In RoboCalc we are developing a framework for integrating the modelling, simulation, and programming. We are adopting notations akin to those already in use by the robotics community, but enriched with facilities to specify timed and probabilistic behaviours and the environment. For simulation we are developing a language that is tool independent. Our framework will ensure that models and simulations are consistent by using sound verification techniques.

Modelling
- Eclipse plug-in supporting both graphical and textual specification notation.
- State-machine based notation with facilities to specify reactive behaviour and data operations, as well as time properties and the environment.

Analysis
- Model-checking using FDR3 with CSP semantics.
- Model-checking using UPPAAL with timed automata semantics for analysis of properties concerning continuous time.
- Theorem proving using Isabelle/UTP

Simulation
- High-level domain specific language
- Platform independent
- Supports automatic code generation for simulation and deployment

RoboSim
- Simulation
- Deployment
- CSP
- UPPAAL
- ...

RoboCheck
- Model-checking using UPPAAL
- Model-checking using FDR3

RoboChart
- RoboTool graphical environment
- Wheeled robot example

RoboCalc: A Calculus for Software Engineering of Mobile and Autonomous Robots
Overview
Modelling and development of robotic and autonomous systems is based on standard state machines, without formal semantics, to specify the robot controller only. This guides the development of a simulation, but no rigorous connection is established. Moreover, implementation requires ad-hoc adjustments to cater for the reality gap between simulation and the robot hardware and the environment.

In this scenario, evidence of safety is scarce; full verification is beyond the state of the art.

In RoboCalc we are developing a framework for integrating the modelling, simulation, and programming. We are adopting notations akin to those already in use by the robotics community, but enriched with facilities to specify timed and probabilistic behaviours and the environment. For simulation we are developing a language that is tool independent. Our framework will ensure that models and simulations are consistent by using sound verification techniques.

Modelling
- Eclipse plug-in supporting both graphical and textual specification notation.
- State-machine based notation with facilities to specify reactive behaviour and data operations, as well as time properties and the environment.

Analysis
- Model-checking using FDR3 with CSP semantics.
- Model-checking using UPPAAL with timed automata semantics for analysis of properties concerning continuous time.
- Theorem proving using Isabelle/UTP

Simulation
- High-level domain specific language
- Platform independent
- Supports automatic code generation for simulation and deployment

RoboSim
- Simulation
- Deployment
- CSP
- UPPAAL
- ...

RoboCheck
- Model-checking using UPPAAL
- Model-checking using FDR3

RoboChart
- RoboTool graphical environment
- Wheeled robot example

RoboCalc: A Calculus for Software Engineering of Mobile and Autonomous Robots
Overview
Modelling and development of robotic and autonomous systems is based on standard state machines, without formal semantics, to specify the robot controller only. This guides the development of a simulation, but no rigorous connection is established. Moreover, implementation requires ad-hoc adjustments to cater for the reality gap between simulation and the robot hardware and the environment.

In this scenario, evidence of safety is scarce; full verification is beyond the state of the art.

In RoboCalc we are developing a framework for integrating the modelling, simulation, and programming. We are adopting notations akin to those already in use by the robotics community, but enriched with facilities to specify timed and probabilistic behaviours and the environment. For simulation we are developing a language that is tool independent. Our framework will ensure that models and simulations are consistent by using sound verification techniques.

Modelling
- Eclipse plug-in supporting both graphical and textual specification notation.
- State-machine based notation with facilities to specify reactive behaviour and data operations, as well as time properties and the environment.

Analysis
- Model-checking using FDR3 with CSP semantics.
- Model-checking using UPPAAL with timed automata semantics for analysis of properties concerning continuous time.
- Theorem proving using Isabelle/UTP

Simulation
- High-level domain specific language
- Platform independent
- Supports automatic code generation for simulation and deployment

RoboSim
- Simulation
- Deployment
- CSP
- UPPAAL
- ...

RoboCheck
- Model-checking using UPPAAL
- Model-checking using FDR3

RoboChart
- RoboTool graphical environment
- Wheeled robot example