

# WASP Project Course 2021

## #9: Software Technology for Reliable Autonomous Systems

### Background

A big challenge in autonomous cars, drones and robots is to secure safety and reliability. The main goal of this project is to bring relevant software technology tools into software for autonomous systems that have the possibility to improve safety and reliability. As a result we would expect lower time-to-failure and higher up-time.

This goal will be achieved through two sub-goals:

1. to explore and demonstrate the use of the advanced robotics software SkiROS for programming of autonomous systems.
2. to analyse the connections between SkiROS components, including cross-cutting concerns (logging, replay, debugging, . . . ) and expose them in a domain model. This domain model will then be used to design a domain-specific language that can improve reliability and safety.

#### What is SkiROS?

SkiROS stands for “Skills for ROS”. It allows to close the action perception loop of autonomous systems by encapsulating functions for perception, learning and actuation as robot skills. SkiROS offers a complete infrastructure to run skills, incl. a world-model, suitable APIs, Behavioral-Tree infrastructure, and a STRIPS-like planner to identify the right sequence of skills for a given task goal. SkiROS has a TRL-6 level as evaluated at the car-production site in Metz, France, of the PSA-Group.

Due to its structure, SkiROS is able to foster collaboration within a lab and across research sites between students and post-docs.

SkiROS has received funding as the very first WALP-project. During the WALP assessment, SkiROS has been identified as highly relevant tool that speeds up programming of robots in the manufacturing context, in particular for manufacturing SMEs. SkiROS exists as open-source software that is accessible worldwide.

This is an interdisciplinary project: Robotics students and software technology students are meant to support each other with their expertise. In order to do this project software technology students are not required to have prior knowledge of robotics software and do not have to program any robots.

**Constraints:** For simplicity we suggest keeping sub-goal 1 in Lund for this WASP project. The DSL-related sub-goal 2 is **not** constrained to LTH and can be joined across campuses.

### Participants

**Industrial partner:** RiACT A/S: Startup company funded by KAW and WALP

**Industrial supervisor:** Francesco Rovida, f.rovida@riact.eu

**Academic supervisor:**

- **Subgoal 1:** Volker Krueger, volker.krueger@cs.lth.se, LU
- **Subgoal 2:** Christoph Reichenbach, christoph.reichenbach@cs.lth.se, LU

**Coordinating WARA representative:** Christoph Reichenbach, WARA-SW

**Suggested WASP PhD students:** Matthias Mayr, Faseeh Ahmad, Momina Rizwan, Alexander Dürr, Hampus Åström, Johan Oxenstierna, Julian Salt Ducaju

## Challenges to investigate

The students will design a domain-specific language for facilitating the integration of heterogeneous robotics components both with each other and with cross-cutting concerns like logging, mocking, replay, debugging, profiling, robustness, and safety. Specifically, the language will be designed to enable high-level static and dynamic analysis techniques, such as type analysis or delta debugging", a technique that automatically simplifies input that produces erroneous behaviour to pinpoint faulty code or faulty interactions between components.

To design this language, the students will work with robotics engineers and with the existing SkiROS documentation to first derive a domain model, and to identify existing key SkiROS concepts, components, and concerns, as well as their interactions. They will then look for extensions to this domain model to improve reliability and safety, and design a DSL together with a suitable toolchain that materialises their insights. They will build a (partial) prototype implementation, e.g., as an internal DSL in Python, Racket, or Scala, and evaluate the DSL by encoding example robotics programs, especially the demonstrator from Sub-Goal 1, in this language and identifying strengths and weaknesses in their design.

To ensure that the students or a follow-up project guided by the students can systematically evaluate the DSL and its tools (type checker, delta debugger, . . . ), the students working on Sub-Goal 1 will make systematic use of standard software engineering practices, including but not limited to revision control, unit testing, continuous integration and issue tracking.

### ***Sub-Goal 1: Demonstrate SkiROS for use in WASP***

To demonstrate the use and advantages of SkiROS, we will

1. Demonstrate how WASP PhD students can integrate their recent publications into a joint system.
2. Demonstrate the resulting system on multiple robots, incl. KUKA iiwa, YuMi and a mobile robotic MIR platform with a UR5e robotic arm.
3. Ideally demonstrate the complete system within the WARA-Robotics to verify and validate the complete system

*This goal requires robotics expertise.*

## ***Sub-Goal 2: Domain-specific languages for SkiROS for Reliability and Safety***

To design a domain-specific language that improves Reliability and Safety, we will

1. Extract a domain model that exposes the key concepts and cross-cutting concerns in SkiROS
2. Design a domain-specific language that exposes these domain concepts as appropriate and enables high-level static and dynamic program analysis (and/or transformation) techniques
3. Demonstrate the language's expressiveness through example SkiROS programs based on work in Sub-Goal 1 and/or software tools such as static checkers or delta-debuggers.

*This goal requires software technology expertise.*

## **Resources**

At LTH:

- **Hardware:** Dual-arm Kuka iiwa, YuMi-Robot, mobile manipulator with MIR-mobile platform and a UR5e robotic arm, multiple cameras and LiDAR, multiple grippers.
- **Revision Control:** Shared git repository and issue tracker (internal), if needed
- **Continuous Integration:** tbd
- **Communication:** Slack (unregistered) or Matrix/Element (free)

## **Deliverables**

- Full documentation on how to run and install SkiROS, and best practices on how to implement skills: This report will summarize how to use SkiROS, how to install it and how to get a fast start on a number of relevant robots. This deliverable will be mostly provided by the robotics students, with the support of the software technology students.
- A domain model for SkiROS: This report will highlight cross-cutting concerns such as movement and safety monitoring, possibly simplified to explicitly exclude certain concerns (depending on time constraints). This deliverable will be mostly provided by the software technology students, with the support of the robotics students.
- A preliminary DSL sketch for SkiROS (if time permits): This report discusses interactions between the cross-cutting concerns and the impact of the concerns and of their interactions on language design, i.e., feasibility of static and dynamic program analysis (this is the one bit where this clearly intersects with the WARA-SW mission).

## **References**

## **Keywords**

kw1, kw2