Bachelor/Master Thesis: Hybrid Active Objects with ABS and KeYmaera

Software Engineering Group – Eduard Kamburjan
Contact: kamburjan@cs.tu-darmstadt.de

TECHNISCHE UNIVERSITÄT DARMSTADT

Context

The Abstract Behavioral Specification language (ABS) [1] is a language to model, specify and verify distributed systems. It has been applied to numerous domains, including biology, railway operations and cloud services. While its design proved to be very robust for modeling and analyzing in multiple domains, its lack of continuous primitives complicate its application to cyber-physical systems.

The KeYmaera tool [2] is the leading deductive verification tool for hybrid systems with application to real world system in aviation and railways. To enable hybrid modeling, the used language incorporates ordinary differential equations (ODEs) for continuous behavior. However, this language is not designed for natural modeling, which leads to low comprehensibility of the model.

To combine these approaches, this thesis is supposed to extend ABS with primitives for differential equations, derived from the ones used in KeYmaera and enable the compiled code to simulate the described physical properties.

The focus will be on integrating the available explicit time annotations of ABS in the framework of differential equations, the required data structures to handle ODEs and the interface between DEQs and discrete steps modeled by the established infrastructure of ABS, as solving the differential equations will be handled externally.

Possible model of a bouncing ball

```
class Ball(Rat damping)
{
  physical{
    Rat a = −9.82; a' = 0;
    Rat v = 0;   v' = a;
    Rat x = 10; x' = v;
  }
                              Unit run(){
                                await x == 0\&\& v < 0;
                                v = −v*damping;
                                this!run();
                              }
}
```

Thesis

The student is expected to implement a prototypical extension of ABS with operations for differential equations, where each object is enriched with continuous behavior.

The student must have *demonstrable* experience with compiler construction.

Approximate Work Distribution

Analysis
Programming
Literature

References

[1] R. Hähnle, "The abstract behavioral specification language: A tutorial introduction," in *Proc. Formal Methods for Component-Based Systems (FMCO)*, 2012.

[2] A. Platzer and J.-D. Quesel, *KeYmaera: A Hybrid Theorem Prover for Hybrid Systems (System Description)*, pp. 171–178. Springer, 2008.