

The SCEL language comes with solid semantics foundations laying the basis for formal reasoning. MISSCEL, a rewriting-logic-based implementation of the SCEL operational semantics is a first step in this direction. MISSCEL is written in Maude, which allows to execute rewrite theories – what we obtain is an executable operational semantics for SCEL, that is, an interpreter. Given a SCEL specification, thanks to MISSCEL it is possible to use the rich Maude toolset to perform (i) automatic state-space generation, (ii) qualitative analysis via Maude invariant and LTL model checkers, (iii) debugging via probabilistic simulations and animations generation, (iv) statistical quantitative analysis via the recently proposed MultiVeStA [SV] statistical analyser that extends PVeStA.

A further advantage of MISSCEL is that SCEL specifications can now be intertwined with raw Maude code, exploiting its great expressiveness. This allows to obtain cleaner specifications in which SCEL is used to model behaviours, aggregations, and knowledge manipulation, leaving scenario-specific details like environment sensing abstractions or robot movements to Maude.

Further Information:

- [User Guide](#)
- [Download](#)