The Real Meaning of State Machine Diagrams

Supervisors: Arend Rensink, Marcus Gerhold

State Machine Diagrams (SMDs) are a well-known, powerful graphical modelling language for describing the expected behaviour of large, complex systems. Its power lies in the availability of multiple features for the compact representation of frequently occurring behavioural patterns, such as concurrency, non-determinism, multiple exit points, triggers and synchronisation. However, at the same time, this wealth of features makes it hard to analyze all their possible combinations and to be sure that the expected behaviour is under all circumstances well-understood.

Recent work on the particular variant of SMDs incorporated in SysML has resulted in a transformation pipeline that simplifies SMDs by transforming many of their special features into more basic ones, essentially gaining simplicity at the cost of expressiveness. Moreover, after this transformation, the simplified SMDs are converted into finite state machines (FSMs), effectively providing a translational semantics of SMDs in terms of a well-defined formal model.

Though quite successful, this work has left some open questions, to which this MSc assignment aims to provide a solution:

- The transformations implemented in this pipeline are partially but not wholly formally defined;
- The implementation itself is imperative rather than declarative, making it harder to establish that it actually follows the definition;
- There is no direct operational semantics of the SMDs, so that there is no way of knowing whether the simplification of SMDs and their transformation into FSMs is correct

This is what this MSc assignment aims to improve upon. Concretely, the outcome of the project should consists of:

1. A complete definition of the SMD transformations described above;
2. Hand in hand with this, a corresponding (declarative) implementation;
3. A direct operational semantics of the SMDs that is compatible with the transformations;
4. A proof that the SMD transformations and the SMD-to-FSM transformation are semantics-preserving

To carry out this steps, the project is expected to use Graph Transformation, as implemented in the tool GROOVE (see groove.cs.utwente.nl).

The assignment is modular in that it is not clear how much work will go into these steps. At minimum, we expect 1+2 to be feasible; the complexity of the other steps can be investigated as part of the project.