**Motivation**

*Interface Automata* are a formalism for modelling behaviour of components through combination of required and provided interfaces. Some of their notable properties are:

+ well-established formalism,
+ straightforward use in model checking,
+ composition,
+ refinement,
+ orientation toward open systems,
  - lack of support for unbounded threading.

Component behaviour with potentially unbounded degree of parallelism tends to result in an infinite model when subjected to explicit-state model checking.

The goal of the thesis was to propose a method based on interface automata allowing one to capture behaviour of common components and to

▶ allow their replication to an arbitrary degree of parallelism,
▶ allow compositions of the resulting models with compatible counterparts.

**Solution**

▶ Relaxed *Interface Automaton* definition for intermediate results
▶ Annotations dividing model into non-replicable, replicable, and critical sections
▶ Replication operation
▶ Revised composition operation ensuring compliance with standard *Interface Automaton* definition

**Properties**

▶ Finiteness of models
▶ Repeatability of operations
▶ Selective replication
▶ Basic synchronisation awareness
▶ Running composition avoiding message ambiguity and unnecessary state space growth

**References**