

# SSTraGen A (semi-)Automated Extraction Method for Behavioral Models

A MasCot Software Restructurings Project (17933)

P.H.M. van Spaendonck, Msc.

In collaboration with:



## Behavioral Models are Great!

Models can be used for:

- The automated generation of tests via MBT;
- Software verification using safety/liveness requirements;
- Detecting behavioral regression via equivalence checking;
- Visualization of behavior.

## Don't learn models! Use the code instead

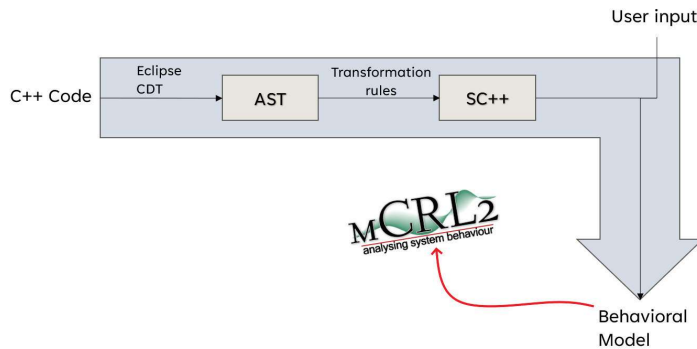


Figure 1: Eclipse CDT is used for parsing C++; the AST is then transformed according to our rules into an unbounded model; with user input, the model is bounded and ready.

Using mCRL2[1] as modeling formalism, gives us access to efficient and reliable model-based techniques for model generation, reduction, verification, and manipulation.

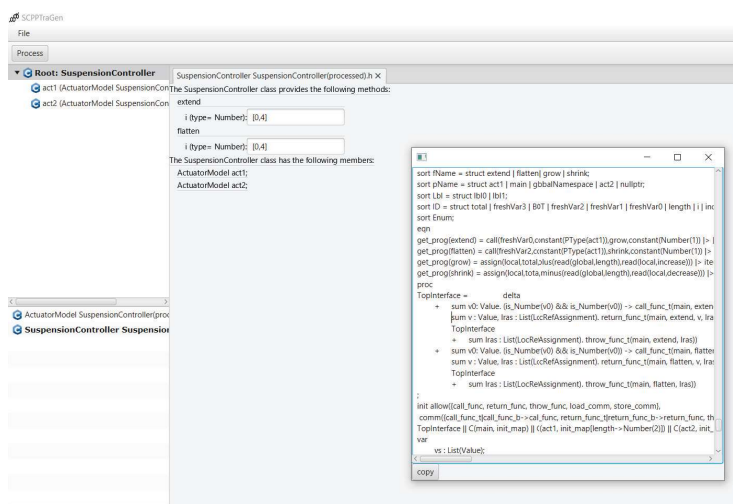


Figure 2: The tool provides a GUI for manipulation of the intermediate SC++ model; allowing the user to submit bounds, abstract away components, and generate models.

## Handling Scalability

One potential technique aimed towards avoiding state-space explosion and thus improving the scalability of the tool:

**Component-wise generation and reduction** Generating models for individual sub-components first and then reducing these models, reduces potential exponential growth caused by parallel composition.

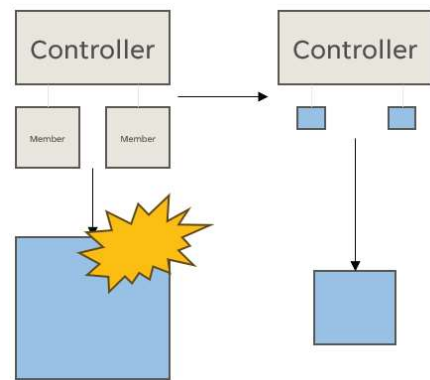


Figure 3: Using reduced models of sub-components, can allow circumvention of a potential state-space explosion.

## Extensive Programming Feature Support

SSTraGen supports most modern OOP features, including:

- Pass-by-reference,
- higher order functions, e.g. lambda functions,
- recursion,
- alias'ing,
- inner classes

## Case Studies

So far, we have successfully used the tool on several components within Philips, including:

- A component of 300 LOC, that was used to visualize an invariant over its instances.
- A component of 600 LOC, that we have shown to fit within a preexisting UML diagram.

## References

- [1] Olav Bunte, Jan Friso Groote, Jeroen J. A. Keiren, Maurice Laveaux, Thomas Neele, Erik P. de Vink, Wieger Wesselink, Anton Wijs, and Tim A. C. Willemse. The mcr2 toolset for analysing concurrent systems. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 21–39, Cham, 2019. Springer International Publishing.