Translating Erlang state machines to UML using triple graph $grammars^1$

Dániel Lukács, Melinda Tóth

Department of Programming Languages and Compilers, Eötvös Loránd University, Budapest {dlukacs,tothmelinda}@caesar.elte.hu

In this paper, we present a method that transforms event-driven Erlang state machines into highlevel state machine models represented in UML. We formalized the transformation system as a triple graph grammar, a special case of graph rewriting. We argue in this paper that using this well-defined formal procedure opens up the way for verifying the transformation system, synchronizing code and formal documentation, and executing state machine models among many other possible use cases. We also provide an example transformation system and demonstrate its application in action on a small Erlang state machine. We also present our evaluation of our full system implementation tested on real world Erlang state machines.

References

- [1] Andy Schürr: Specification of graph translators with triple graph grammars 20th International Workshop, WG '94 Herrsching, Germany, June 16–18, 1994 Proceedings. pp 151-163 doi:10.1007/3-540-59071-4_45 1995.
- Melinda Tóth & István Bozó: Static Analysis of Complex Software Systems Implemented in Erlang,
 Central European Functional Programming Summer School Fourth Summer School, CEFP 2011, Revisited Selected Lectures, Lecture Notes in Computer Science (LNCS), Vol. 7241, pages 451-514, Springer-Verlag, ISSN: 0302-9743, 2012.
- [3] Doron Drusinsky: Modeling and Verification Using UML Statecharts: A Working Guide to Reactive System Design, Runtime Monitoring and Execution-based Model Checking Newnes.
 ISBN: 0750679492 2006.

 $^{^1{\}rm The}$ project has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002).