Applying model checking to critical PLC applications: An ITER case study

B. Fernández, D. Darvas, E. Blanco, CERN, Geneva, Switzerland
Gy. Sallai, BME, Budapest, Hungary
I. Prieto, IBERINCO, Madrid, Spain
G. Lee, Mobis Co. Ltd., Seoul, South Korea
B. Avinashkrishna, Y. Gaikwad, S. Sreekuttan, Tata Consultancy Services, Pune, India
R. Pedica, Vitrociset s.p.a, Rome, Italy

GOAL: Verification and better understanding of the PLC program implementing the HIIOC protocol

Model checking
Model checking is a formal verification method that checks the satisfaction of a formal requirement on a formal model with mathematical precision for all possible executions. In case a violation of a requirement is found, a counterexample can be provided that shows a trace leading to the violation.

PLCVerif's verification workflow

Pattern-based verification
- Pattern-based requirement: Fixed English sentence with placeholders to be filled by the verifier
- nuxmv: State-of-the-art symbolic model checker tool

Assertion-based verification
- Verification assertion: Logic expression in the code that must always be satisfied
- CBMC: Bounded model checker to check assertion violations in C code
- Fast verification
  - Assertions can only represent simple requirements
  - Bounded model checking ensures correctness only for certain length

Implementation
WinCC OA
SCADA
HIOC
S7-400
PLCs

PLCVerif
- Model checking solution for PLC programs
- Hides the formal details from the users
- Integrates multiple model checking engines
- Developed at CERN

Outcome
- Formal proof of correctness
  - Ongoing work
  - Formalising and checking all important requirements is an ongoing work
  - Difficult to ensure completely: All tools in the toolchain must be verified

- Improved understanding
  - Via counterexamples
    - A counterexample can show a witness of an incorrect behaviour
    - Similarly, counterexamples can be used to provide examples (traces) of any behaviour
    - Such trace may reveal peculiar, unexpected functionality
  - Via requirement formalisation
    - Model checking requires formal requirements
    - Removing all ambiguity from informal specifications is difficult and often reveals interesting corner cases
    - Needs collaboration of specifiers, developers and verifiers

You can find the paper and more information at
http://cern.ch/plcverif
http://iter.org

We thank the ITER interlock team for their support of this work.