CrAVES : Credible Autocoding and Verification of Embedded Software


Abstract:
The CrAVES project seeks to lay down intellectual foundations for credible autocoding of embedded systems, by which model-level control system specifications that satisfy given open-loop and closed-loop properties are automatically transformed into source code guaranteed to satisfy the same properties. The goal is that the correctness of these codes can be easily and independently verified by dedicated proof checking systems. During the autocoding process, the properties of control system specifications are transformed into proven assertions explicitly written in the resulting source code. Thus CrAVES aims at transforming the extensive safety and reliability analyses conducted by control system engineers, such as those based on Lyapunov theory, into rigorous, embedded analyses of the corresponding software implementations. CrAVES comes as a useful complement to current static software analysis methods, which it leverages to develop independent verification systems. In this poster presentation we will illustrate various results of the CrAVES project:

Credible compilation: We have developed a toolchain targeting embedded systems developed in Simulink. In a first stage, the toolchain takes as input Simulink models enriched with safety requirements and control semantics such as stability proofs and compiles them either directly to C or to Lustre programs with annotations. In a second stage, the Lustre programs and their annotations are compiled into C code.

Safety verification via model checking: At the Lustre level, we perform SMT-based model checking to validate the specified safety requirements. The analyses combine different verification techniques, such as K-induction and property-directed reachability.

Static and dynamic verification: At the C level, we perform two complementary analysis: (i) static verification to validate stability proofs via interactive theorem proving in PVS; (ii) dynamic verification by performing runtime assertion checking.

Floating point analysis: The approach is to generate an equivalent proof of the control properties for the code as it has been done in the credible compilation chain but in a form that is analyzable, for its correctness in the semantics of floats, by an abstract interpreter such as fluctuat.

Case studies: The toolchain has been applied to three case studies: (i) an Engine FADEC (ii) a Fault Tolerant Quanser 3-DOF helicopter and (iii) the Transport Class Model (TCM).

The CrAVES project is also related to an associated French ANR project CAFEIN ANR-INSE-2012-007 (Combination of Formal Analyses for the Study of Numerical Invariants) with seven partners gathering academics, industrial users (Rockwell Collins) and tool providers (Prover Technology). The collaboration between the two projects has been sustained by visits, exchange of students and six coauthored publications (NSF-ANR) including a best paper award at FMICS'13.

Temesghen Kahsai
License: Creative Commons 2.5

Other available formats:

CrAVES : Credible Autocoding and Verification of Embedded Software