Source code static analysis of the IXV OBSW (on board software) using a SonarQube based code quality platform

Activity:

Prime contractor: SpazioIT ~ Presenter: Maurizio Martignano

ESA TO: Andreas Jung

Abstract:

In order to perform this activity Spazio IT has developed a code-centric quality platform based on SonarQube and using proprietary tools (e.g. SCITOOLS Understand, Gimpel Software PC-Lint) as well as open source tools (e.g. Carnegie Mellon CBMC and CEA-List / INRIA Frama-C). The presentation describes both the developed platform and a methodology Spazio IT has identified to apply in an effective way model checking and abstract interpretation techniques also to large code bases. Finally some information is provided on how the same code quality platform is used by AIRBUS Helicopters for the maintenance of Ada code.

Background:

SonarQube: A code-centric quality platform

SonarQube is an open platform for managing source code quality. SonarQube takes as input a set of source code files and a set of analyses results (produced by some external tools), stores both sources and results in a database and makes available the gathered information via a dynamic website where the results are shown in the context of the code itself. Analyses on the same code base can be performed at different moments in time and SonarQube keeps track of the code changes, code evolution. Problems found during analyses (a.k.a. issues) can be managed directly from within SonarQube, e.g. identifying false positives, assigning issues to developers, checking their status (if they have been solved), and so on… SonarQube can be used for one-off audits, but has been designed to support global continuous improvement strategy on code quality and should be used as a shared central repository for quality management.

CBMC & Frama-C: effective use of model checking and abstract interpretation

Model checking and abstract interpretation seem to be very promising techniques for source code static analyses. CBMC is a Bounded Model Checker for ANSI-C and C++ programs that allows verifying array bounds (buffer overflows), pointer safety, exceptions and user specified assertions. Frama-C Value Analysis plugin studies how variables are used by a given portion of code and is based on abstract interpretation while the WP plugin uses the weakest precondition calculus to verify assertions specified in ACSL. Unfortunately, the problem with these tools is that the amount of computational resources they require grows exponentially with the size of the portion of code to be analysed; most of the times it is practically unfeasible using them to analyse real case code bases. Spazio IT has developed a methodology (i.e. a process - a “how-to”- plus a set of ancillary scripts and tools) allowing the application of CBMC and Frama-C also to large code bases. The basic idea on which this methodology is based on “lowering the expectation”, e.g.: “Though checking the model of a software system may be unfeasible, is it still possible to obtain some interesting information by performing only the model generation phase?” Or “Even if it is not possible to make a value analysis of a software system as a whole, if this analysis is performed on all its functions (one function at a time) are the obtained results still of interest?” Or “How a large system can be sensibly partitioned in smaller “chunks” that can be “digested” by CBMC and Frama-C?” The presentation will show how this methodology has been applied to the IXV code base. Obviously the results of the performed analyses have been integrated into SonarQube.