DAEDALUS *

Validation of Critical Software by Static Analysis and Abstract Testing

Executive Summary

Abstract

Present software verification methods (such as testing, simulation, code review and formal methods including deductive methods or model checking) do not scale up for software of several hundred thousand lines, in particular for essential properties in embedded critical software such as absence of runtime errors, worst-case execution time, data races, precision of floating-point computations, etc. The DAEDALUS project has explored static analysis methods based on abstract interpretation. The approach has been shown to be effective on software provided by Airbus France.

Most security problems can neither be recognized by exhaustive tests nor by checking imperfect models so that the testing process of infinite state systems has a low coverage. Some required properties can hardly be checked at all because the controllability and observability conditions are simply not reproducible. Since present software validation techniques for concurrent, component-based, real-time software do not scale up, the consortium was built from the desire of Airbus' group in charge of software validation to

*DAEDALUS was the shared-cost research and technology development (RTD) project IST-1999-20527 of the European IST Programme of the Fifth Framework Programme (FP5) on the "validation of software components embedded in future generation critical concurrent systems by exhaustive semantic-based static analysis and abstract testing methods based on abstract interpretation".

explore static analysis and abstract testing. The project aimed at the industrialisation of methods and tools to support this new end-user methodology. The consortium has been built to solve major technical problems identified by the end-users by grouping world-wide recognised and renowned academic European researchers from Denmark, France, Germany and Israel to handle mid-term problems and prototype solutions as well as two pioneering companies (AbsInt from Germany and PolySpace Technologies from France) to cope with short and medium-term R+D and industrialisation.

The following tools have been developed or extended:

- Prediction of the Worst Case Execution Time (AbsInt, Saarland University)
- PolySpace Verifier, the first tool for the automatic detection of runtime errors at compilation time through static analysis of C and Ada 83 applications (PolySpace Technologies)
- FLUCTUAT: static analysis of floating-point operations (CEA-LIST)
- The Trier Data-Race Analyzer of multi-threaded C programs (Trier University)

All these analysers address concrete current verification problems as they can occur on real projects. The assessments by the end-user used realistic benchmarks. As a general result, experiments and assessment results show the effectiveness of abstract-interpretation-based analysis for checking dynamic properties of programs in realistic industrial contexts (check real properties on real programs with tools used by practitioners).

Based on DAEDALUS results, decisions have been taken by the end-user Airbus France to introduce abstract-interpretation-based static analysis as a verification technique within the Airbus software verification workbench.

Besides the tools the following prototypes have been produced:

- The size-change termination analyzer (DIKU, Copenhagen)
- Abstract interpretation of mobile systems – pi-calculus + ambient calculus (École normale supérieure)
- Octagone based relational analyzer (library, prototype) (École normale supérieure)

- TVLA, 3-Valued Logic Analysis Engine (Tel-Aviv University)
- The Universal Tiny Problem Solver as a feasibility study for checking reachability of deadlocks (Trier University)

DAEDALUS has also produced many advances in basic research that will help to improve future generations of program analysis and verification tools (see the list of publications). Areas are: abstract interpretation in software verification, correctness of optimizations, analysis for program termination and computational complexity, pipeline modelling for timing analysis, numerical stability of loops, relational numerical abstract domains, shape analysis, analysis of mobile systems, and certification of assembly code.

The cooperation of partners within the project was very intense. Especially the role of the end-user who invested serious effort to provide continuous feedback was highly acknowledged among the other partners.

The seminar for potential industrial users held in Saarbrücken, Germany, September 27, 2002 has attracted many attendees from automotive, public transport, avionic industry and software development companies.

With respect to exploitation, the project results are very promising. Most advances of the PolySpace Verifier are already integrated into the product and already used by many customers. WCET products will be available as standard products by AbsInt a few month after the end of the project. AbsInt has already found customers for WCET products outside the avionic area. For many other tools, bilateral cooperations have already started to continue the work. All partners regret the rather short duration of the project and the partners would have welcomed a continuation in form of a follow-up European project. It is expected that the project results will open considerable market opportunities for the European industry since the methodology equally applies e.g. to avionic, medical and automotive software.