Poster

**Formal Contracts for Runtime Verification Support In the Ada Programming Language**

André Pedro*
David Pereira*
Luis Miguel Pinho*
Jorge Sousa Pinto

*CISTER Research Center
CISTER-TR-150412

2015/03/24
Formal Contracts for Runtime Verification Support in the Ada Programming Language

André Pedro*, David Pereira*, Luis Miguel Pinho*, Jorge Sousa Pinto

*CISTER Research Center
Polytechnic Institute of Porto (ISEP-IPP)
Rua Dr. António Bernardino de Almeida, 431
4200-072 Porto
Portugal
Tel.: +351.22.8340509, Fax: +351.22.8321159
E-mail: anmap@isep.ipp.pt, dmrpe@isep.ipp.pt, lmp@isep.ipp.pt
http://www.cister.isep.ipp.pt

Abstract
Formal Contracts for Runtime Verification Support in the Ada Programming Language

André Pedro, David Pereira, Luís Miguel Pinho, and Jorge Sousa Pinto
{anmap, dmpe, lmp}@isep.ipp.pt, jsp@di.uminho.pt

Motivation

• Static Verification is not sufficient to cope with many of the challenges of modern and future generation real-time embedded systems:
  • state-explosion problem of model-checking;
  • limited automation in deductive reasoning, even with recent advances in SAT and SMT solvers.
• Most of the data important to certify a real-time embedded system is related to extra-functional properties:
  • Duration of tasks;
  • Energy consumption;
  • Temperature management;
  • Other cyber-physical properties.
• Unfortunately, most of the extra functional data is only available and verifiable during execution time.

Run-time Verification

• Run-time Verification is the discipline that studies formal theories and that proposes methods to generate monitors capable of observing and verifying formal specification during execution time:
  1. Formal specifications determine the property of interest that must be verified;
  2. Monitors are generated from that specification and are instrumented into the system.
• Typical contracts establish properties about the program that are verified via static approaches
• Run-time Verification behavior should follow the same principles:
  • Users define contracts about properties that he wishes to see verified upon execution;
  • The system is responsible for generating the monitors from those contracts.

Ada 2012 and Contracts

• Contracts enhance trust in the system by establishing a compromise between requirements and implementation
• Ada 2012 provides a sub-language for specifying contracts:
  • Checked at runtime via asserts, or;
  • Statically verified using the SPARK toolset.
• Contract language provides the ideal environment to specify properties that we need to be checked upon run-time (e.g., timed behavior of tasks)
• Run-time Verification contracts can be pre-processed to generate the monitors, and afterward removed, thus preserving the standard Ada 2012 contracts

References