ACE: Assurance, Composed and Explained

Goals of the Seedling Project

- Quantify and expose the assurance and risk of safety-critical software systems across their components using information from multiple sources of evidence.

1. Close the gap between traditional engineering and formal methods to better assess the cost/risk trade-off of system assurance.

2. Help to concentrate verification and validation efforts by guiding design and implementation improvement while supporting the isolation and solution of errors.

3. Combine reasoning with uncertainty and static analysis for quantified and supported metrics for assurance and risk for system components considering multiple assurance aspects.
ACE: Assurance, Composed and Explained

Overview of the Concept
ACE: Assurance, Composed and Explained
Approach and General Architecture

Data Providers
- Formal Verification
  - NuSMV/UPPAAL
- Static Analysis
  - Compiler/Doxygen
- Debugger
  - VS 2010
- Simulator
  - Physics Engine
- Memory Check
  - Valgrind
- Robot
  - Kuka API

System Model
Robotics
C++ code base
Controls
Mobile Robot

Artifacts
- Rule violations
- Counterexamples
- Errors and warnings
- Inheritance hierarchies
- Source code, methods, functions
- Traces
- Tag values
- Callstacks
- Threading and communication
- Collisions
- Forces
- Speeds
- Accelerations
- Memory leaks
- Uninitialized memory
- malloc() and free() read/writes
- Configuration
- Hardware parameters

ACE Algorithms
- Risk Propagation
- GoG Builder from Evidence
- Traceability to Root Cause

Graph of Graphs (GoG)

Unrestricted © Siemens AG 2019
Use Case: Industrial Robot Control and Human Safety

- Industrial robot control application developed using an in-house modeling and simulation environment

- Work environment shared by robots and humans

- Three levels of system logic to be analyzed
  - Application behavior defined in a graphical data-flow language
  - Simulation framework based on reusable and configurable objects (C++, C#)
  - Real-time control system runtime in the industrial controller (C++)

- Information from multiple data providers
  - Parsers for C++, C#, simulation model, architecture model
  - Static code analysis and model analysis tools
  - Output of simulation runs
ACE: Assurance, Composed and Explained
Composed Data Model for the Analysis

Architecture

Architecture + Source Files

Architecture + Source Files + Static Analysis

Architecture + Files + Code + Static Analysis

Architecture + Files + Code Management

Architecture View

Unrestricted © Siemens AG 2019
ACE: Assurance, Composed and Explained
Interactive Web Application with Assurance Heat Map
ACE: Assurance, Composed and Explained
Outlook: Data-driven approach complements existing techniques

Testing and Validation
- Detect reproducible errors in system components and integrated system
- Commonly manual and inherently incomplete

Formal Methods
- Exhaustive and conclusive w.r.t. specification
- Precise system models required
- Applicable to limited portions of the system

ACE
- Quantitative assessment of risk and assurance
- Leverages multiple and heterogeneous sources of evidence to identify possible issues
- Dynamic visual monitoring of calculated assurance
- Scalable from components to complete system
Dr. Gustavo Quirós
Senior Key Expert
Siemens Corporate Technology

755 College Road East
Princeton, New Jersey 08540
United States of America

Mobile: +1 (609) 216-5497
E-mail: gustavo.quiros@siemens.com

siemens.com