High Confidence Embedded Systems

ARO Workshop, May 1-2, 2003
Industry Session – Tim Johnson
GE Global Research
(johnsontl@research.ge.com)
Industrial Motivation for High Confidence Embedded Software

- Increasing use of software in all products
- Increasing complexity of programming languages and constructs
- Requirements and coding are manual processes – need executable specifications
- Composability, distributed systems
- Software is the final common path to end product performance.
Software Disasters
(MIT Tech Review, July 2002)

- Ariane 5 Launch
- NASA Mars Polar Lander Mission
- Radiation Therapy Overdose
- London Ambulance System Shutdown
- Civil Aircraft mistaken for military target
- Basic coding error rate: 100-150 errors/KLOC (professional programmers)
Some GE Products with Embedded Software

- Aircraft Engines
- Appliances
- Power Systems
  - Turbines
  - Generators
  - Substation Automation
- Locomotives
  - RR Signaling
- Industrial Systems
  - Breakers
  - Distribution Equipment
  - Security
- Medical Equipment
  - MR, X-Ray, CT
  - Hospital Networks

Many Standards must be Satisfied!
**CENELEC**: European Committee for Electromechanical Standardization.

<table>
<thead>
<tr>
<th><strong>Development of products which conform to European Standards for Homologation</strong> requires strict adherence to the criteria for:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Quality Management</td>
</tr>
<tr>
<td>- Safety Management</td>
</tr>
<tr>
<td>- Technical Safety Reports</td>
</tr>
</tbody>
</table>

The norms are applied either “entirely” or “partially” depending on the required System Safety Integrity level [SIL] which is to be achieved.

**Verification** is the activity which, through analysis and testing, checks that the result of every phase of the lifecycle is in accordance with CENELEC norms.

**Validation** is the activity that, through analysis and testing, checks that all the specified requirements are respected in accordance with CENELEC norms.
The process of **Verification** occurs at the end of each phase of the product development and it is therefore a “Top-Down” process.

By contrast, **Validation** is a “Bottom-Up” process starting from the lowest levels.

Traditional V&V Processes Are Manual – Cost/Reliability are Strong Automation Drivers
Safety Integrity Levels (SIL’s)

SIL = 0  non-safety-related

SIL = 2  only some specific functions are safety-related

SIL = 4  this is the maximum requirement level and is applied to all functions.

Many Control Systems are SIL = 4!
Embedded Systems, Safety, & V&V

- Partial overlap
- Embedded designs usually driven by performance requirements
- Safety may involve both embedded and non-embedded functions
- V&V needs usually high for both safety and embedded systems

Different Institutional Stakeholders Increase Complexity
Software Safety System Working Group
MIT, February 2003 – N. Leveson

Good academic/industry/government mix
ADI’s BEACON Successful in Aircraft Engines Use – Initiated at GE Research
ASSENT MISRA-C

ASSENT is a tool used to develop C programs for high-integrity, embedded applications. It provides a simple and effective way of checking the compliance of a program with the rules of MISRA-C via an attractive browser-based interface which provides easy navigation of program code and the MISRA-C guidelines.

Published originally for the benefit of the automotive industry, it is now finding popularity in a range of other industries where there is a need to develop reliable software in C.

For further information visit our products website [www.praxis-cs.com](http://www.praxis-cs.com).
HyTech: The HYbrid TECHnology Tool

HyTech is an automatic tool for the analysis of embedded systems. HyTech computes the condition under which a linear hybrid system satisfies a temporal requirement. Hybrid systems are specified as collections of automata with discrete and continuous components, and temporal requirements are verified by symbolic model checking. If the verification fails, then HyTech generates a diagnostic error trace. The standard reference to the HyTech algorithm is [1], and the standard reference to the HyTech tool is [2].


The HyTech Team

HyTech was developed by Tom Henzinger, Pei-Hsin Ho, and Howard Wong-Toi.

The HyTech Demo

The HyTech Papers

- Introduction, and guide to the literature
- The specification formalism of hybrid automata, and its theory
- The verification algorithm of *hybrid model checking*, and some verification examples
- The *original tool*, and more verification examples
Reactive Systems “Tester” - Recent Commercial Entry
Example: 21st Century Electric Infrastructure Project

Integrated Energy and Communications System Architecture

The project will develop an open, standards-based systems architecture for the data communications and distributed computing infrastructure that will enable the integration of a wide variety of intelligent electric power system components. This infrastructure will build upon prior industry infrastructure work, leverage the newest communications and distributed computing technologies available and will provide the interoperability/interoperability foundation for system development. This infrastructure will enable innovative services such as real-time pricing and energy management.

**Project Goal**

Develop an overall integrated energy and communications system architecture for the data communications networks and intelligent equipment necessary to support the self-healing grid and the integrated consumer communications interface.

**Project Objectives**

- Develop a complete set of systems requirements and architecture documents to support industry-wide enterprise architecture for the self-healing grid and integrated consumer communications interface.
- Contribute project results as appropriate to relevant Standards Development Organizations (SDOs) and industry consortia to effectively move the development of key open standards forward to develop a robust industry infrastructure.
- Apply Systems Engineering to the development of the architecture including
Discussion Topics

- Are new Certification Standards Needed?
- Embedded Systems Requirements Capture – More rigorous approaches needed?
- Can Simulation of Operating Environment Help with Safety?
- Automatic Code Generation?
- Automatic Test Generation?
- Automatic Verification?
- Transition paths for University Research?