Bi-Directional Traceability: The Hi-Five Framework Approach to Reliable Validation of Early System Designs

Martin Ouimet and Kristina Lundqvist
Embedded Systems Laboratory
Massachusetts Institute of Technology
July 2nd 2007
Project Outline

- Motivations
  - High cost of Validation & Verification (V&V)
  - Benefits of modeling
    - Building confidence into the system early on
    - Economics of bug detection and correction
  - Reuse of the state-of-the-art
    - Verification
    - Test case generation
Overview of Hi-Five

Current State

Ongoing Research

Abstracted

Needs Statements

Semi-Formal Specification

Formal Component and Interface Models

Implemented Code

Literate Specification Language

WCET Analysis

Implemented Code

Automated Test Case Generation

Autocode Generation

Leverage Existing Tools for Formal Analysis

Level of Abstraction

High

Low
Modeling Time and Non-Functional Properties

- Can time be reliably estimated for software, without an implementation?
  - Maybe you don’t care
    - Level of abstraction
    - Speed of software vs. rest of system
    - Nature of system
  - Maybe you care
    - Do the best you can
    - Estimates become constraints on implementation
    - Use feedback from implementation in model
    - Develop around a known platform with a library of components
Test Case Generation

- Model-Based Testing
  - Use model to generate test cases
  - Model acts as an oracle
  - Meaning of model coverage vs. implementation coverage?
The TASM Language

- Literate modeling language based on ASM
- Function + Time + Resources
- Duration is the key paradigm to represent time
- Time specified as interval to capture BCET, WCET, and uncertainty
The TASM Toolset

- Graphical Front-End for Specification, Simulation, and Analysis

- Integrates UPPAAL
  - To verify BCET and WCET paths in the model

- Integrates the SAT4J SAT solver
  - To verify Completeness and Consistency of models
R1: b0 -> b1
{ t := 605;
  if temp_seq_b = b0 then
    temp_seq_b := b1;
}

R2: b1 -> b2
{ t := 2285;
  if temp_seq_b = b1 and temperature >= 26 then
    temp_seq_b := b2;
    trying_to_cool_system := True;
    cooling := on;
}

R3: b1 -> b0
{ t := 1730;
  if temp_seq_b = b1 and temperature < 26 then
    temp_seq_b := b3;
}

R4: b2 -> b3
{ t := 1625;
  if temp_seq_b = b2 and temperature > 22 then
...
Bi-Directional Traceability

Requirements
Model
Model
Implemented System

Test Suite

Stimulus (Input)
Observation (Output)
Bi-Directional Traceability

- Requirements
- Model
- Implemented System
- Stimulus (Input)
- Observation (Output)
- Test Suite
Overview of Hi-Five

Abstracted

Implemented

Traceability

Literate Specification Language

High Level System Models

Component Models

WCET Analysis

Leverage Existing Tools for Analysis

Automated Test Case Generation

Automated Code Generation

Implemented Code
Case Study

- The Timeliner System
  - How long should the timeslice be?
  - What is the maximum execution time for one “pass”
  - How about the minimum execution time?
  - What are the timing properties of scripts?
Other Case Studies

- Electronic Throttle Controller (Ford)
- N-Modular Redundant Avionics (Draper)
- Production Cell
Questions?

- Thank you for your time
- For more information
  - http://esl.mit.edu/tasm
  - tasm@mit.edu