Architecture
Differencing for Self Management

Alexander Egyed
(Dave Wile and Robert Balzer)

Teknowledge Corporation

WOSS 2004
October 31, 2004
Overview

• How to detect deviation between the system behavior and its expected behavior?
  - Self awareness
  - Self Healing
  - Self Protection (Security)

• Architectural Differencing
  - Execute system
  - Simulate its model (iff model is behavioral)
  - Feed simulation with input given to execution
  - Compare output of simulation and execution
Architecture Differencing

Real Environment (Deployment)

Real System

Translator

Simulated System

Simulated Environment (Model)

Reflection

Differencer

List of Conflicts

Real Responses

Simulated Responses

Filter

in

out

in'

out'

in

out
Stimuli/Response

From/to user interface, environment, other components

System, subsystem, components, classes
Video-On-Demand
Observing the Real System

Real Environment (Deployment)

- **Probes:**
  - Instrumented source code, mediated interfaces (COTS)

- **Translators:**
  - Pre-defined mapping table
  - Generally 1:1 and many:1 mappings (aggregates)
System Reflection

Observed system stimuli (after translation)

Observed system response (after translation)
Simulating the Real System

Real Environment (Deployment)

Real System

Simulated Environment (Model)

Simulated System

Translator

Reflection

Real Responses

Concurrent execution and simulation of software components

(CD’04 paper)
Simulating components

Outside stimuli

Parameter for stimuli

Progress/state visualization
Real and Simulated Responses

Real Environment (Deployment)

Real System

Translator

Simulated System

Reflection

Real Responses

Simulated Responses
Model Reflection

Model stimuli (=observed, translated system stimuli)

Model response observed during simulation
Response Differencing

Real Environment (Deployment)

Real System

Translator

Simulated System

Simulated Environment (Model)

List of Conflicts

Reflection

Real Responses

Simulated Responses

Filter
Differencing

• *given: system input = model input*
• *assume: model and system are consistent*
  - Model state = system state

=> *Compares real response with simulated one*

If a difference exists then model or system wrong
Differencing

Real System, Stimuli, and Responses

outside

player

outside

?DISPLAY_MOVIE_LIST

?SELECT_MOVIE

?PLAY

SHOW_FRAME

SHOW_FRAME

SHOW_FRAME

SHOW_FRAME

?PAUSE

Δ1

t1

t2

Filtered, Model System, Stimuli, and Responses

outside

player1: VODPlayer

outside

?DISPLAY_MOVIE_LIST

?SELECT_MOVIE

?PLAY

SHOW_FRAME

SHOW_FRAME

Δn

4 events

2 events

+ ordering
Self Awareness

- System stimuli/responses
- Model stimuli/responses
- Simulation state reflects system state
- Coarse-grained
- External observer (avoids weaving into code)
Self Management

• Testing (fix system, fix model)

• Self Healing (vod, jbi)
  – Probes: *actual* system behavior (state)
  – Simulation: *expected* system behavior/state
  – Recovery benefits from knowing about their difference

• Self Protection (safe email)
  – Model and system consistent but system misbehaves!

• Self Optimization

• Self Reconfiguration

=> Models suggest self management
Thank you!

Alexander Egyed
aegyed@teknolodje.com
Final Observations

• **Purpose**
  - Testing
  - Self healing, protection, management (model-based)

• **Models useful during...**
  - development: design, testing
  - deployment: observe, correct, manage
  => Level of abstraction

• **Models describe ...**
  - the system
  - the deployed environment and expectations for self-management
Testing

• Generate input (stimuli) and observe the response (output)
  - *Input generated manually or automated*
  - *Expected output usually generated manually*

• Differencing computes the expected output automatically and compares it with the observed output