Dynamic and Static Testing

Prof. Chris Johnson,
School of Computing Science, University of Glasgow.
johnson@dcs.gla.ac.uk
http://www.dcs.gla.ac.uk/~johnson
Introduction: Testing

- The processes used during validation and verification.

- White and black boxes.

- Static and Dynamic techniques

- Case Study...
Definitions and Distinctions

- Black box tests:
  - tester has no access to information
  - about the system implementation.

- Good for independence of tester.

- But not good for formative tests.

- Hard to test individual modules...
• White box tests:
  – tester can access information about
  – the system implementation.

• Simplifies diagnosis of results.

• Can compromise independence?

• How much do they need to know?
Definitions and Distinctions

• Module testing:
  – tests well-defined subset.

• Systems integration:
  – tests collections of modules.

• Acceptance testing:
  – system meets requirements?

• Results must be documented.

  Changes will be costly.
• Functional testing:
  – test cases examine functionality;
  – see comments on verification.

• Structural testing:
  – knowledge of design guides tests;
  – interaction between modules...
  – test every branch (coverage)?

• Random testing:
  – choose from possible input space;
  – or beyond the ``possible"...
• **Dynamic testing:**
  – execution of system components;
  – is environment being controlled?

• **Static testing:**
  – investigation without operation;
  – pencil and paper reviews etc.

• Most approaches use both.

• Guide the test selection by using:
  – functional requirements:
  – safety requirements (see previous lecture).
<table>
<thead>
<tr>
<th>Lifecycle phase</th>
<th>Dynamic testing</th>
<th>Static testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements analysis and specification</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Top-level design</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Detailed design</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Implementation</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Acceptance testing</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>
Dynamic Testing

• Where do you begin?

• Look at the original hazard analysis;
  – demonstrate hazard elimination?
  – demonstrate hazard reduction?
  – demonstrate hazard control?

• Must focus both on:
  – expected and rare conditions.

• PRA can help - but for software?
• All of this will cost time and money.

1. Review test plans.
2. Recommend tests based on the hazard analyses, safety standards and checklists, previous accident and incidents, operator task analyses etc.
3. Specify the conditions under which the test will be conducted.
4. Review the test results for any safety-related problems that were missed in the analysis or in any other testing.
5. Ensure that the testing feedback is integrated into the safety reviews and analyses that will be used in design modifications.

• Must be planned, must be budgeted.
Dynamic Testing Techniques

• **Partitioning:**
  – identify groups of input values;
  – do they map to similar outputs?

• **Boundary analysis:**
  – extremes of valid/invalid input.

• **Probabilistic Testing:**
  – examine reliability of system.

• **(State) Transition tests:**
  – trace states, transitions and events.
Dynamic Testing Techniques

- **Simulation:**
  - assess impact on EUC (IEC61508).

- **Error seeding:**
  - put error into implementation;
  - see is test discover it (dangerous).

- **Performance monitoring:**
  - check real-time, memory limits.

- **Stress tests:**
  - abnormally high workloads?
Dynamic Testing: Software Issues

- Boundary conditions.
- Incorrect and unexpected inputs sequences.
- Altered timings - delays and over-loading.
- Environmental stress - faults and failures.
- Critical functions and variables.
- Firewalls, safety kernels & other safety features.
- Usual suspects...automated tests?
Limitations of Dynamic Testing

• Cannot test all software paths.
• Cannot even test all hardware faults.
• Not easy to test in final environment.
• User interfaces very problematic:
  – effects of fatigue/longitudinal use?
  – see section on human factors.
• Systems CHANGE the environment!
• How can we test for rare events? $10^9$ years?
Testing can prove the presence of errors, but not their absence.
Static Testing

- Don’t test the system itself.
- Test an abstraction of the system
- Perform checks on requirements?
- Perform checks on static code.
- Scope depends on representation...
Static Testing Techniques

• Walkthroughs:
  – peer review by other engineers.

• Fagan inspections:
  – review of design documents.

• Symbolic execution:
  – use term-rewriting on code;
  – does code match specification?

• Metrics:
  – lots (e.g. cyclomatic complexity);
  – most very debatable...
Static Testing Techniques

- **Sneak Circuit Analysis:**
  - find weak patterns in topologies;
  - for hardware not software.

- **Software animation:**
  - trace behaviour of software model;
  - Petri Net animation tools.

- **Performance/scheduling theory:**
  - even if CPU scheduling is static;
  - model other resource allocations.

- **Formal methods:** cf 00-60 with DO-178B...
GPS and GLONASS Augmentation

GPS/GLONASS Satellites With Pseudolite Augmentation

1- User calculates Position

2- Applies correction to Calculated Position

3- Calculates error

4- Transmits error

1- Receiver calculates Estimated Position

2- Compares to its own known Position

Space-Based Augmentation System

GPS/GLONASS Satellites

Reference Stations

Master Station

GEO

Integrity & Ranging+ corrections

Augmented Navigation

ICAO Required Navigation Performance

- **Accuracy.**
  - How correct is the aircraft position estimate.

- **Integrity.**
  - Largest aircraft position error without detection;

- **Availability.**
  - How often systems give desired Accuracy/Integrity;

- **Continuity.**
  - Probability that operation can be completed.
## ICAO SARPS high-level integrity requirements on Signal In Space

<table>
<thead>
<tr>
<th>Typical Operation</th>
<th>Horizontal Alert Limit</th>
<th>Vertical Alert Limit</th>
<th>Integrity</th>
<th>Time to alert</th>
<th>Continuity</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>En-route</td>
<td>2 NM</td>
<td>N/A</td>
<td>$1 \times 10^{-7}$ /h</td>
<td>15 s</td>
<td>$1 \times 10^{-4}$ /h to $1 \times 10^{-8}$ /h</td>
<td>0.999 to 0.99999</td>
</tr>
<tr>
<td>En-route (terminal)</td>
<td>1NM</td>
<td>N/A</td>
<td>$1 \times 10^{-7}$ /h</td>
<td>15 s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial approach, NPA departure</td>
<td>0.3 NM</td>
<td>N/A</td>
<td>$1 \times 10^{-7}$ /h</td>
<td>10 s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>APV-I</td>
<td>40.0 m</td>
<td>50 m</td>
<td>$1-2 \times 10^{-7}$ /app (150s)</td>
<td>10 s</td>
<td>$1 \times 10^{-6}$ /h in any 15s</td>
<td></td>
</tr>
<tr>
<td>APV-II</td>
<td>40.0 m</td>
<td>20 m</td>
<td>$1-2 \times 10^{-7}$ /app (150s)</td>
<td>6 s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAT I</td>
<td>40.0 m</td>
<td>15-10 m</td>
<td>$1-2 \times 10^{-7}$ /app (150s)</td>
<td>6 s</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Three EGNOS Services

- **Open Service**
  - Free service started October 2009.

- **Safety-of-Life Service (SoL).**
  - For safety-critical industries certified against Single European Sky/ICAO requirements 2011.

- **EGNOS Data Access Server (EDAS):**
  - Terrestrial commercial data disseminates through non-GEO means, EGNOS data within performance boundaries in real time supporting professional market.
New Forms of Interaction with SBAS

- 40% of losses Controlled Flight into Terrain

Crews must assimilate
- barometric and radio altitude instruments,
- vertical speed indicator,
- ground proximity warning systems,
- terrain depiction systems,
- flight management computer (FMC) etc.

- Opportunity for SBAS support…
Emergency broadcast if cellular infrastructure destroyed/unavailable.
New Forms of Interaction– CNSAS Architecture

Integration of EC Projects in Emergency Infrastructures.
New Forms of Interaction with SBAS

Integration of EC Projects in Emergency Infrastructures.
New Forms of Interaction with SBAS

Integration of EC Projects in Emergency Infrastructures.
There Can Still Be Problems...

EGNOS is divided into four functional segments:

1) The ground segment is composed of the following stations/centres which are mainly located in Europe and are interconnected between themselves through a local network:

- 34 Ranging and Integrity Monitoring Stations (EUMS) + seven being deployed: they receive the satellite signals and send this information to the MCC centres.
- 4 MCC (control and processing centres) receive the information from the EUMS and generate correction messages to improve satellite signal accuracy and inform messages on the status of the satellites (integrity). The MCC acts as the EGNOS "brain".
- 24 PPS (stations that access the geostationary satellites): they receive the correct messages from the MCCs for the upload of the data stream to the geostationary satellite and the generation of the GPS-like signal. This data is then transmitted to the EGNOS users via the geostationary satellite.

Home > EGNOS SYSTEM > System description

Current Architecture

Signal in space

Normal operation

Normal operation

Normal operation

PRN124

SIS outage

Since:
07:05 UTC

Expected recovery:
19/05 07:50 UTC

Histories
Conclusions: Testing

- The processes used during
  - validation and verification.

- White and black boxes.

- Static and Dynamic techniques

- Case Study...
Any Questions…