Decentralized Model-Based Testing of Distributed Systems
Presented by João Pascoal Faria and Bruno Lima
J. Pascoal Faria

Associate Professor for Software Engineering at FEUP & researcher at INESC TEC.

15+ years of research experience in test automation.

President of CS/03 – Sectorial Commission for Quality in ICT, of the Portuguese Quality Institute (IPQ)
Outline

Previous Work

UML Checker
Industry Survey

Current Work

Testing Distributed Systems
Observability Analysis & Enforcement
Controllability Analysis & Enforcement
Looking for More Case Studies
Previous Work
Previous Work: UML Checker\textsuperscript{[1]} (1/2)

**What:** Toolset for automatically testing OO implementations against behavior specifications given by test-ready UML sequence diagrams (SDs).

**How:**
- A plug-in for the Enterprise Architect modeling tool that, with a single click, generates extended JUnit test cases from the SDs in the model, executes them on the Java implementation under test, and presents back visually in the model test results and coverage info.
- A runtime test library based on AspectJ, that provides significant extensions to JUnit to handle internal interaction checking (method calls), test stubs, and user interaction testing.

**Why:** Improve the reliability of OO implementations and the consistency with design specifications.

https://blogs.fe.up.pt/sdbt
Previous Work: UML Checker[1] (2/2)
Please rate the degree of importance of testing each of the following features of distributed and heterogeneous systems:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Very Small</th>
<th>Small</th>
<th>Medium</th>
<th>High</th>
<th>Very High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple platforms</td>
<td>4%</td>
<td>5%</td>
<td>28%</td>
<td>33%</td>
<td>30%</td>
</tr>
<tr>
<td>Non deterministic behaviors</td>
<td>4%</td>
<td>12%</td>
<td>33%</td>
<td>34%</td>
<td>16%</td>
</tr>
<tr>
<td>Time constraints</td>
<td>3%</td>
<td>12%</td>
<td>33%</td>
<td>35%</td>
<td>17%</td>
</tr>
<tr>
<td>Parallelism and concurrency</td>
<td>3%</td>
<td>10%</td>
<td>32%</td>
<td>36%</td>
<td>19%</td>
</tr>
<tr>
<td>Interactions between components of the system</td>
<td>1%</td>
<td>22%</td>
<td>32%</td>
<td>38%</td>
<td>44%</td>
</tr>
<tr>
<td>Interactions between the system and the environment</td>
<td>1%</td>
<td>3%</td>
<td>24%</td>
<td>38%</td>
<td>33%</td>
</tr>
</tbody>
</table>

147 valid responses by professionals attending UCAAT 2015 and Testing Portugal 2015
Please rate the degree of importance of each of the following features of a test automation solution for distributed and heterogeneous systems:

- Support for multiple platforms:
  - Very small: 3%
  - Small: 5%
  - Medium: 22%
  - High: 41%
  - Very high: 30%

- Support for automatic test stub generation:
  - Very small: 1%
  - Small: 13%
  - Medium: 36%
  - High: 36%
  - Very high: 14%

- Support for test coverage analysis:
  - Very small: 2%
  - Small: 8%
  - Medium: 31%
  - High: 41%
  - Very high: 18%

- Support for automatic test case generation:
  - Very small: 4%
  - Small: 13%
  - Medium: 35%
  - High: 35%
  - Very high: 13%

- Support for automatic test case execution:
  - Very small: 1%
  - Small: 3%
  - Medium: 22%
  - High: 34%
  - Very high: 41%
Testing nowadays distributed, heterogeneous and time-constrained systems, such as IoT systems, is particularly important and challenging.

Some of the challenges are:

- the difficulty to test the system as a whole due to the number and diversity of individual components;
- the difficulty to coordinate and synchronize the test participants and interactions, due to the distributed nature of the system;
- the difficulty to test the components individually, because of the dependencies on other components.
Current Work: Distributed Scenario-based Testing Approach

1. Specification of test scenarios using an industry standard notation (e.g. UML SDs).
2. Pre-processing of test scenarios, to ensure readiness for decentralized testing.
3. Automatic translation of test scenarios into executable tests.
4. Distributed test execution, with local testers running close to the distributed CUTs, coordinated by a central tester, to increase test effectiveness and efficiency.
   - Better fault detection and localization
   - Minimal communication overhead
5. Automatic mapping of test results back onto the visual model.
Current Work: Observability Analysis & Enforcement \[4,5\]

Non-locally observable

\[\begin{array}{c}
\text{L1} \\
\downarrow^a \\
\text{opt} \\
\downarrow^{\text{b}} \\
\text{L3}
\end{array}\]
Current Work: Observability Analysis & Enforcement [4,5]
Non-locally observable

Locally observable

Current Work: Observability Analysis & Enforcement [4,5]
Current Work: Controllability Analysis & Enforcement [4,5]

Non-locally controllable

L1  L2  L3

alt a b

a

b

c
Current Work: Controllability Analysis & Enforcement \([4,5]\)

Non-locally controllable

\[
\begin{array}{ccc}
\text{alt} & a & b \\
L1 & & \\
\text{c} & & \\
L2 & b & \\
L3 & & \\
\end{array}
\]

race

non-local choice
Current Work: Controllability Analysis & Enforcement \[4,5\]

Non-locally controllable

Locally controllable

Non-local choice
Current Work: Controllability Analysis & Enforcement \[4,5\]

<table>
<thead>
<tr>
<th></th>
<th>L1</th>
<th>L2</th>
<th>L3</th>
</tr>
</thead>
<tbody>
<tr>
<td>alt</td>
<td>a</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>c</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Non-locally controllable

<table>
<thead>
<tr>
<th></th>
<th>L1</th>
<th>L2</th>
<th>L3</th>
</tr>
</thead>
<tbody>
<tr>
<td>alt</td>
<td>a</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>c</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Locally controllable

- Race
- Non-local choice

- Coordination message
Current Work: Controllability Analysis & Enforcement [4,5]

Non-locally controllable

Care Receiver

Fall Detection APP

AAL4ALL Portal

- fall_signal
- confirm? 
- yes
- no
- notify_possible_fall
- notify_fall
Current Work: Controllability Analysis & Enforcement \[4,5\]

Non-locally controllable

- Care Receiver
- Fall Detection APP
- AAL4ALL Portal

Locally controllable

- Care Receiver
- Fall Detection APP
- AAL4ALL Portal

alt

---

Current Work: Controllability Analysis & Enforcement \[4,5\]

- Non-locally controllable
- Locally controllable

fall_signal
confirm? \{..1\}

notify_fall
notify_possible_fall

yes \{..1\}
no \{..1\}
Current Work: Controllability Analysis & Enforcement \([4,5]\)

Non-locally controllable

- **Fall Detection APP**
  - fall_signal
  - confirm? \{..1\}

- **AAL4ALL Portal**
  - e1
  - e2
  - e3
  - e4
  - e5
  - e6
  - e7
  - e8
  - e9
  - e10
  - e11
  - notify_possible_fall

Locally controllable

- **Fall Detection APP**
  - fall_signal
  - confirm? \{..1\}

- **AAL4ALL Portal**
  - e1
  - e2
  - e3
  - e4
  - e5
  - e6
  - e7
  - e8
  - e9
  - e10
  - e11
  - notify_possible_fall

- **notify_fall**
  - e13

Current Work: Controllability Analysis & Enforcement \([4,5]\)
Current Work: Looking for more Case Studies

What we need:

Real world case studies of distributed, heterogeneous and time-constrained IoT systems that require testing.

Optionally, already defined test scenarios in natural language or diagrams (otherwise, they will be collaboratively created).

What you’ll get:

Short term: Formalized test scenarios, already verified and improved regarding their feasibility and adequacy for decentralized test execution (pre-processing stage).

Mid/long term: Infrastructure for automatic, decentralized, test execution of those scenarios.
References


Future Events

QUATIC 2019
12th International Conference on the Quality of Information and Communications Technology
Ciudad Real, Spain, September 10-13, 2019

ICST 2020
13th IEEE International Conference on Software Testing, Verification and Validation
Porto, Portugal, March 23-27, 2020
Questions?