Testing the reliability of systems with unstable or low-quality network connectivity

Miroslav Bures, CTU in Prague

Red Hat Research Day Europe, Brno, Sep 16th 2022
Better tools for system testing

- Faster testing process able to detect an increased amount of defects
- **Complex software, electronics, IoT, mission-critical systems**

**IoT projects in integrated rescue system, medicine and defence**

- Czech Army, NATO ACT Innovation Hub, University of Defence, ...
Selected projects

Automated model-based generation of test scenarios for integration and end testing of automobiles

Test strategy and test automation for IoT-based rescue mission planning and management system

Monitoring of soldier vital functions to allow for more accurate triage and to minimize casualties
Too much dependent on a data network?

Reliability of a system operating with a limited network

IoT and complex software systems
Live example: Our Digital Triage Assistant (DTA) project

Soldiers’ status estimation

DTA back-end (can be mobile)

Integration with other defense systems

Soldiers’ body sensors

Smart glasses

User interface

Live example: Our Digital Triage Assistant (DTA) project

Soldiers’ body sensors

Smart glasses

User interface
Forget about quality 5G network
GSM use only in emergency
Mesh network needed
Every component can move
Stealth mode might be needed

Soldiers’ body sensors

DTA back-end (can be mobile)

Integration with other defense systems

Soldier’s status estimation

Smart glasses

User interface
Weak network situation examples (video)

Source: NATO Multimedia and University of Defence
Typical challenges

Stealth mode ON / OFF

A mesh network + terrain →
• Low bandwidth
• Intermittent connection
• Connectivity disrupted and restored

SYSTEM PROCESSES MUST RUN IN A RELIABLE WAY
Manual testing

1. Test basis
2. Test scenarios
3. Test execution

Test automation now

1. Test basis
2. Test scenarios
3. Test execution
Test automation now & future

- test basis
- test scenarios
- test execution

MBT way

- test basis
- SUT model
- test scenarios
- test execution

automate!
Test automation in the future?

Future?
How to test it?

A process flow model:

Process parts handled by devices / system modules

Model of network outage probability
How to test it?

Test the process by paths through it
System modeling

Oxygen platform by STILL
What is a “good test set”?  

4 test coverage criteria

<table>
<thead>
<tr>
<th>Evaluation criterion</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>T</td>
</tr>
<tr>
<td>$\bar{t}$</td>
<td>Average length of test cases in test set $T$</td>
</tr>
<tr>
<td>$l(T)$</td>
<td>Total length of test set $T$ measured in number of edges</td>
</tr>
<tr>
<td>$s(T)$</td>
<td>Length dispersion of the test cases in test set $T$, expressed by standard deviation of test case lengths; test case length is measured in number of edges</td>
</tr>
<tr>
<td>$u_nodes(T)$</td>
<td>Number of unique nodes in test set $T$</td>
</tr>
<tr>
<td>$u_edges(T)$</td>
<td>Number of unique edges in test set $T$</td>
</tr>
<tr>
<td>$b_nodes(T)$</td>
<td>Number of border nodes in test set $T$ for all LCZs of $G$</td>
</tr>
</tbody>
</table>

$eff\_edges(T) = \frac{u\_edges(T)}{l(T)} \cdot 100\%$  

$eff\_b\_nodes(T) = \frac{b\_nodes(T)}{l(T) + |T|} \cdot 100\%$  

How to compute it?

Number of algorithms possible

AI gives good results, e.g.
Artificial ANT colony  
Genetic algorithm  
...
AI example: Artificial Ant colony

Nature-inspired algorithm

Ants depositing their pheromone path

“Together” they compute the best test set
An example of results

Much better than common techniques currently in place
To get the best solution for sure

Algorithm adapts to particular system model

Algorithms combined together
• Portfolio strategy
• Composition of algorithm from blocks

Machine Learning used
Solving similar issues in your software testing? Get in touch!

Miroslav Bures
Software Testing IntelLigent Lab (STILL)
Dept. of Computer Science
Faculty of Electrical Engineering
Czech Technical University in Prague

miroslav.bures@fel.cvut.cz
http://still.felk.cvut.cz

@intelligent_lab