Multi-Platform SCADA GUI Regression Testing at CERN

Paul Burkimsher
Manuel Gonzalez-Berges
Stefan Klikovits
ICE Group, EN Department, CERN, Geneva, Switzerland
a.k.a.

Practical Experience With Sexy Software
Automatic Testing

• Hands-on, practical talk

• Why are you interested?
  – You can laugh at other people’s mistakes
  – You may be about to make the same journey (with same mistakes!) as me 😊

• What will you gain from this talk?
  – A desire to actually read our paper (?)
  – A chance to get your Testing System right, 1st time
Salient Points

• Beware the siren song
  – Truthful(!) salesmen
  – But what remains unsaid?

• Testing Cross-platform (vs Multi-platform)
  – Messy

• Virtual machines
  – Appropriate
Layout Of This Talk

- Introduction: GUI Quality Assurance
- Testing what? JCOP Framework
- How? Automatic; Squish
- Orchestration: Continuous Integration tool.

- Lessons: “In the light of experience…”
  - What went badly 1st time?
  - How did we respond?
GUI Quality Assurance (QA)

- Lots of people test their software (some even document it...)
  - Unit testing; Black box, white box
    - ~ Easy(?) Straightforward

- Testing SCADA (control) systems
  - Many Graphical User Interface (GUI) panels
    - Need testing too
    - Don’t want to do it manually

  - Not so easy. Especially when there is control logic behind the buttons etc
Testing What?

• JCOP Framework
  – A toolkit for end users to build their own control applications
  – Long-lifetime project (-10..+15 years)
  – Staff rotation
    • Natural turnover
    • Use Experiment staff/students
    • New developers -> unintended side-effects

(Re-)Testing is crucial
Commit new code & bug fixes to SVN

Many Components;

Different Developers over time

FwWg

User Community

Developer Community

Morning after: Emailed test report

Commit new code & bug fixes to SVN

Daily: JCOP Fw CIR

“Current Internal Release”

Automatic tests

Squish

Results
How Do We GUI-Test JCOP-Fw?

- Commercial tool:
  - (Rational Robot)
  - Squish (Qt) from Froglogic GmbH
- Very powerful
  - Intuitive
  - “Record my keystrokes”
  - “Record my mouseclicks”
  - Generates source code
  - Replay 😊
  - Regression testing done!
How do you know it worked?

• A library call like sqrt(2) would be easy to check

• But Fw tool is used to
  – Declare hardware in a new control system
  – Define alarms on values
    • Replay & verify get same definitions as yesterday.
How do you know it worked?

• A library call like sqrt(2) would be easy to check

• But Fw tool is used to
  – Declare hardware in a new control system
  – Define alarms on values
    • Replay & verify get same definitions as yesterday.

• User presses a button, e.g. to connect to the database
  – Verify that the LED is green
Squish can’t do everything

• Tools to verify screen conditions:
  – Fields contain correct values, are correct colour,

<table>
<thead>
<tr>
<th>Start</th>
<th>tst_verifyChangeManagementMode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pass</td>
<td>Verified</td>
</tr>
<tr>
<td>Pass</td>
<td>Verified</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Detail</th>
<th>True expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail</td>
<td>suite_FwCDSI:5.6:tst_verifyChangeManagementMode == Central</td>
</tr>
</tbody>
</table>

  | Pass | Comparison |

• …cannot verify application-specific things:
  – WinCC-OA (PVSS) from Siemens (ETM)
    • Export definitions from WinCC
  – Write our own comparison tool (Totem)
Problem: Software Versions

- Squish tests the JCOP-Framework,
  - which uses WinCC-OA,
    - which is built on Qt,
      - which uses Windows or Linux

- Any changes can break the pre-recorded tests
Problem: Software Versions

• Squish tests the JCOP-Framework, which uses WinCC-OA, which is built on Qt, which uses Windows or Linux

• Any changes can break the pre-recorded tests

Lesson 1

• Simply recording keystrokes and clicks is not scalable to many tests
  – We had to seriously re-factor the code into libraries to make it maintainable
Cross-platform Difficulties

• Squish development environment is pretty…
  …but that’s only half the story.
  – Overnight-runs started from a command line script

• Difficulties with the fundamentals
  – Bash to trigger the production runs:
    • Windows and Linux.
    • Seemed like a good idea at the time.
  – Incompatibilities
    • Cygwin-Bash (on Windows) vs Linux-Bash
    • One script for 2 platforms => Messy scripts
Powerful Solution?

- Squish has a client-server mode of operation

Unique (clean!) test scripts on Windows

Target application (AUT) on Linux.
Problems with Client-Server

• Many Linux X–Windows
  – Display back to the client (on Microsoft Windows)

• Remote file access
  – Returning results across the link

• In practice 😞 we suffered networking timeouts
  – All our displays (and open files) would then collapse

• Despite trying various workarounds (e.g. VNC), we had to change tack
Problems with Client-Server

• Many Linux X–Windows
  – Display back to the client (on Microsoft Windows)
  – Remote file access

• In practice
  – All our displays (and open files) would then collapse

• Despite trying various workarounds (e.g. VNC), we had to change tack

Lesson 2

Cross-platform gets messy

- Messy scripts.
- Messy timeouts.
Profound Reflection
Result of the Re-Think

- Abandon Bash
  - Use Python
    - Better compatibility across platforms
    - Consistent with use of Python within Squish

- Returned to running the Squish test scripts on the target platform being tested

- And...
Test Management

- Orchestrate many different platform instances (Windows XP, W7, Linux SLCn, n+1 etc)

- Looked at tools to
  - Distribute the testing
  - Centralise the results.

- Bamboo (from Atlassian)?
  - but their model is different to ours.
Chose Hudson

- Hudson is a Continuous Integration (CI) tool that is
  - Flexible
  - Open Source (free licensing)
- Works well
Operational Environment

• CERN Virtual Machine (VM) service.
• Dedicated machines make life simple
  – Different O/Ses
  – Different software versions
• VMs work like the real thing 😊
  – Functionality tests, not performance tests!
  – Speed/timing issues on different platforms (real or virtual)!

• CERN is happy too. (VM Server reallocation)
“Experience” Is Unfinished

- Real life is ongoing (fortunately).
- Setting up on Linux again right now.
Conclusions - I

• Sexiness doesn’t scale (Ask your girlfriend/boyfriend…)
  – Investment in good old-fashioned coding (libraries)

• Cross-platform can be tricky

• Virtual machines are great
Conclusions - II

- We’ve made a system to test Qt GUls, designed for multi-platform
  - Scalable
    - # tests
    - # machines
  - General
    - Applicable to testing other Qt GUls (e.g. Unicos)
  - Useful (!)
    - Reduced manual effort (time!) to release each new version of the Fw
    - Steady trickle of errors found (1..2 per month). (Pays off!)
Conclusions - III

• As for Froglogic’s Squish:
  – We love it, but (like with a husband or a wife) you’ve got to be prepared to work within (and on) the relationship
Questions?

© Scott Adams, Inc./Dist. by UFS, Inc.