GSC Conformance Test Suite
Architecture and Implementation

Elizabeth Fong and Eric Dalci
NIST
Smart Card Interoperability Workshop CTST 2003
2PM. May 12, 2003
Outline

- Conformance Test System
- Conformance Test System for the GSC-IS BSI
- Benefits of the Architectural Approach
- Step-by-Step Illustrations of the Demonstration

Disclaimer: Any commercial product mentioned is for information only; it does not imply recommendation or endorsement by NIST nor does it imply that the products mentioned are necessary the best available for the purpose.
Conformance Test System

Test assertions

Test suite

Test Harness

Implementation

Under Test

Printed output results
GSC-IS Conformance Test System for BSI

BSI test assertions (in Narrative Form) → BSI test assertions (in XML) → XSLT Stylesheet

XSLT Processor XALAN

Data parameters → BSI test source code → Candidate Implementation (Simulator)

Printed HTML
Output results
Benefits of Architectural Approach

- XML document provides a single source for maintenance and modification to test suites.
- Test source code is automatically generated by XSLT.
- XML document provides traceability and flexibility.
- Can reuse the XSLT for additional BSI commands.
- Fast development of tests for additional language bindings.
Step-by-Step Illustration of the Demonstration

- Test assertions in narrative form are testable statements and result in one or more test cases.
- Test Assertions in XML according to the DTD.
- XSLT Processor
- Automatic Generation of C or Java Source Code.
- Test Output Results.
Test Assertions in XML

• Document Type Definition (DTD) provides tags and attributes to be used to structure XML documents.
• 21 BSI test assertions are converted into 21 XML documents using the DTD.
• XML authoring tool – XML Spy™ is used.
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE test-method SYSTEM "javadtd.dtd">
<!--Connect  BSI Java Binding- Dec 16, 2002  Elizabeth Fong-->
<test-method>
  <method method-name="gscBsiUtilConnect">
    <description>This is the function to connect a card to a reader and get a card handle.</description>
    <reference>GSC-IS 4.4, F.2.2</reference>
    <parameter way="OUT" type="int" pname="hCard0200" initial-value="0"/>
    <parameter way="IN" type="String" pname="uszReaderName0200" initial-value="constants.defReaderName"/>
    <assertions id="U2.1">
      <purpose>To test gscBsiUtilConnect() using a good card inserted into a specified reader.</purpose>
      <scenario>
        <testcase-no>2.1</testcase-no>
        <called-method>mybsi.gscBsiUtilConnect(uszReaderName0200)</called-method>
        <expected-results>
          <return-value>hCard0200</return-value>
        </expected-results>
      </scenario>
    </assertions>
  </method>
</test-method>
XML to C Transformation using XSLT

- `gscBsiutilAcquireContext.xml`
- `gscBsiUtilConnect.xml`
- `StyleSheet.xslt`

Transform XML to C

- `gscBsiutilAcquireContext.c`
- `gscBsiUtilConnect.c`
- `Etc.`

`XSLT Processor`
Automatic Generator of Source Code

• XSLT processor, Xalan™ is used to generate C or Java code.
• Constants and parameters are defined in the data file.
• The source code is compiled and executed with a simulator. (This simulator will be replaced by the real implementation under test)
• Test results are compared with the expected output results to produce a test output report.
### GSC 13 Conformance testing suite

#### Java Binding ####

<table>
<thead>
<tr>
<th>Test Environment</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Software:</td>
<td>Java 200</td>
</tr>
<tr>
<td>Hardware:</td>
<td>AMD Athlon 2.0 GHz</td>
</tr>
<tr>
<td>Card reader:</td>
<td>Contact</td>
</tr>
<tr>
<td>Data model:</td>
<td>April 2003 ISO/IEC 7816-3 CS</td>
</tr>
<tr>
<td>Test Operator Name:</td>
<td>Bruce Bakken</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Method</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Method:</td>
<td>getCardInfo</td>
</tr>
<tr>
<td>Description:</td>
<td>This is the function to connect a card to a reader and get a read handle</td>
</tr>
<tr>
<td>Parameters:</td>
<td>String m_readHandle</td>
</tr>
<tr>
<td>Parameters OCT:</td>
<td>testCard888</td>
</tr>
</tbody>
</table>

### Test Result

<table>
<thead>
<tr>
<th>Assertion: U2.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose of the assertion:</td>
</tr>
<tr>
<td>Test case no:</td>
</tr>
<tr>
<td>Called method:</td>
</tr>
<tr>
<td>Thrown (BS)Exception:</td>
</tr>
<tr>
<td>Test status:</td>
</tr>
</tbody>
</table>

| Test case no: | 2.2.1 |
| Comment: | collected event test for error code: |
| Called method: | gcBeH( FileHandle.Err(Err1) ) |
| Expected exception code: | 255 NO.TXT AVAILABLE |
| Thrown (BS)Exception: | 255 NO.TXT AVAILABLE |
| Test status: | PASS |

| Test case no: | 2.2.2 |
| Comment: | collected event test with bad handle |
| Called method: | gcBeH( FileHandle.Err(Err1) ) |
| Expected exception code: | 255 BAD_HANDLE |
| Thrown (BS)Exception: | 255 NO.TXT AVAILABLE |
| Test status: | PASS |
Demonstration of BSI Conformance Testing