Testing from CSP-CASL

Markus Roggenbach (Swansea, Wales)
cooperation with
A Cavalcanti, M-C Gaudel, T Kahsai, B-H Schlingloff

Etelson, July 2010
Specification System Specification
System Boundaries

Figure 1 ep2 Context Diagram

- PUI-PMS User
- SEI-Settlement
- FII-Finance Institute
- CII-Card Issuer
- Attendant
- Merchant
- POS Mgmt. System
- Point of Service
- Cardholder
- Card
- Finance Institute
- Issuer
- Terminal
- Card

Part of the Specification ep2 (detailed)
Part of the Specification ep2 (overview)

M.Roggenbach: Testing; Etelsen, July 2010
The EP2 Consortium

Cornèr Bank Card Center, Credit Suisse / Swisscard aecs, Swiss Post, Telekurs Multipay AG, Telekurs Card Solutions AG, Diners Club Schweiz AG, JCB International Co. Ltd., Verband Elektronischer Zahlungsverkehr VEZ.

Some terminal manufacturers:

EP2 in CSP-CASL

- Architectural Level
- Abstract Component Level
- Concrete Component Level

Informal Design Process
Modelling
Formal Specification
Analysing / Proving

Csp-Casl Spec $S_{p0}$
Csp-Casl Spec $S_{p1}$
Csp-Casl Spec $S_{p2}$

Informal Refinement
Formal Refinement
Modelling / Implementation
Testing

M. Roggenbach: Testing; Etelsen, July 2010
Overview

Testing from CSP-CASL
Testing from CSP/CIRCUS by MCG/AC
Relating the two approaches
Test practice with EP2
Testing from CSP-CASL
Does a test case encode the specified behaviour?

The color of test $T$ with respect to $(D, P)$ is a value in \{red, yellow, green\}.
The formal definition of coloring

For consistent $D$:

- **color(T) = green** iff
  for all $M \in \text{Mod}(D)$ and all $\nu : X \rightarrow M$:
  
  (a) $\text{traces}([T]_\nu) \subseteq \text{traces}([P]_{\emptyset : \emptyset \rightarrow \beta(M)})$ and
  (b) for all $tr = \langle t_1, \ldots, t_n \rangle \in \text{traces}([T]_\nu)$, $1 \leq i \leq n$:
    $(\langle t_1, \ldots, t_{i-1} \rangle, \{t_i\}) \notin \text{failures}([P]_{\emptyset : \emptyset \rightarrow \beta(M)})$

- **color(T) = red** iff
  for all models $M \in \text{Mod}(D)$ and $\nu : X \rightarrow M$:
  
  $\text{traces}([T]_\nu) \not\subseteq \text{traces}([P]_{\emptyset : \emptyset \rightarrow \beta(M)})$

- **color(T) = yellow** otherwise.
From terms to stimuli and observations

Given: System under Test (SUT) and specification \((D, P)\)

A PCO \(\mathcal{P} = (\mathcal{A}, \|\ldots\|, D)\) of an SUT consists of:

- an alphabet \(\mathcal{A}\) of primitive events
- a mapping \(\|\ldots\| : \mathcal{A} \rightarrow T_\Sigma\)
- a direction \(D : \mathcal{A} \rightarrow \{ts2sut, sut2ts\}\).
Test experiment with evaluation “on the fly”

\[ \ldots \]

Red test case: “observation a expected”

If the direction \( D(a) = \text{sut2ts} \) and we receive a we obtain the test verdict by continuing to execute the SUT against the remaining test case.

If the direction \( D(a) = \text{sut2ts} \) and we receive some \( b \) different from \( a \) or if a timeout occurs, then the test verdict is \textit{pass}.

\[ \ldots \]
Assumption: SUT is a “deterministic” system.

The execution of a test $T$ at a particular SUT yields a verdict in

$\{ \text{pass, fail, inconclusive} \}$

w.r.t. to a specification $(D, P)$.

• Pass – increased confidence in SUT w.r.t. $(D, P)$
• Fail – violation of the intentions described in $(D, P)$
• Inconclusive – neither increased nor destroyed confidence
Testing from CSP / Circus by MCG/AC
Here: CSP and its traces model only.

Related to Jan Peleska’s Test Theory for CSP:
- JP: detects safety failure
  (also other classes of implementation faults)
- MCG/AC: characterize traces refinement.
Testability hypotheses

- SUT behaves like some unknown CSP process \( \text{process}(SUT) \).

- Complete testing assumption:
  "There is some known integer \( k \) such that, if a test experiment is performed \( k \) times, then all possible behaviours are observed."
Test cases

$Exhaust_{\mathcal{T}}(P) := \{ T_{\mathcal{T}}(s, a) \mid s \in traces(P), s \dashv a \notin traces(P) \}.$

$T_{\mathcal{T}}(s, a) :=$

\[ inc \rightarrow a_1 \rightarrow inc \rightarrow a_2 \rightarrow inc \cdots \rightarrow a_n \rightarrow pass \rightarrow a \rightarrow fail \rightarrow Stop, \]

where $s = \langle a_1, a_2, \ldots, a_n \rangle.$
Test execution

\[
\text{Execution}_{\text{process}(SUT)}^{Sp}(T) = \\
(\text{process}(SUT) \parallel [\alpha(Sp)] \parallel T) \setminus \alpha(Sp)
\]
Characterization theorem

\[ Sp \sim_T \text{process}(SUT) \]

iff

for all tests \( T \in Exhaust_T(Sp) \) and

for all \( t \in \text{traces}(\text{Execution}^{Sp}_{\text{process}(SUT)}(T)) : \)

\( \text{last}(t) \neq \text{fail}. \)
Relating the two approaches
(Work in progress)
Restrictions

1. Data in CSP-CASL: primitive events, e.g.
   \[
   \text{spec } \text{Alphabet}_A = \\
   \quad \text{free type } s_A ::= \ a_1 \mid a_2 \mid \ldots \mid a_n \\
   \text{end}
   \]
   This allows to “confuse” (D,P) in CSP-CASL with P in CSP.

2. Events of the SUT = alphabet

3. SUT is a “deterministic” system only.
Coloring and $Exhaust_\mathcal{T}(P)$

1. For all CSP processes $T \in Exhaust_\mathcal{T}(P)$ holds:

$$\text{color}(T \setminus \{\text{inc, pass, fail}\}, P) = \text{red}.$$ 

2. For all red linear test cases $R$ holds: there exists a $T \in Exhaust_\mathcal{T}(P)$ such that

$$R \rightarrow_\mathcal{T} T \setminus \{\text{inc, pass, fail}\}.$$
Test verdict: “From TK/MR/HS to MCG/AC”

TK/MR/HS approach

Let $T \in \text{Exhaust}_T(Sp)$.

Let the execution of $T \setminus \{\text{inc, pass, fail}\}$ at the SUT yield “pass” for some PCO with $A = \alpha(P)$. 
Test verdict: “From TK/MR/HS to MCG/AC”

TK/MR/HS approach
Let $T \in \text{Exhaust}_T(Sp)$.
Let the execution of $T \setminus \{\text{inc, pass, fail}\}$ at the SUT yield “pass” for some PCO with $A = \alpha(P)$.

MCG/AC approach
Then one can argue:
For all $t \in \text{traces} \left( \text{Execution}^{Sp}_{\text{process} \left( \text{SUT} \right)}(T) \right)$: $\text{last}(t) \neq \text{fail}$. 

M.Roggenbach: Testing; Etelsen, July 2010
Future work in this cooperation

- Complete the comparison on the CSP level
- Figure out the Circus and CSP-CASL level (?)
- Test selection / generation
Test practice: EP2 in CSP-CASL
Hardware-in-the-loop

EP2 Test Environment

EP2 Test Driver and Monitor
Test: Configuring 8 different Credit-Cards

sessionStart::D_SI_Init_SessionStart [sut2ts]

ntf1::D_SI_Init_Notification [ts2sut]
ack1::D_SI_Init_Acknowledge [sut2ts]

ntf2::D_SI_Init_Notification [ts2sut]
ack2::D_SI_Init_Acknowledge [sut2ts]

...

ntf1::D_SI_Init_Notification [ts2sut]
ack1::D_SI_Init_Acknowledge [sut2ts]

sessionEnd::D_SI_Init_SessionEnd [ts2sut]

Color: “green”

M.Roggenbach: Testing; Etelsen, July 2010
Test case excerpt: XML encoding

... 
<?xml version="1.0" encoding="UTF-8"?>
  <ep2:actcfgdataack msgnum="2634">
    <ep2:AcqID>00000000004</ep2:AcqID>
    <ep2:TrmID>TERM1234</ep2:TrmID>
  </ep2:actcfgdataack>
</ep2:message>

<?xml version="1.0" encoding="UTF-8"?>
  <ep2:sessend msgnum="2635">
    <ep2:AcqID>00000000004</ep2:AcqID>
    <ep2:TrmID>TERM1234</ep2:TrmID>
    <ep2:TrxSeqCnt>23534</ep2:TrxSeqCnt>
  </ep2:sessend>
</ep2:message>
Future work in testing EP2

Find a “nice” PCO description (equivalence classes of XML messages).

In cooperation with Six Card Solutions:
• Color and automatize the company’s test suite.
• Color and automatize the certifying test suite of EP2 (?).