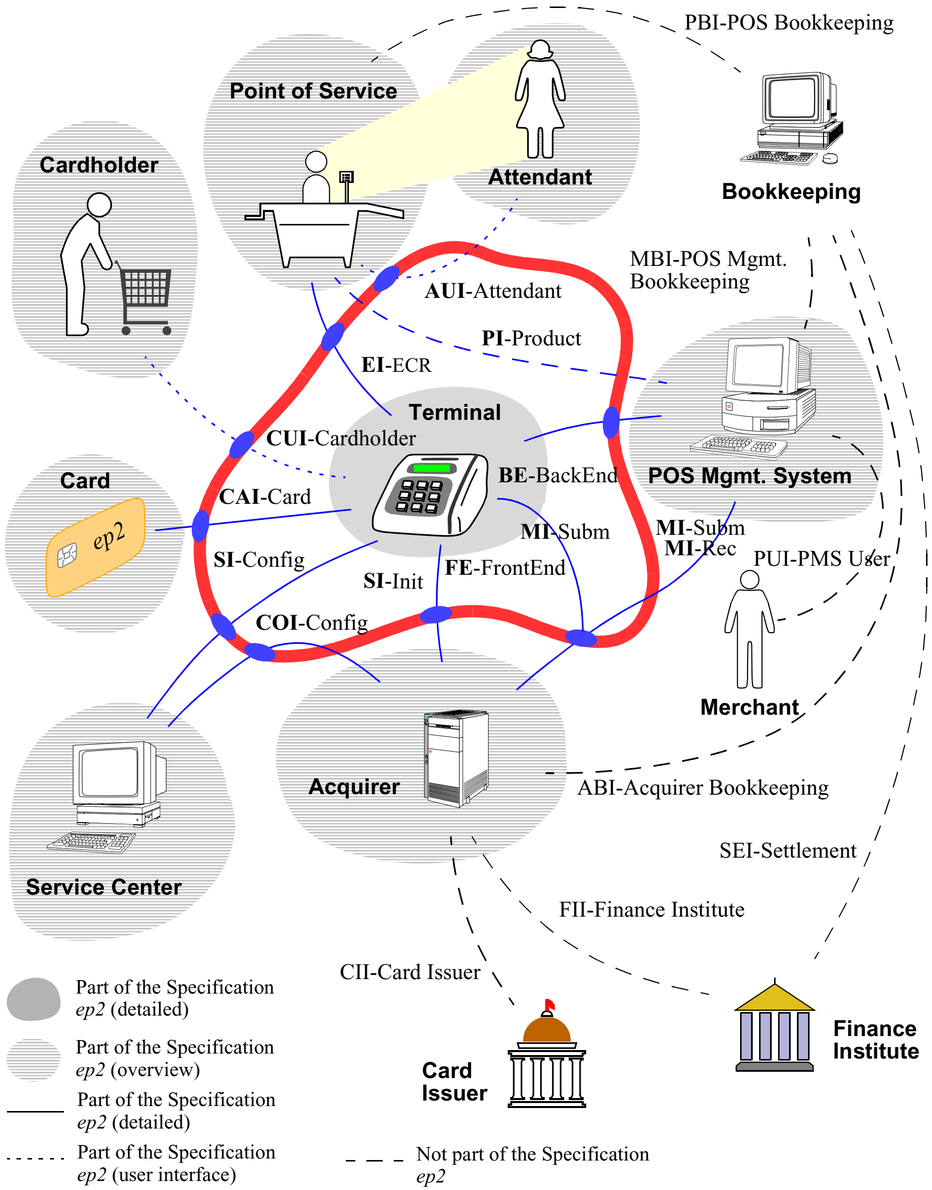
Testing from CSP-CASL

Markus Roggenbach (Swansea, Wales)

cooperation with

A Cavalcanti, M-C Gaudel, T Kahsai, B-H Schlingloff

Etelsen, July 2010


PBI-POS Bookkeeping
Point of Service
Attendant
Bookkeeping
Cardholder
MBI-POS Mgmt.
Bookkeeping
AUI-Attendant
PI-Product
EI-ECR
Terminal
CUI-Cardholder
POS Mgmt. System
Card
BE-BackEnd
CAI-Card
ep2
MI-Subm
MI-Subm
MI-Rec
PUI-PMS User
SI-Config
SI-Init
FE-FrontEnd
COI-Config
Merchant
Acquirer
ABI-Acquirer Bookkeeping
Service Center
SEI-Settlement
FII-Finance Institute
CII-Card Issuer
Finance
Institute
Card
Issuer
Part of the Specification
ep2 (detailed)
Part of the Specification
ep2 (overview)
Part of the Specification
ep2 (detailed)
Part of the Specification
ep2 (user interface)
Not part of the Specification
ep2

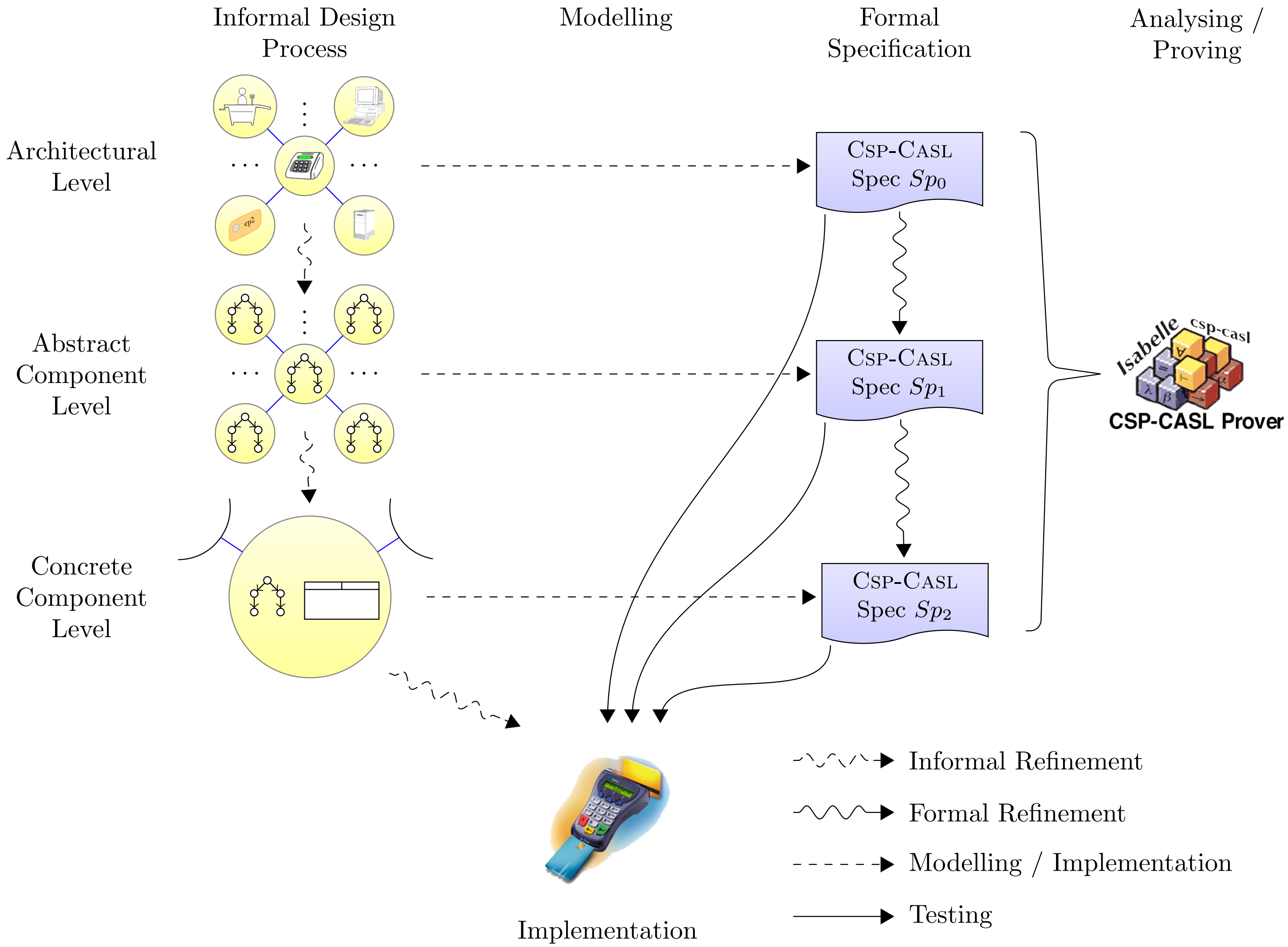
The EP2 Consortium

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:

Six Card Solutions, Epsys AG, CCV-CardPay AG jeronimo SA, CCS Card Solutions, Telekurs Card Solutions AG, ICP Paysys GmbH, Thales e-Transactions GmbH.

EP2 in CSP-CASL



Overview

Testing from CSP-CASL

Testing from CSP/ CIRCUS by MCG/AC

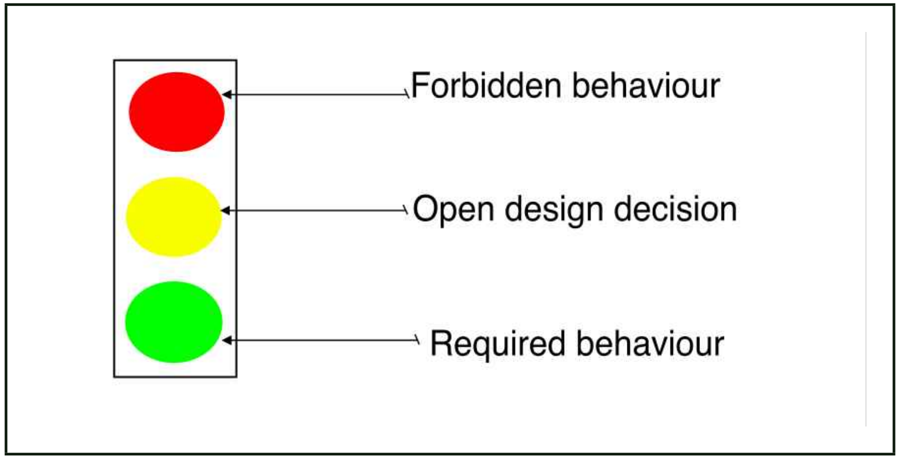
Relating the two approaches

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 $M \in \mathbf{Mod}(D)$ and all $\nu : X \to M$:
  (a) $traces([\![T]\!]_\nu) \subseteq traces([\![P]\!]_{\emptyset:\emptyset\to\beta(M)})$ and
  (b) for all $tr = \langle t_1, \ldots t_n\rangle \in traces([\![T]\!]_\nu)$, $1 \leq i \leq n$:
  $(\langle t_1, \ldots, t_{i-1}\rangle, \{t_i\}) \notin failures([\![P]\!]_{\emptyset:\emptyset\to\beta(M)})$
- color(T) = red iff
  for all models $M \in \mathbf{Mod}(D)$ and $\nu : X \to M$:
  $traces([\![T]\!]_\nu) \not\subseteq traces([\![P]\!]_{\emptyset:\emptyset\to\beta(M)})$
- color(T) = yellow otherwise.

From terms to stimuli and observations

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

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

- an alphabet $\mathcal{A}$ of primitive events
- a mapping $\|...\| : \mathcal{A} \longrightarrow T_\Sigma$
- a direction $D : \mathcal{A} \longrightarrow \{ts2sut, sut2ts\}$.

Test experiment with evaluation "on the fly"

. . .

Red test case: "observation a expected"

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

. . .

Test verdict

Assumption: SUT is a “deterministic” system.

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

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

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

$$process(SUT).$$

- 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

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

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

$inc \to a_1 \to inc \to a_2 \to inc \cdots \to a_n \to pass \to a \to fail \to Stop,$

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

Test execution

$$Execution^{Sp}_{process(SUT)}(T) =$$

$$(process(SUT) \,|[\, \alpha(Sp) \,]|\, T) \setminus \alpha(Sp)$$

Characterization theorem

$$Sp \rightsquigarrow_T process(SUT)$$

iff

for all tests $T \in Exhaust_{\mathcal{T}}(Sp)$ and

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

$$last(t) \neq fail.$$

Relating the two approaches (Work in progress)

Restrictions

1. Data in Csp-Casl: primitive events, e.g.

```
spec Alphabet_A =
   free type s_A ::=  a_1 | a_2 | ... | a_n
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:

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

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

$$R \leadsto_{\mathcal{T}} T \setminus \{inc, pass, fail\}.$$

Test verdict: "From TK/MR/HS to MCG/AC"

TK/MR/HS approach

Let $T \in Exhaust_{\mathcal{T}}(Sp)$.

Let the execution of $T \setminus \{inc, pass, fail\}$ at the SUT yield "pass" for some PCO with $\mathcal{A} = \alpha(P)$.

Test verdict: "From TK/MR/HS to MCG/AC"

TK/MR/HS approach

Let $T \in Exhaust_{\mathcal{T}}(Sp)$.

Let the execution of $T \setminus \{inc, pass, fail\}$ at the SUT yield "pass" for some PCO with $\mathcal{A} = \alpha(P)$.

MCG/AC approach

Then one can argue:

For all $t \in traces(Execution^{Sp}_{process(SUT)}(T))$: $last(t) \neq fail.$

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

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”

Test case excerpt: XML encoding


```
...
<?xml version="1.0" encoding="UTF-8"?>
<ep2:message xmlns:ep2="http://www.eftpos2000.ch" specversion="0400">
  <ep2:actcfgdataack msgnum="2634">
    <ep2:AcqID>0000000004</ep2:AcqID>
    <ep2:TrmID>TERM1234</ep2:TrmID>
  </ep2:actcfgdataack>
</ep2:message>

<?xml version="1.0" encoding="UTF-8"?>
<ep2:message xmlns:ep2="http://www.eftpos2000.ch" specversion="0400">
  <ep2:sessend msgnum="2635">
     <ep2:AcqID>0000000004</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 (?).