# Testing Networked Systems: Theory and Practice

Mohammad Mousavi



#### A discipline of testing is:

# extremely important, and can be rigorous, too.

#### Part 1:

#### (Model-Based) Testing Fundamentals: Theory and Practice

# Based on joint work with:

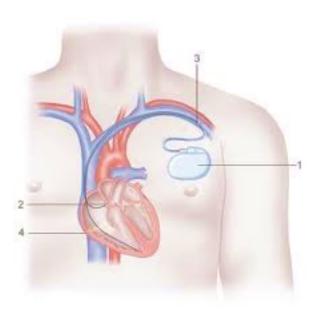
- Hamid Reza Asaadi (U Tehran, now at Stony Brook U),
- Rachid Kherrazi (Philips Healthcare, now at Promedico),
- Ramtin Khosravi (U Tehran),
- Mehmet Kovacioglu (Philips Healthcare, now at Credit Suisse),
- Neda Noroozi (TU Eindhoven, now at Nspyre),
- Mahsa Varshosaz (Halmstad U),
- Vivek Vishal (Philips Healthcare, now at ASML), and
- Tim Willemse (TU Eindhoven).

## Testing:

#### Why? What? How?

#### Why? Software at Your Heart

Software glitches in pacemakers "Company said it has not received any reports of deaths or clinical complications resulting from the glitch, which appears in about 53 out of every 199,100 cases."



[Killed by Code, 2010]

#### Why?

#### Software at Critical Infrastructure

... a glitch caused more than 3,200 US prisoners to be released early. The software calculates a prisoner's sentence depending on good/bad behaviour and was introduced in 2002.



Photo by Thomas Hawk @ Flicker



#### Why? Software at Your Car

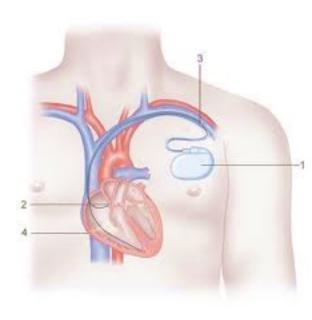
Over the past two years Nissan has been recalling airbags adding up to over 1 million cars ... due to a glitch in the airbag's sensory detectors. There has been a reported two accidents due to this software failure.



Photo from Wikipedia

#### Why? Software at Your Heart

At least 212 deaths from device failure in five different brands of implantable cardioverterdefibrillator (ICD) according to a study reported to the FDA



[Killed by Code, 2010]

## Why? Bugs (Faults): Facts of Life

"Coders introduce bugs at the rate of 4.2 defects per hour of programming. If you crack the whip and force people to move more quickly, things get even worse."

[Watts Humphreys]



#### Why? Bugs (Faults): Facts of Life

"Cost of software faults in 2016: 1100 Billion USD,

Number of people affected by software faults: 4.4 Billion people."

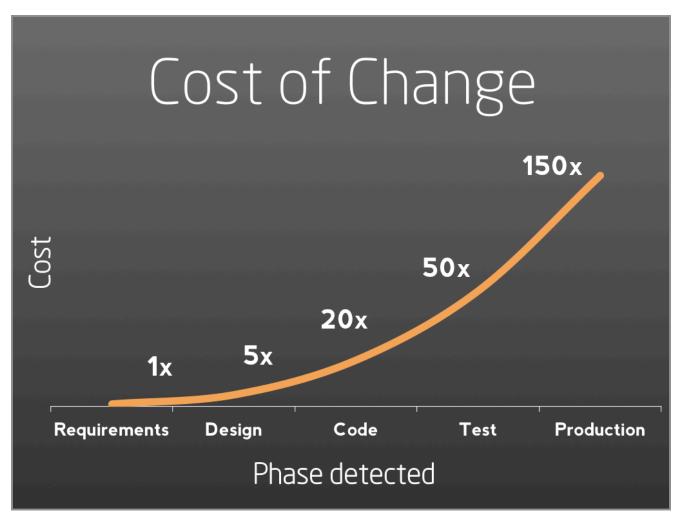


[Tricentis, Software Fail Watch, 2016]

Amounts to over 50% of the world's population

Photo Copyright: Tricentis

#### Why? Boehm's Curve



## What? Faults, Errors, Failures

- Fault: incorrect implementation:
  - commission: wrong implementation
  - omission: forgotten implementation (the more difficult one)
- Error: incorrect system state
- Failure (anomaly, incident) : visible error in the behavior

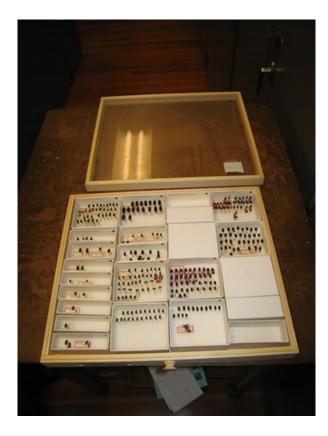


Photo from Wikipedia

Spec: inputs an integer, and outputs 2\*i<sup>3</sup>

Implementation: #include <iostream> #include <math.h>

```
int main() {
    int i;
    cin >> i;
    i = 2 * i;
    i = pow(i, 3);
    cout << i;
    return 0;
}</pre>
```

- 1. cin >> i;
- 2. i = 2 \* i;
- 3. i = pow(i, 3);
- 4. cout << i;
- Conceptual mistake: confusing the binding power of operators
- Fault: Statements 2 and 3 are in the wrong order
- Error: State of the program after line 3 may have the wrong value for i.
- Failure:
  - Test-case: input 1, expected output 2.
  - Actual execution: input 1 ... output 8!

#### What? Testing

Planned experiments to:

- reveal bugs (turn faults into failures, test to fail), "Testing can show the presence of bugs, but not their absence." [Dijkstra]
- 2. gain confidence in software quality (test to pass)

#### What?

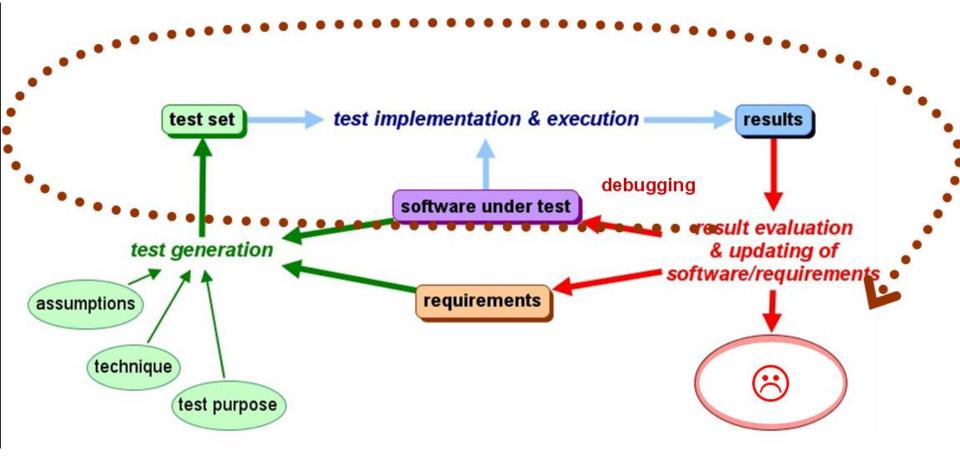
**Testing: Validation and Verification** 

- Validation: Have we made the right product; compliance with the intended usage (often: usercentered, manual process, on the end product)
- Verification: Have we made the boduct right; compliance between artice of different phases (often: artifact-driver). formalizable and mechanizable process among all phases)

#### How? Test-Case, Test-Suite

- Test-Case: a pair of
  - inputs (e.g., running environment, input values or pre-conditions, timing of events) and
  - expected outputs (e.g., concrete output values or symbolic properties input and output)
- Test-Suite: a set or list of test-cases

#### What? Testing



# What? Levels of Testing Subsystems

Units

#### Pasargad School 2018

System Testing (Acceptance, User)

Selenium, EyeAutomate GUI Model Testing, Automated GUI Testing

Integration Testing

jUnit, Mockito Dependency Injection Mocking

Unit Testing jUnit, QuickCheck, EvoSuite Test-Driven Dev. Equivalence Partitioning, Decision Tables, Classification Trees

Test-Driven Development of a SimpleStack Class (in Java)

Step 1:

}

Fix the signature of the class

package example.stack; public class SimpleStack { public boolean isEmpty(); public int pop(); public void push(int item)

Test-Driven Development of a SimpleStack Class (in Java)

Write class invariants and a few properties for each method:

#### /\*

**Pre-Condition:** True (can be called in all states, with all inputs) **Returns:** 

- true on an empty initialized SimpleStack

- false on a SimpleStack on which more successful "push"es are performed than "pop"s

State remains unchanged in both cases \*/

Test-Driven Development of a SimpleStack Class (in Java)

Step 3:

Start with a test:

@Test

```
public void testNewStackIsEmpty() {
   SimpleStack stack = new SimpleStack();
   Assert.assertTrue("New stack should be empty!",
        true == stack.isEmpty());
}
```

Test-Driven Development of a SimpleStack Class (in Java)

Step 4:

Test and check if any test fails.

If so, write the minimal amount of code to pass the test(s):

public class SimpleStack {
 public boolean isEmpty() {
 return true;
 }

Test-Driven Development of a SimpleStack Class (in Java)

Step 5:

Refactor the code if needed.

Repeat steps 3 to 5 until the requirements are covered.

Test-Driven Development of a SimpleStack Class (in Java)

Step 3:

Start with a test:

@Test

}

public void testNewStackPush() {

SimpleStack stack = new SimpleStack();

int item = 1;

stack.push(item);

Assert.assertFalse("Stack shouldn't be empty after a push!",

stack.isEmpty());

Step 4:

Test and check if any test fails.

If so, write the minimal amount of code to pass the test(s):

```
public class SimpleStack {
    boolean empty = true;
    public void push(int item) {
        empty = false;
    }
    public boolean isEmpty() {
        return empty;
    }
```

Test-Driven Development of a SimpleStack Class (in Java)

Step 5:

Refactor the code if needed.

Repeat steps 3 to 5 until the requirements are covered.

#### What We Do <u>Not</u> Cover: Test Management and Policy



What We Do <u>Not</u> Cover: Alternatives to Testing

- Model Checking: test the state-space (all executions) for formally specified properties
  - + rigorous analysis, push-button technology
  - not (yet) scalable to very large systems (state-space explosion)

#### What We Do <u>Not</u> Cover: Alternatives to Testing

• **Static Analysis:** test abstract properties without running the program, e.g., division by zero and empty/unspecified cases

- + automatic and scalable for generic and abstract properties;
- + existing powerful tools;
- involves approximation (true negatives and false positives);
- complicated (may involve theorem proving) for concrete and specific properties (proving the abstraction function to be "correct")

# Theory:

#### Introduction to Model-Based Testing

#### **Model**-Based Testing

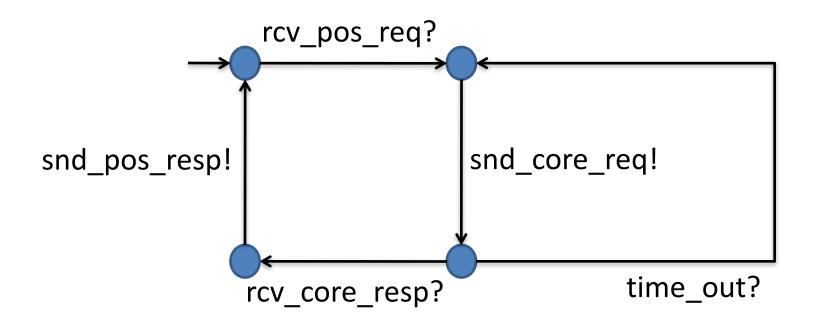
- Abstractions from reality
- Separating different concerns
- Approximating system behavior and / or its environment





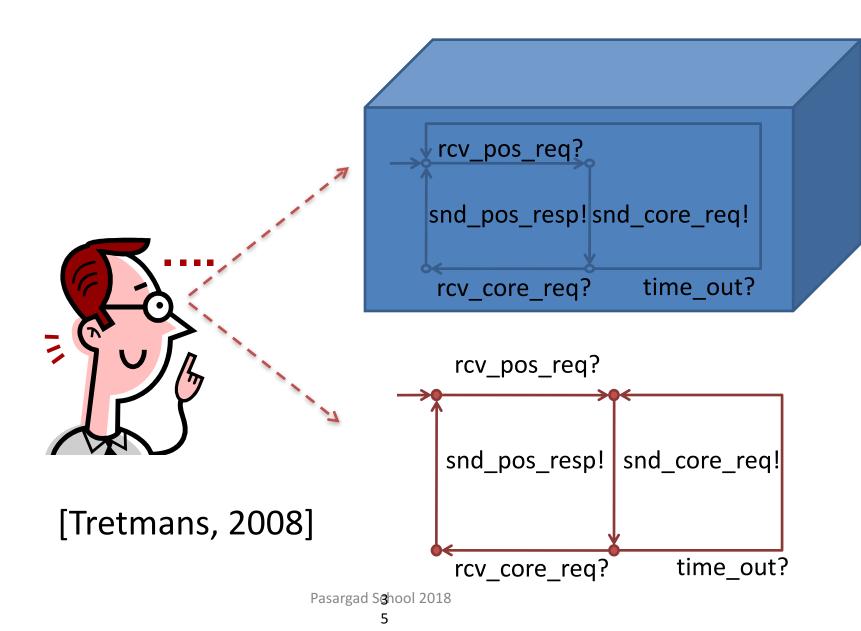
#### **Model**-Based Testing

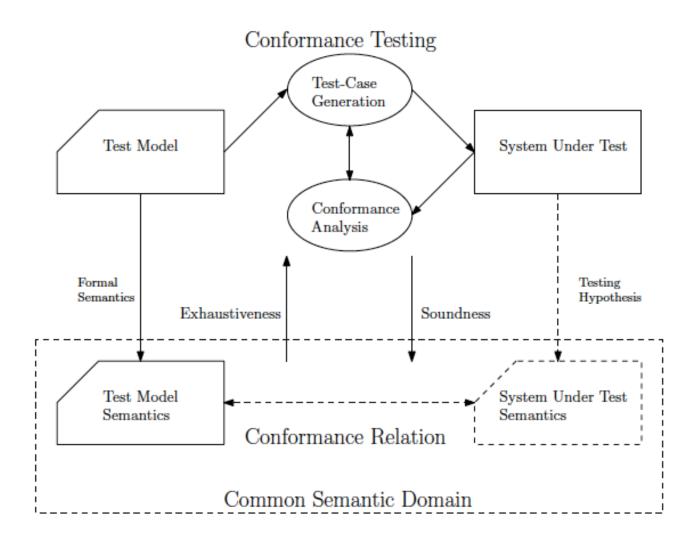
Modeling the desired behavior (system) / possible interactions (environment)



Pasargad School 2018

#### Model-Based Testing

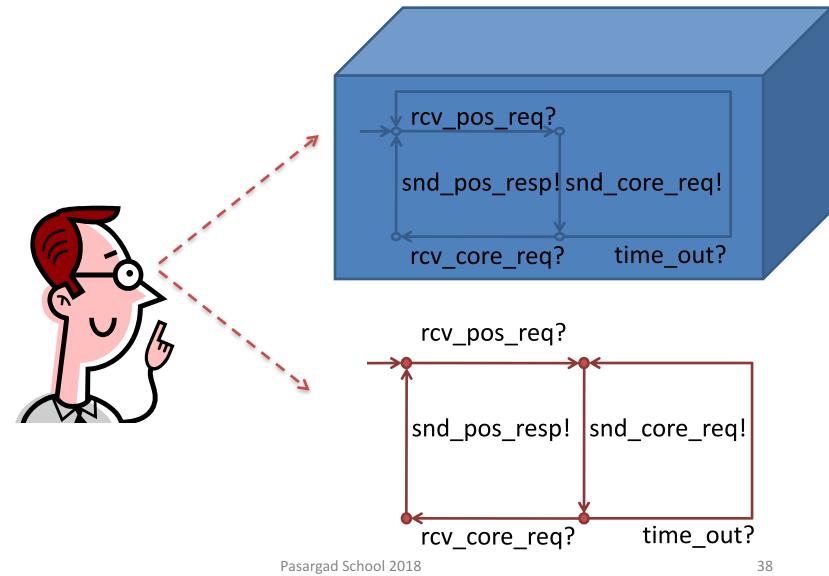


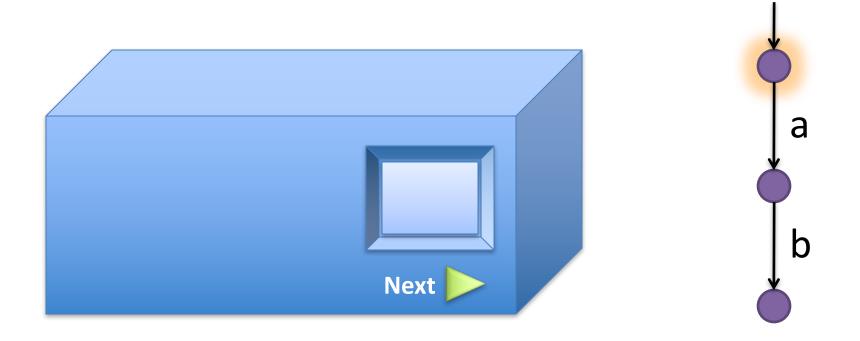


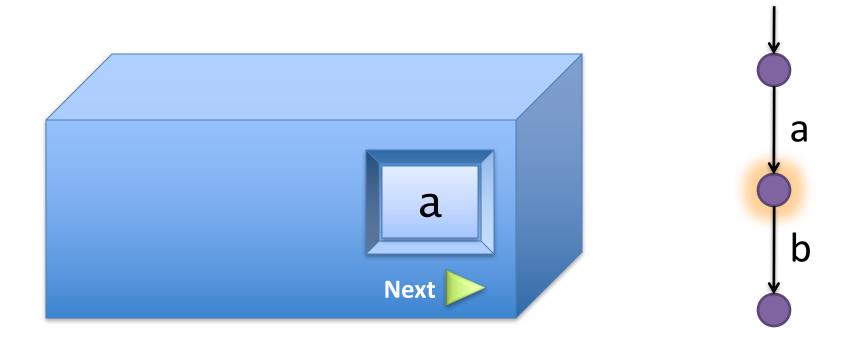
## Theory:

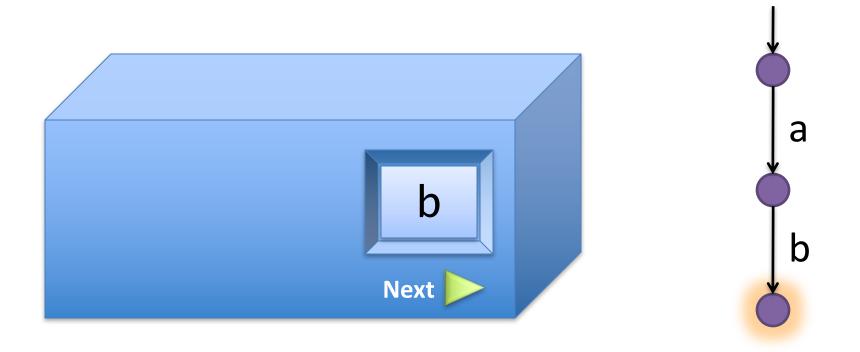
#### Testing from Labeled Transition Systems

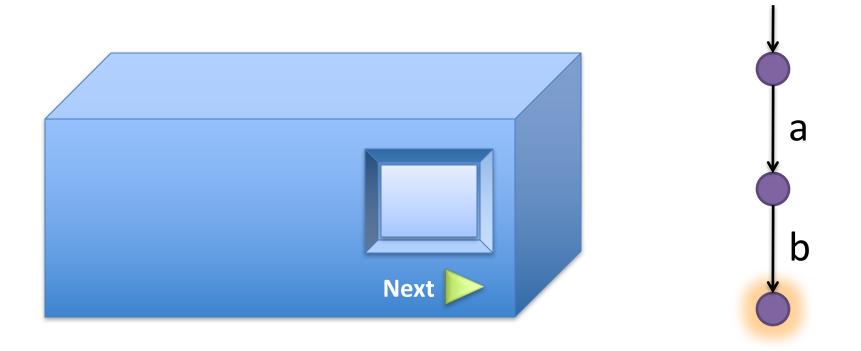
## Equivalence By Observation







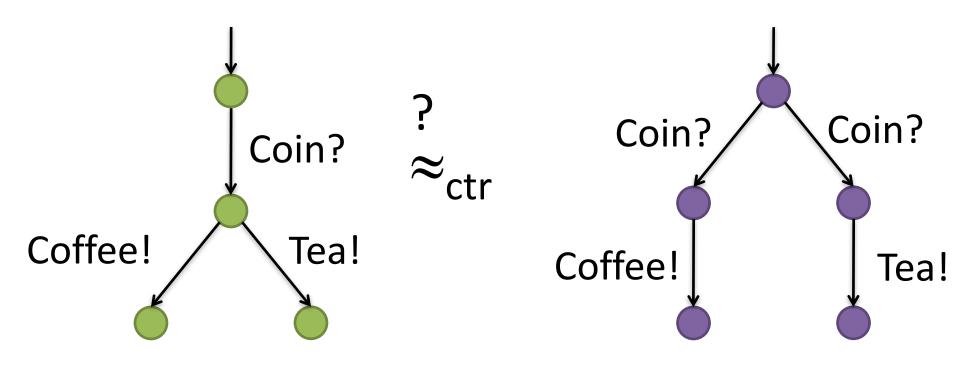




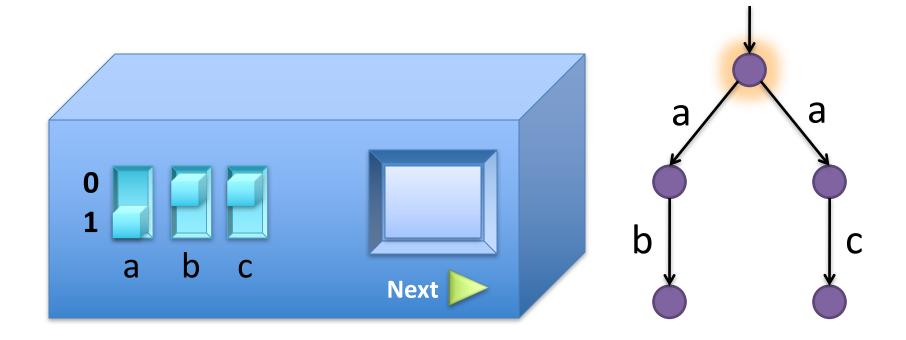
### **Completed Trace Equivalence**

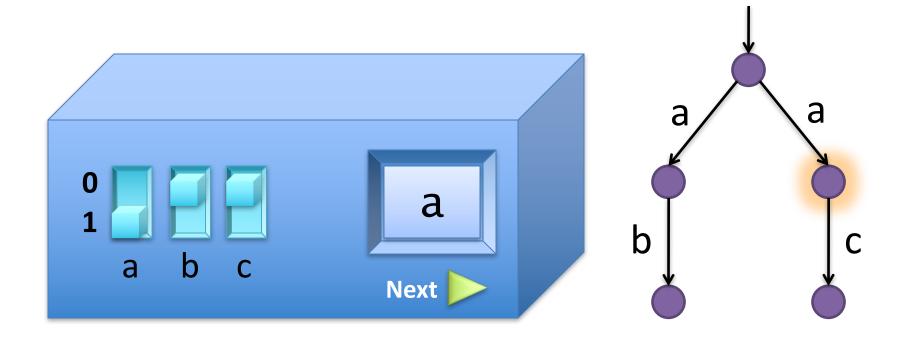
$$I \approx_{ctr} S$$

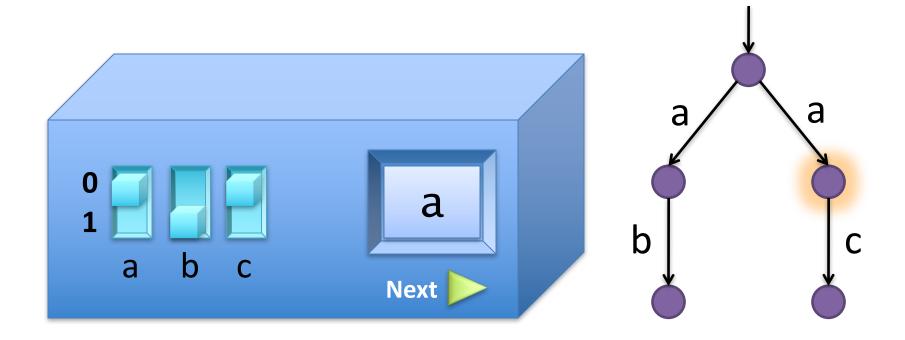
# traces(I) = traces(S) c\_traces(I) = c\_traces(S)

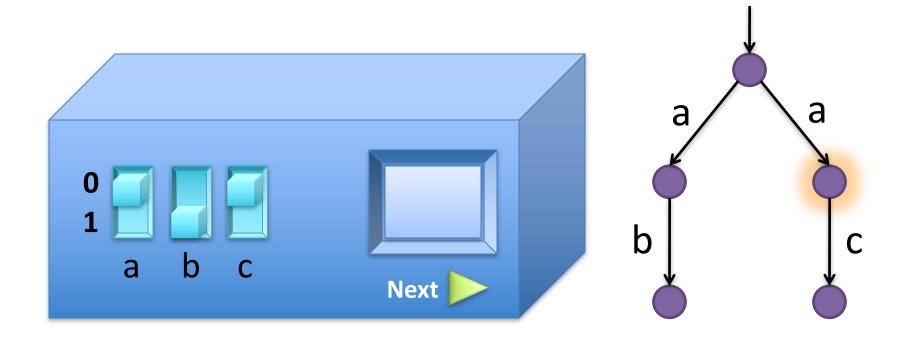


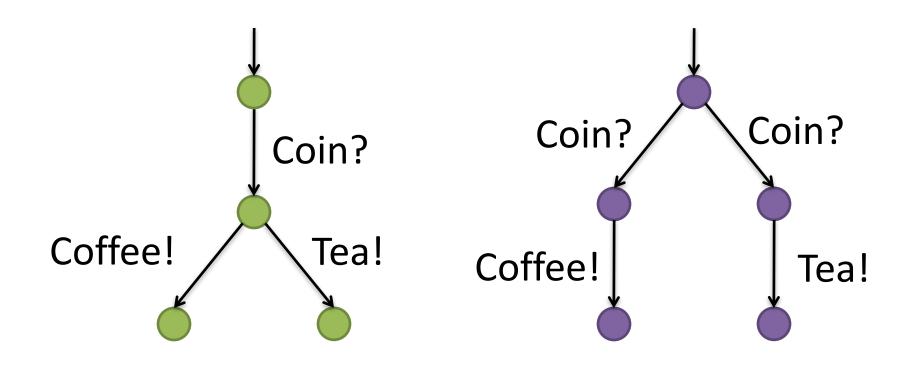
Moral of the story: trace equivalence is too coarse for conformance testing open systems

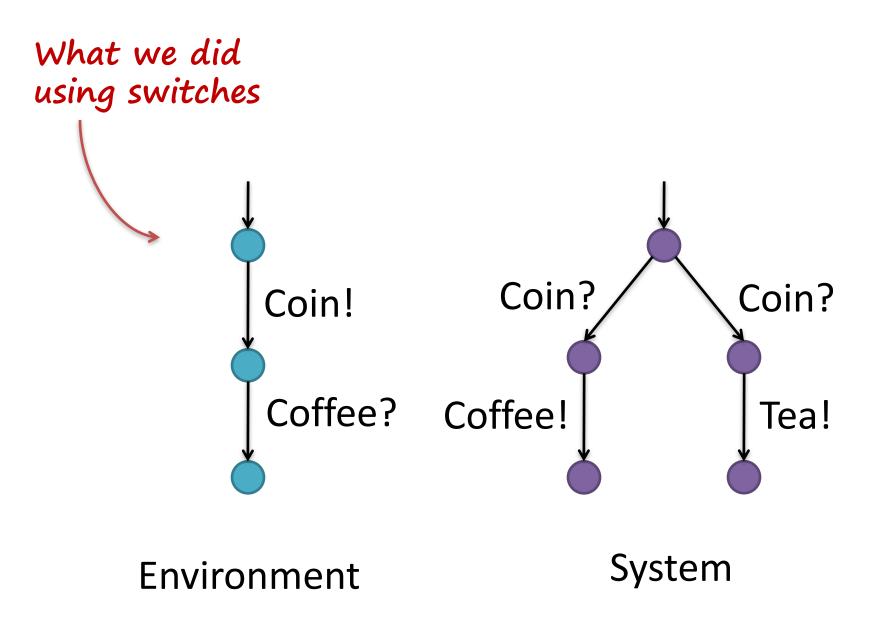




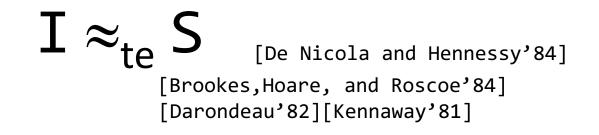






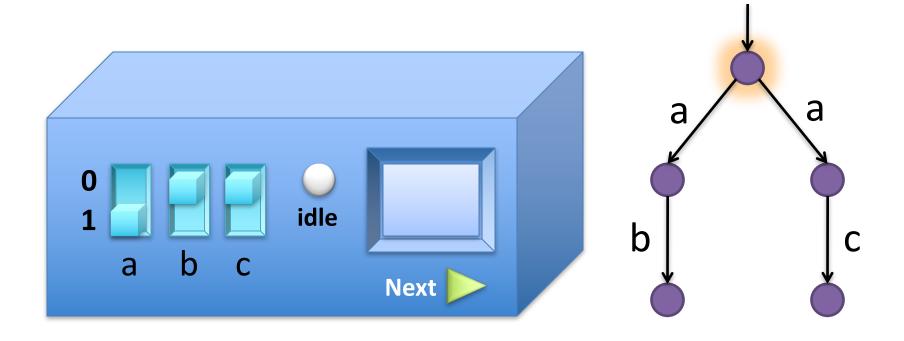


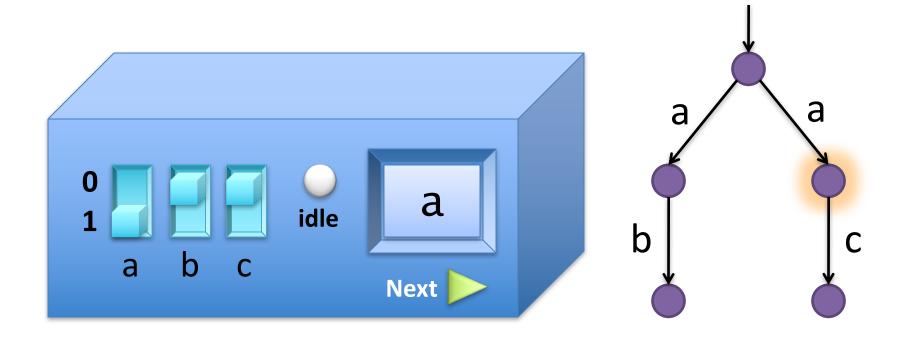
## **Testing Equivalence**

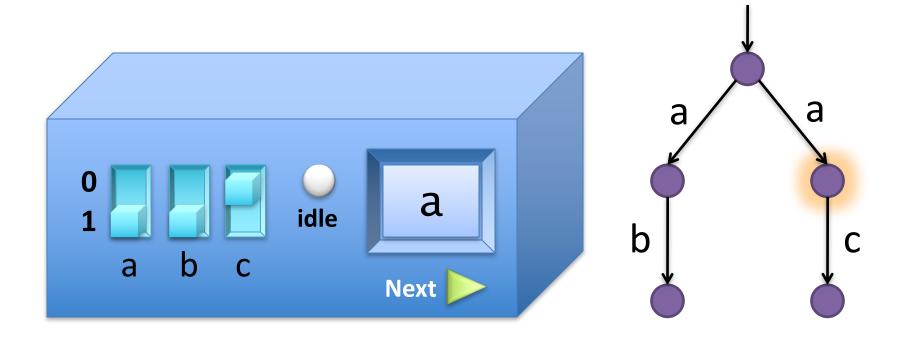


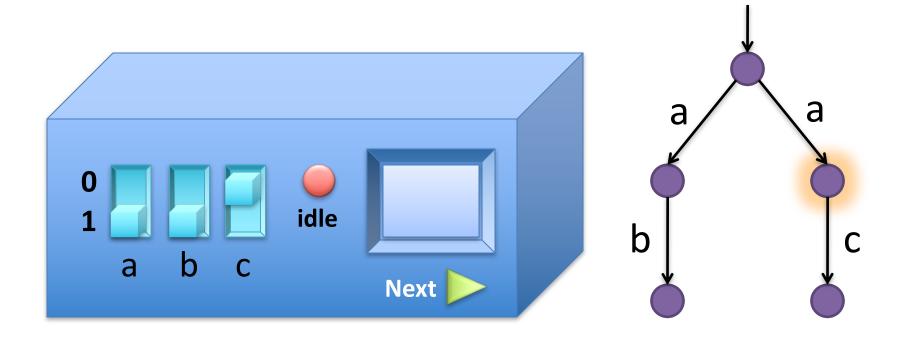
*for every environment* E:

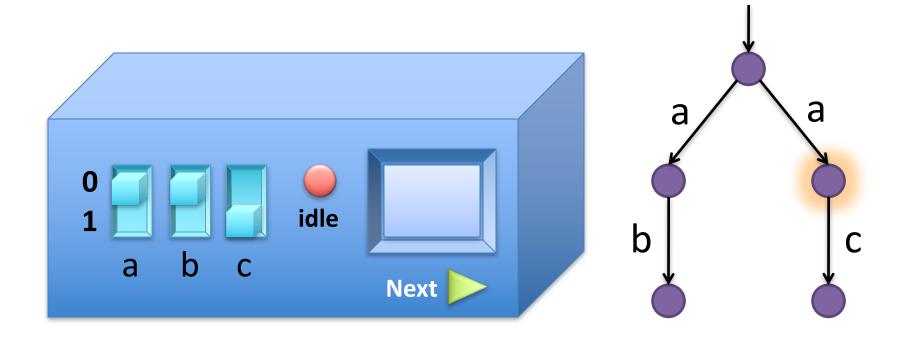
# traces(E || I) = traces(E || S) c\_traces(E || I) = c\_traces(E || S)

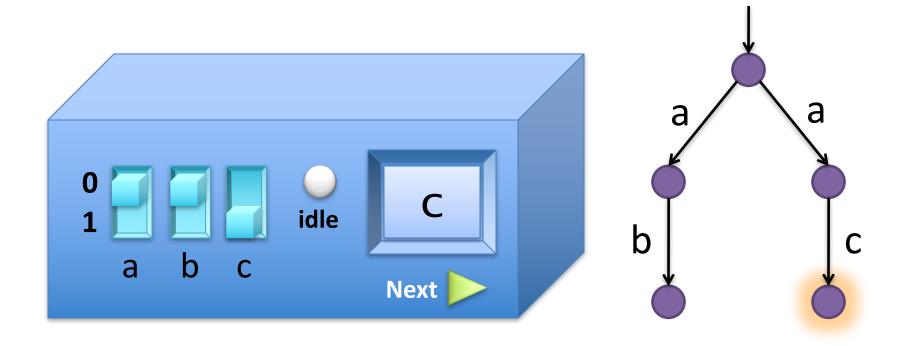


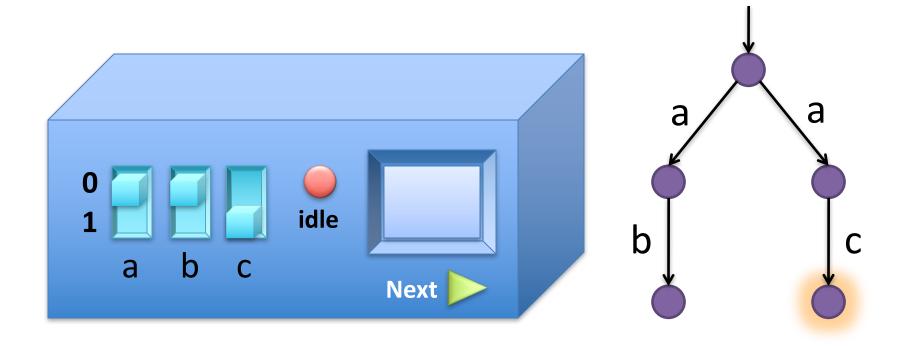


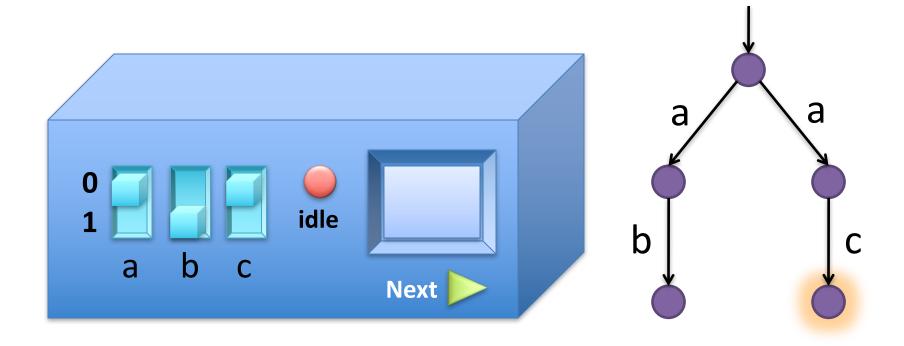


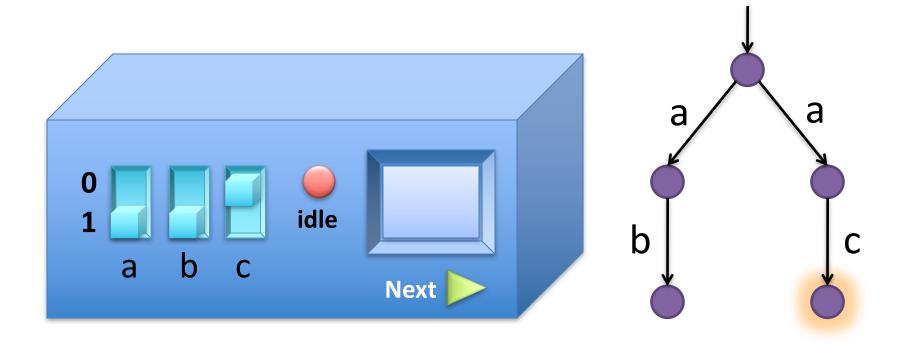


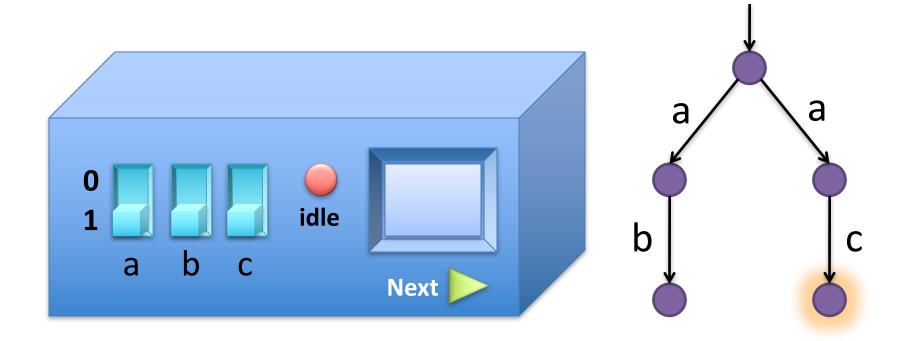




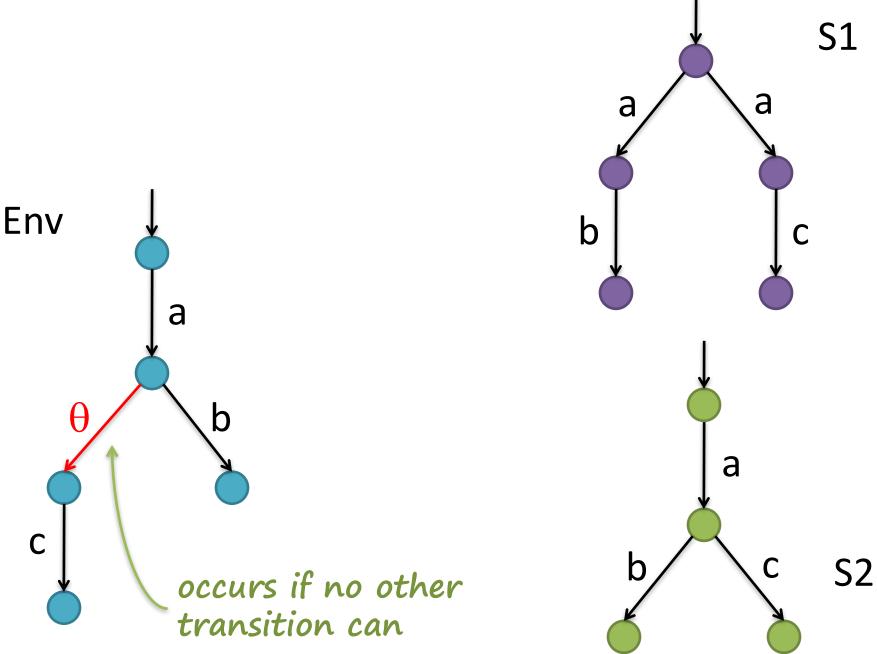




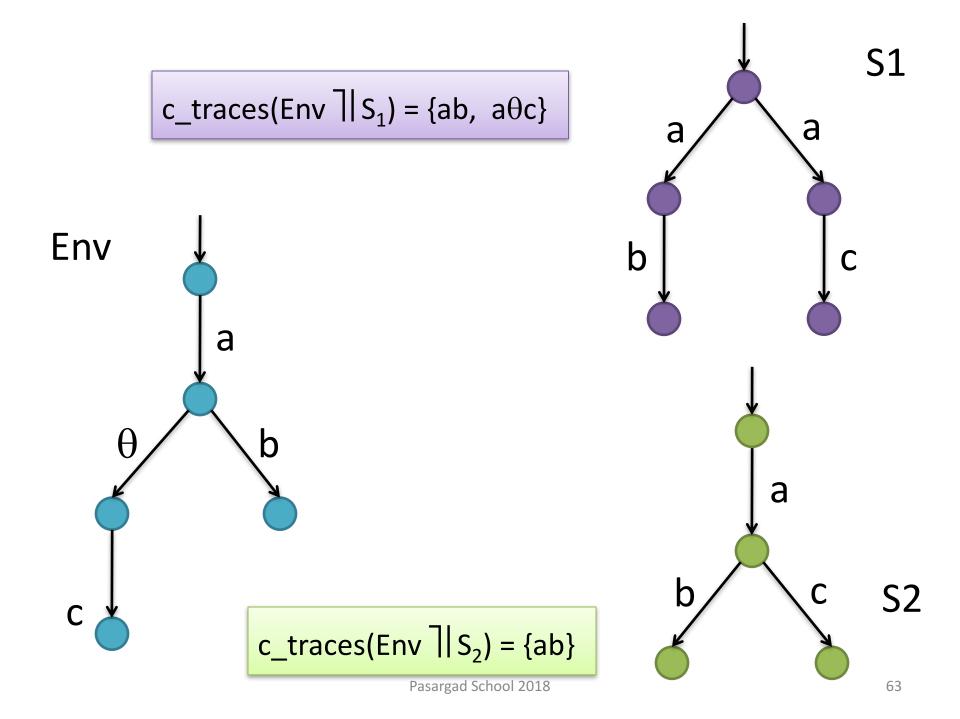




#### Now, we know that we have completed a trace.



Pasargad School 2018

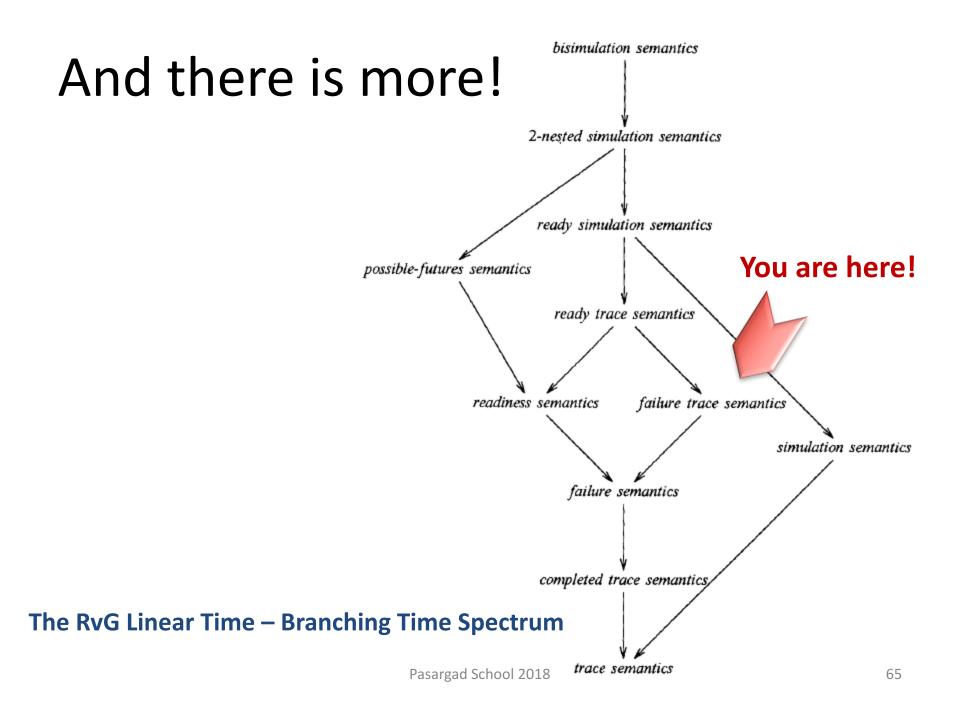


## **Refusal Equivalence**

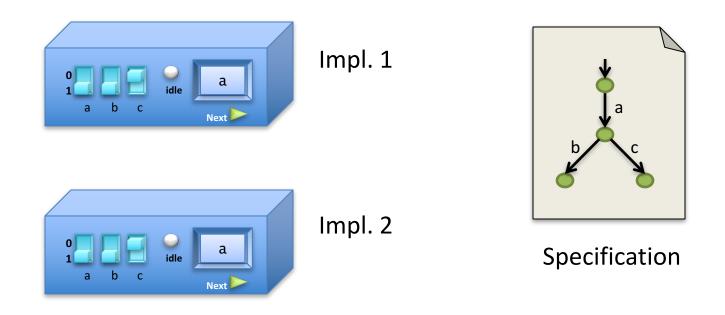
$$I \approx_{rf} S$$
 [Philips'87]

#### *for every environment* E:

# traces(E □ I) = traces(E □ S) c\_traces(E □ I) = c\_traces(E □ S)



#### Defining specifications at a higher-level



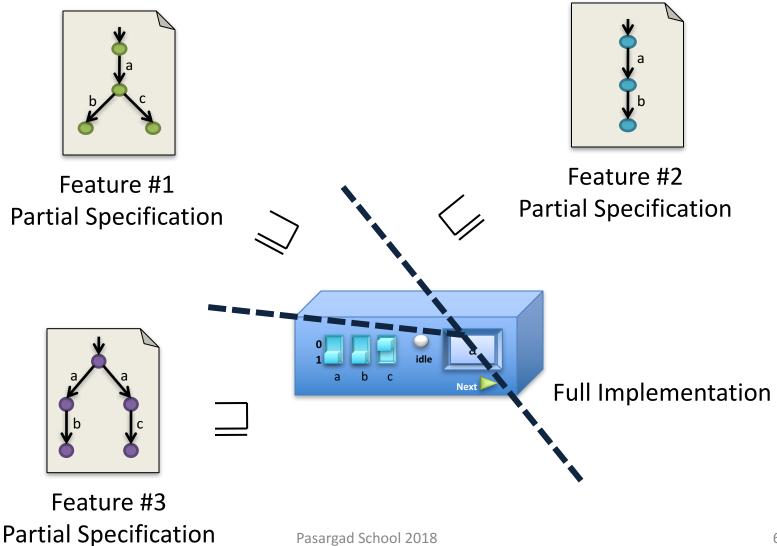
### **Testing Pre-order**

$$\mathbf{I} \sqsubseteq_{\mathsf{te}} \mathbf{S}$$
 [De Nicola and Hennessy'84]

#### *for every environment* E:

# traces( $E \parallel I$ ) $\subseteq$ traces( $E \parallel S$ ) c\_traces( $E \parallel I$ ) $\subseteq$ c\_traces( $E \parallel S$ )

## **Restriction to Specification**



#### **Restriction to Specification**

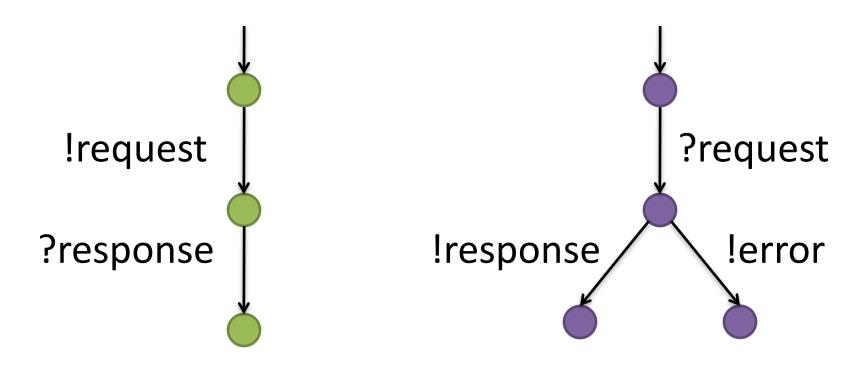
## I conf S [Brinksma87]

#### for every environment E:

traces(E || I)  $\cap$  traces(S)  $\subseteq$  traces(E || S) c\_traces(E || I)  $\cap$  traces(S)  $\subseteq$  c\_traces(E || S)

# I/O Transition Systems

Distinguishing between input and output actions



#### Pre-orders on I/O transition systems

- The same notions apply here
  - I/O test pre-order
  - I/O refusal pre-order

$$I \sqsubseteq_{ior} S$$

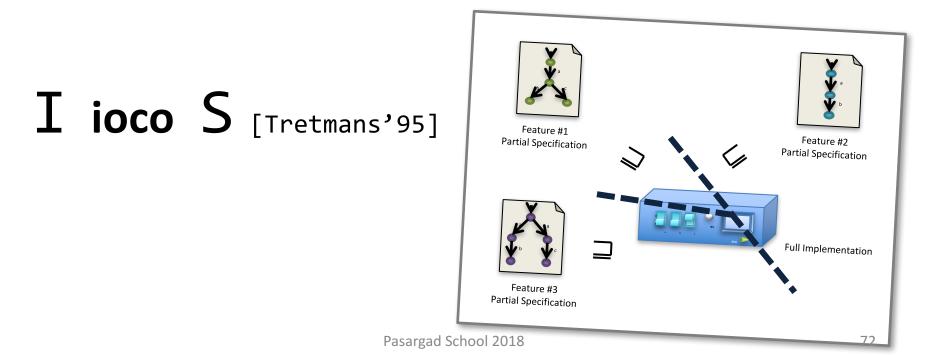
for every environment E:

traces(E  $\dashv$  I)  $\subseteq$  traces(E  $\dashv$  S) c\_traces(E  $\dashv$  I)  $\subseteq$  c\_traces(E  $\dashv$  S)

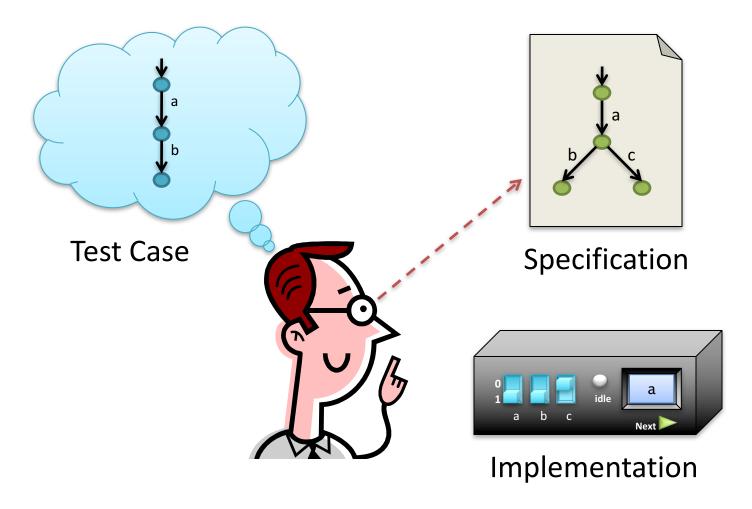
# I/O Conformance

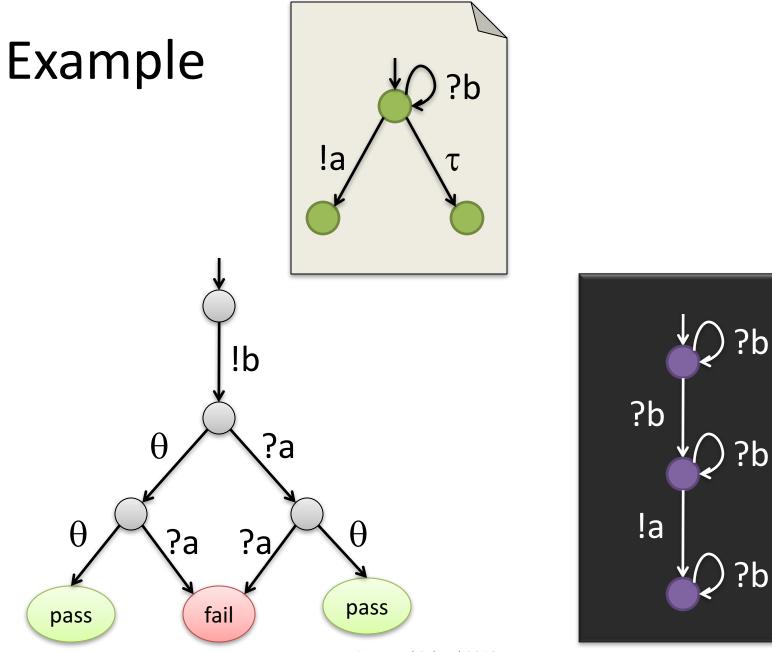
Informally,

#### I/O Conformance = I/O Refusal restricted to specification traces



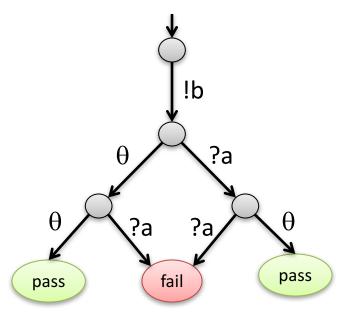
### Black-box testing for ioco





### ioco Test Cases

- I/O transition systems
- The only terminal states: (
- Reversed I/O actions
- Special action  $\boldsymbol{\theta}$
- Finite and deterministic



fail

and

pass

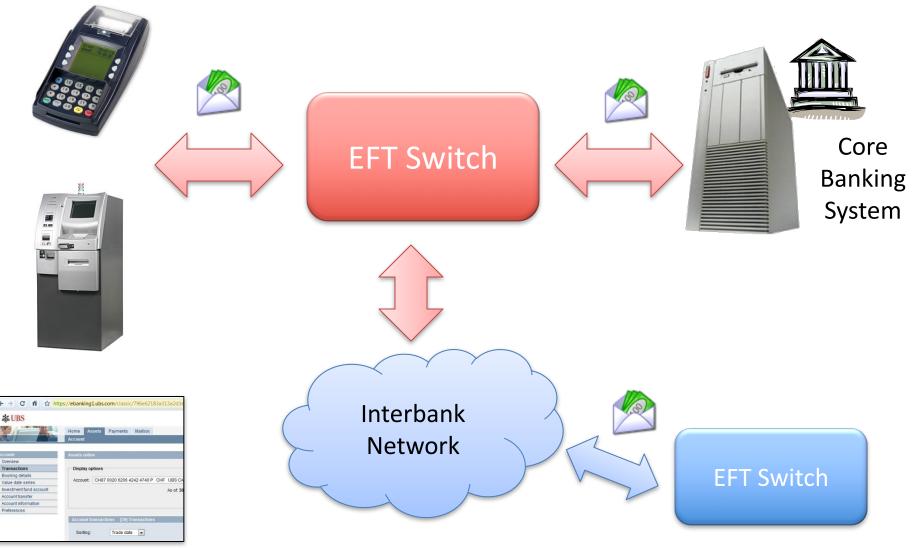
## Automatic Test Case Generation

- Init:
  - Generate an initial state
- Recursion:
  - At each point in the recursion choose nondeterministically between:
    - 1. Stopping the recursion
    - 2. Supplying an input
    - 3. Observing an output (one transition per output action)

### Practice:

### EFT and X-Ray Machine Cases

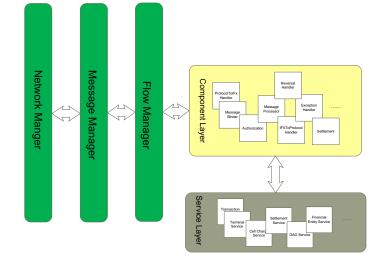
# Electronic Funds Transfer (EFT)



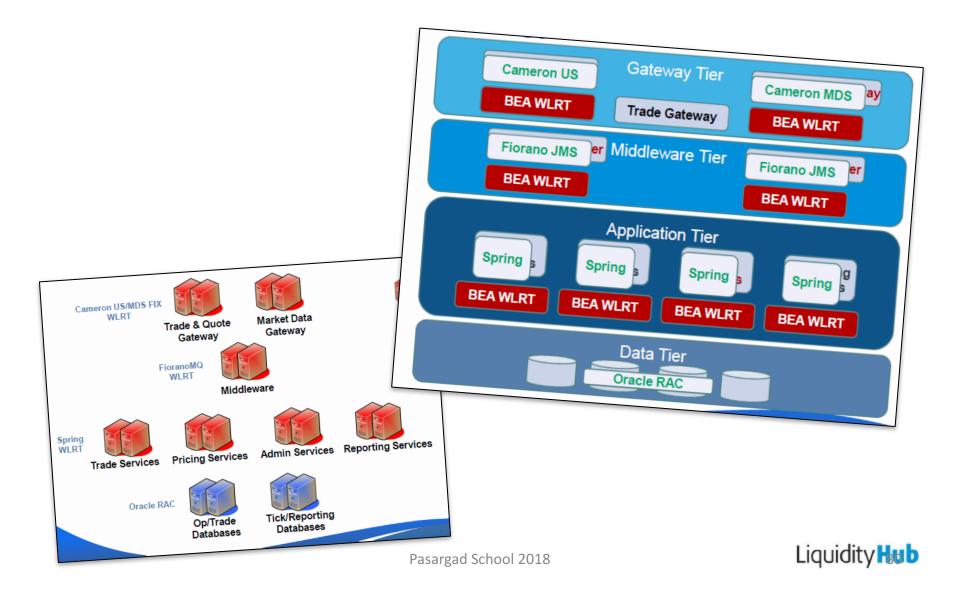
# The System Under Test

- An Operational EFT Switch, developed at Fanap
- Java Application (~100 KLOC)
- Extensive use of Java frameworks

 Already tested, but not with a disciplined view on concurrency



### **Complex Architectures**



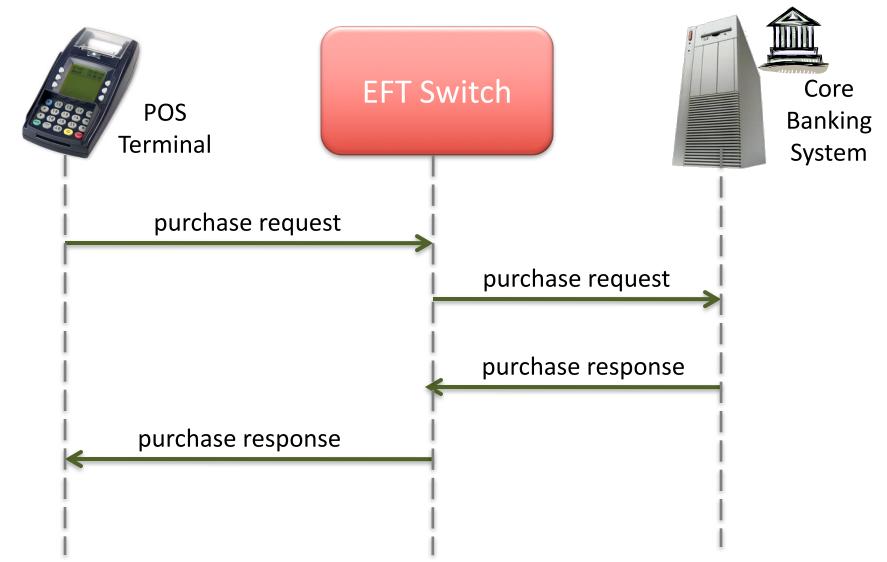
Object Cache	Security	Transaction Management	
Session Management	Domain Logic	Persistence Frameworks	C
Thread Pool	Integration Services	Logging	



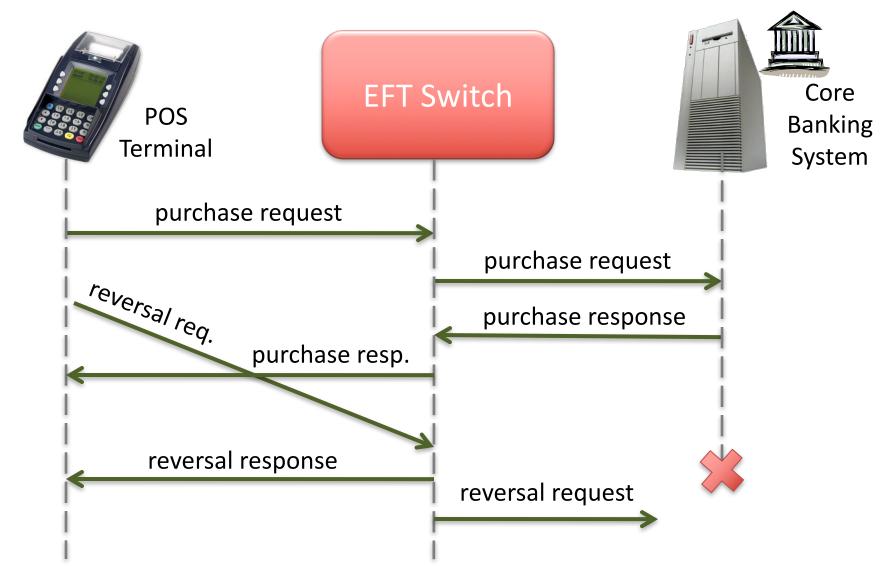
Object Cache	Security	Transaction Management	
Session Management	Domain Logic	Persistence Frameworks	
Thread Pool	Integration Services	Logging	



### **Example Scenarios**

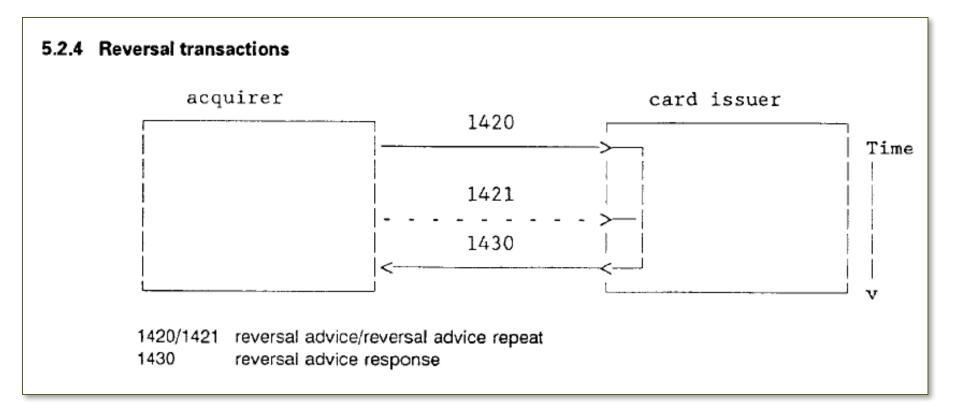


### **Example Scenarios**

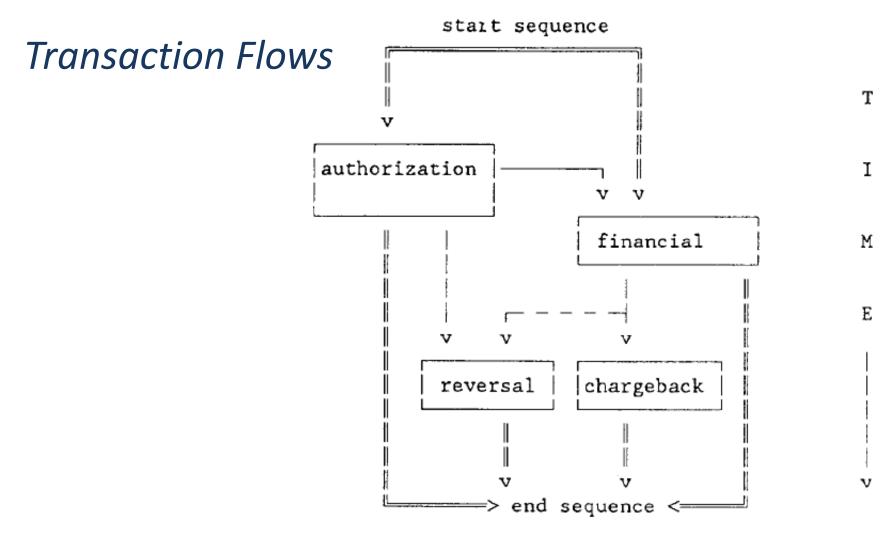


### ISO 8583 Standard

### Message Flows



### ISO 8583 Standard



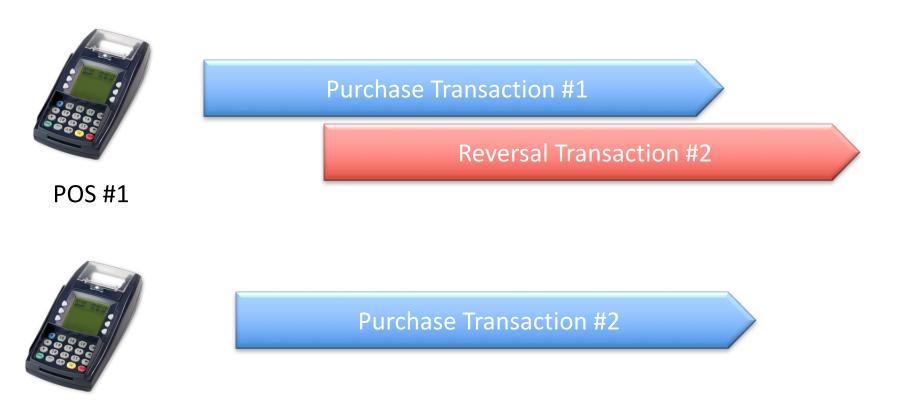
### ISO 8583 Standard

### **Business Rules**

b) The amount, transaction data element in a reversal advice or notification shall contain the amount to be reversed and shall be less than or equal to the original amount as shown in table 3.

c) Whenever the acquirer times out waiting for a response to an authorization or financial transaction request or advice, a reversal advice or notification of the transaction shall be sent (see 5.2.12).

### **Concurrent Transactions**



POS #2

# Our Method

• Using Model-Based Testing

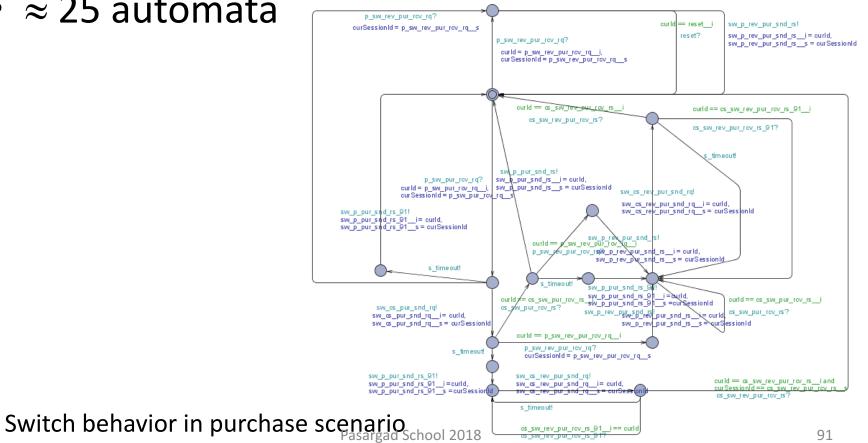
- Using a set of integrated tools
  - Generating and running test cases
  - Logging and prioritizing test cases
  - Measuring test coverage
  - Testing business rules

### Back to the real world!

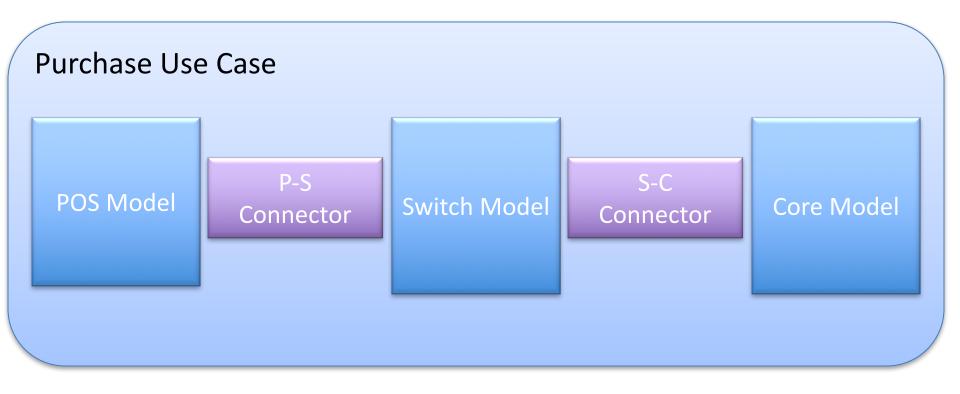


# Switch Specification

- Modeling in UPPAAL
- $\approx 25$  automata

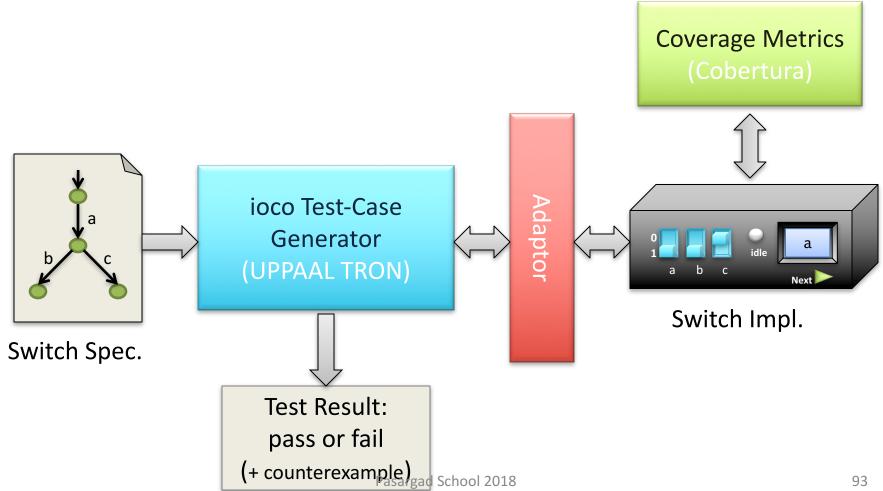


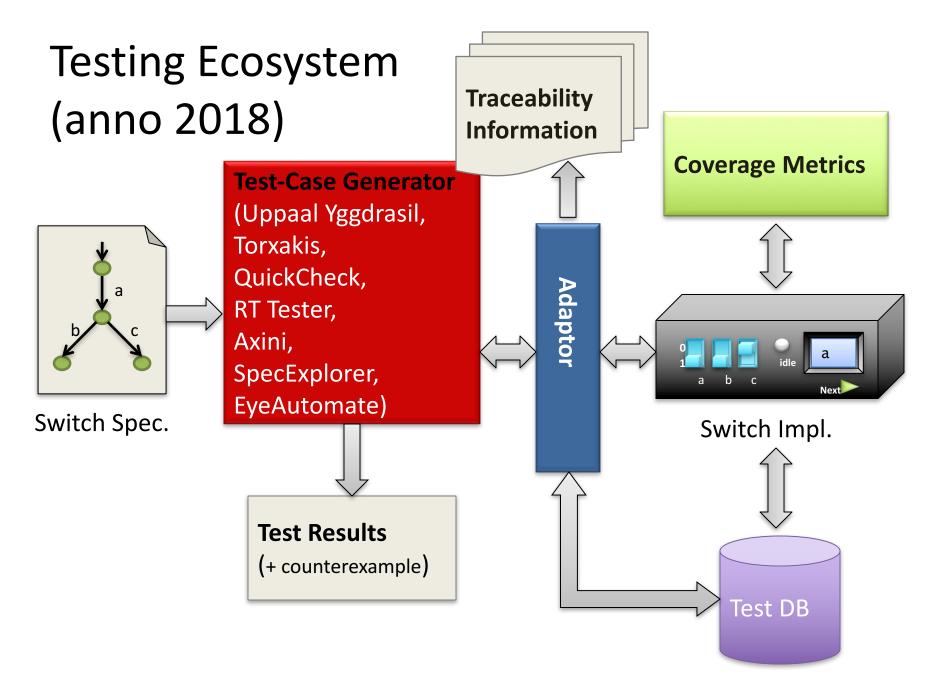
### **Specification Structure**



#### Four uses cases till now

## **Testing Ecosystem** (anno 2008)





## Measuring Coverage

cobertura-html/cobertura-html/index

### Using Cobertura code coverage tool (now use EclEmma)

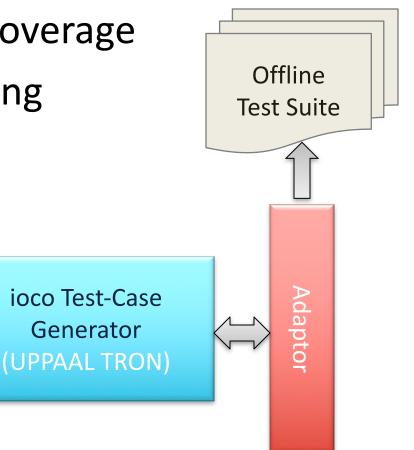
C ☆ file///C/Users/HamidReza/Desktop/cobertura-html/cobertura-html/index.html          W tinks W Wk8books O Online tutorials for L.       Table of Contents O CS656-CS656 F.	anfia ass.components.handikara ass.components. handikara mateciali modina.impl distriction ass.components. distriction ass.components. distriction dis		Line Coverage         "           100%         3/2           100%         3/2           100%         3/2           100%         3/2           100%         3/2           20%         3/2           20%         3/2           20%         3/2           20%         3/2           20%         3/2           6%         4/2           6%         3/2           6%         3/2           6%         3/2           9%         3/2           9%         3/2           9%         3/2           9%         3/2           9%         3/2           9%         3/2           9%         3/2           9%         3/2           9%         3/2           9%         3/2           9%         3/2           9%         3/2           9%         3/2           9%         3/2	50% 20% 30% 26% 25% 43% 8/A	N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A	Complexity Complexity 3. 3. 1. 1. 2. 4. 4. 2. 3.
<ul> <li>← → C ☆ file///C/Users/HamidReza/Desktop/cobertura-html/cobertura-html/index.html</li> <li>Costantial</li> <li>WikeBooks O Omine tutenels for L.</li> <li>Table of Contents O Costo Cost</li></ul>	anfia ass.components.handikara ass.components. handikara ass.components. handi material. handi den ass. handi den ass. handi d	1 4 3 2 2 1 1 1	00%         28/47           76%         23/37           23%         23/37           76%         23/37           74%         25/20/33           70%         15/20/33           70%         15/20/33           70%         15/20/33           64%         16/12/23           65%         6/20/33           65%         4/20           55%         15/20/33           55%         4/6/33           55%         12/27/31           55%         12/27/31           55%         12/27/33           55%         12/27/31           55%         12/27/31           55%         12/27/31           55%         12/27/31           55%         12/27/31	62% NA NA 49% 54% 46% 54% 42% 20% 20% 50% 30% 22% 45% 45% 45% 45% 45% 45%	10/14 N/A N/A 82/345 132/370 33/44 15/22 34/100 4/2102 1/2 26/128 16/34 4/22 26/128 16/34 4/22 16/34 4/22 16/68	3. 1. 1. 1. 8. 1. 4. 2.
Clinics WW80oks Online tutorials for L. Table of Contents CSc656 - CSc556 F. U FAEG Course Page رقال تعديد Contents Online tutorials for L. Table of Contents CSc656 - CSc556 F. U FAEG Course Page (CSC656 F. U	ases.commonents.handlers estears? tetrars? fs fs mostocols.encodina.unel fs fs mostocols.handl.encodina thadi mail disastic.handl.encodina disastic.handl.encodina disastic.handl.encodina consocial.handl.encodina consocial.stat.com estears.com disastic.com disastic.com disastic.com estears.com disastic.com disastic.com disastic.com estears.com disastic.com disastic.com estears.com disastic.com estears.com disastic.com estears.com disastic.com estears.com disastic.com estears.com disastic.com estears.com es	1 4 3 2 2 1 1 1	76%         3227           75%         324           76%         350235           73%         74%           74%         350227           75%         35027           76%         35027           6%         16776           6%         16776           5%         666           7%         15225           6%         467           5%         6674           5%         6674           5%         6674           5%         6674           5%         6674           5%         6674           5%         6674           5%         67228           5%         6674           5%         6728           5%         6728           5%         6728           5%         6728           5%         6728           5%         6743           5%         6728           5%         6738           5%         6743           5%         6738           5%         6743	N/A 49% 60% 54% 46% 50% 20% 50% 20% 50% 22% 45% 45% 45% 45%	N/A N/A 12/255 222/375 35/64 11/922 54/100 4/25102 1/2 28/130 1/2 28/130 1/2 28/130 1/2 28/130 1/2 1/2 28/130 1/2 1/2 28/130 1/2 1/2 28/130 1/2 28/130 1/2 28/130 1/2 28/130 1/2 28/130 1/2 28/14 1/2 28/1 28/14 1/2 28/	3. 1. 1. 1. 8. 1. 4. 2.
Vinks WW80oks D Online tutorials for LL D Table of Contents D CSc656-CSc656-F_ U FMEd Course Page ج <sup>2</sup> () two TestIng course () TestIDN-TC accessing () TestIDN-TC accessi	ase, components estemats for for massibility that that that that observed. Insures estemats. An entit Amount the another and the another another another the another another the another another the another another the another another the another	1 4 3 2 2 1 1 1	744         524           744         55033           734         744,001           704         59027           684         141205           676         161224           677         15225           664         -467           653         -666           644         -32025           599         -3264           594         -167/33           594         -167/33           594         -167/33           594         -167/33           594         -167/33           594         -167/33           594         -167/33           594         -167/33           594         -167/33           594         -167/33           594         -167/33           594         -167/33           594         -117/54	8/A 49% 60% 54% 46% 20% 20% 50% 20% 22% 22% 45% 43% 43% 43%	N/A 82/566 322/370 35/64 15/22 54/100 1/2 28/100 1/2 28/100 1/2 28/100 1/2 1/2 1/2 1/2 1/4 1/4 4.5(00)	1 1 8 1 4 2
ackages     35     @ManyToOhe(casedde = CasedeType.ALL)     boomthin       all     \$JointCoLumciane = *sec_proff     boomthin       andional.ir.fanap.nanoswitch.feeDrofis     grivate SecurityProfile     boomthin       all     private SecurityProfile     boomthin       andional.ir.fanap.nanoswitch.intBikit     private SecurityProfile     boomthin       add     11     public SecurityProfile     boomthin       additional.ir.fanap.nanoswitch.minuteme     foreignoff     boomthin       additional.ir.fanap.nanoswitch.minuteme     foreignoff     boomthin       additional.ir.fanap.nanoswitch.minuteme     foreignoff     boomthin       additional.ir.fanap.nanoswitch.edutorics.com     foreignoff     boomthin       additional.ir.fanap.nanoswitch.edutorics.com     boomthin     boomthin       additional.ir.fanap.nanoswitch.edutorics.discom     foreignoff     boomthin       additional.ir.fanap.nanoswitch.edutorics.discom <td>strands de fan tesarde tesa</td> <td>1 4 3 2 2 1 1 1</td> <td>744         550/35           724         750/25           724         750/25           724         750/25           654         61/75           6574         657/25           6574         657/25           654         22/05           954         66/4           954         66/4           954         22/05           954         66/23           954         66/23           954         62/26           954         22/25           956         222/25           956         222/25           956         222/25           956         222/25           956         222/25           956         222/25           956         222/25           956         222/25           956         222/25           957         4/3           957         9/3</td> <td>49% 60% 34% 46% 54% 20% 20% 30% 22% 22% 22% 22% 22% 25% 45% (/A</td> <td>82/346 222/350 235/64 15/22 54/100 4/35102 28/136 28/136 28/136 18/36 12/22 18/66 11/4 45/100</td> <td>1 1 9 1 1 4 2</td>	strands de fan tesarde tesa	1 4 3 2 2 1 1 1	744         550/35           724         750/25           724         750/25           724         750/25           654         61/75           6574         657/25           6574         657/25           654         22/05           954         66/4           954         66/4           954         22/05           954         66/23           954         66/23           954         62/26           954         22/25           956         222/25           956         222/25           956         222/25           956         222/25           956         222/25           956         222/25           956         222/25           956         222/25           956         222/25           957         4/3           957         9/3	49% 60% 34% 46% 54% 20% 20% 30% 22% 22% 22% 22% 22% 25% 45% (/A	82/346 222/350 235/64 15/22 54/100 4/35102 28/136 28/136 28/136 18/36 12/22 18/66 11/4 45/100	1 1 9 1 1 4 2
ackages     *     38     ØManyGoom (caseder = CasederType, ALE)     monembla       jj     0     \$0 (GoodColum (case = * sec_prof)     monembla       jj     0     #ToreignAy (name* securityProf)     monembla       indionalinfanan.nanoswitch.fesPtofis     40     #ToreignAy (name* securityProf)     monembla       indionalinfanan.nanoswitch.fesPtofis     41     monembla     monembla       indionalinfanan.nanoswitch.fesPtofis     41     monembla     monembla       indionalinfanan.nanoswitch.refactoring     42     monembla     monembla       indionalinfanan.nanoswitch.refactoring     44     monembla     monembla       indionalinfanan.nanoswitch.refactoring     44     monembla     monembla       indionalinfanan.nanoswitch.refactoring     42     monembla     monembla       indionalinfanan.nanoswitch.setHomedi     43     monembla     monembla       indionalinfanan.nanoswitch.setHomedi     50     jiii.name name     HabilerSectionurityParameters     monembla       indionalinfanan.nanoswitch.setHomedi     51     0     this.parameters     (parameters = (parameters   mull)     parameters = nonembla     monembla       indionalinfanan.nanoswitch.setHomedi     51     0     this.parameters     (parameters   mull)     monembla       indionalinfanan.nanoswitch.setHomedi	contección activador a contección a contecci	1 4 3 2 2 1 1 1	234 24/03 2046 199/27 67% 142/29 67% 142/29 67% 142/29 66% 4/6 65% 4/6 65% 4/6 56% 4/6 56% 4/2 35% 16/7 36% 16/7 35% 16/7 35% 122/38 57% 0/3 57% 0/3	60% 34% 34% 34% 34% 20% 20% 30% 25% 25% 45% 45%	222/330 35/64 15/22 54/100 4/2502 28/130 18/36 18/36 18/36 18/36 15/60	
ck.dg/sb     39     \$JoinColumn(rame = *sec_prof")     incommentation       jjj     0     \$Foreign&perprof"     incommentation       junctionalinframe.necosynth.freeProfile     4     incommentation       junctionalinframe.neco	rde messace extensive.chanellanse extensive.chanellanse distantiv.chanellanse extensive.chanellanse distantiv.chanellanse distantive.chan	1 4 3 2 2 1 1 1	70% 39/772 60% 141/202 67% 163/241 67% 163/241 67% 22/731 9% 84/2 9% 94/2 9% 96/2 9% 94/2 9% 96/2 9% 96/2 9% 96/2 9%	34% 46% 54% 41% 20% 20% 30% 37% 26% 25% 43% 43%	35/64 15/22 54/100 4/5102 1/2 26/130 18/36 12/32 15/68 12/4 4/4	
<pre>godincolum(isse = sec_jtot") docenting mndicedir.fanzo.maccewich.indedir docenting mndicedir.fanzo.maccewich.indedir docenting mndicedir.fanzo.maccewich.indedir docenting mndicedir.fanzo.maccewich.edu docenting mndicedir.fanzo.maccewich.edu docenting mndicedir.fanzo.maccewich.edu docenting mndicedir.fanzo.maccewich.edu docenting mndicedir.fanzo.maccewich.edu docenting mndicedir.fanzo.maccewich.edu docenting mndicedir.fanzo.maccewich.edu docenting mndicedir.fanzo.maccewich.edu docenting mndicedir.fanzo.maccewich.edu docenting mndicedir.fanzo.maccewich.edu docenting mndicedir.fanzo.maccewich.edu docenting mndicedir.fanzo.maccewich.edu docenting mndicedir.fanzo.maccewich.edu docenting mndicedir.fanzo.maccewich.edu docenting mndicedir.fanzo.maccewich.edu docenting mndicedir.fanzo.maccewich.edu docenting mndicedir.fanzo.maccewich.edu docenting</pre>	nseaad nseadout cotococit Anaders fuel dud cotococit Anaders fuel cotococit Anaders cotococit Anaders	1 4 3 2 2 1 1 1	61% (4)/205 67% (4)/241 67% (2)/2/245 66% (4)/6 61% (2)/2/25 59% (4)/6 59% (4)/3 59% (4)/3 59% (4)/3 50% (2)/2/3 50% (2)/3 57% (4)/3 57% (4)/3	46% 54% 41% 20% 30% 30% 37% 26% 25% 45% 45%	15/22 34/150 4-2102 1/2 26/126 18/36 12/22 18/66 12/4 45/500	
Attended in fanab. nanosestich.indeligitat     41     private SecurityProfile securityProfile;     incommunity       Indical	steartic-functioned.base consolid.handles fl-adi descritering fl-adiantic consolid.handle.mag descritering fl-adiantical fl-adia	1 4 3 2 2 1 1 1	67% 143/24 67% 122/239 65% 4/6 65% 4/6 55% 236/33 35% 86/7 35% 86/7 35% 86/7 35% 82/7 35% 122/35 55% 221/360 55% 7/3 55% 7/3 35% 111/202	54% 41% 20% 30% 37% 26% 25% 45%	54/100 4.5/102 1/2 26/136 10/38 42/22 18/66 12/4 45/500	
Subscience     42     sconstability       Subscience     11     public     BecurityFunction() (     sconstability       Stonal.infano.nanosithCh.min     44     sconstability     sconstability       Stonal.infano.nanosithCh.min     46     sconstability     sconstability       Stonal.infano.nanosithCh.min     46     sconstability     sconstability       Stonal.infano.nanosithCh.min     46     sconstability     sconstability       Stonal.infano.nanosithCh.setHernet     46     sconstability     sconstability       Stonal.infano.nanosithCh.setHernet     40     sconstability     sconstability       Stonal.infano.nanosithCh.setHernet     40     sconstability     sconstability       Stonal.infano.nanosithCh.setHernet     51     sconstability     sconstability       Stonal.infano.nanosithCh.setHernet     51     sconstability     sconstability       Stonal.infano.nanosithCh.setHernet     51     sconstability     sconstability       <	costocial handlers fuel dual costocial, An numa costocial, An numa costocial, anno costocial	1 4 3 2 2 1 1 1	67% 172/255 66% 4/6 61% 326/53 55% 26/4 55% 44/53 55% 44/53 55% 51/760 55% 122/30 55% 122/30 55% 112/28	41% 50% 20% 30% 37% 26% 25% 45%	+25102 1/2 28/126 10/28 12/32 18/60 1/4 45/100	
Sideoplini, fange, manoparkich, addionation, or     43     11     public SecurityFunction() (     momenting       Sideoplini, fange, manoparkich, addioning     45     11     )     momenting       Sideoplini, fange, manoparkich, addioning     45     11     )     momenting       Sideoplini, fange, manoparkich, addioning     47     0     momenting     momenting       Sideoplini, fange, manoparkich, addioned     49     0     this, manoparkich, addioned     momenting       Sideoplini, fange, manoparkich, addioned     49     0     this, manoparkich, addioned     momenting       Sideoplini, fange, manoparkich, addioned     50     0     this, manoparkich, addioned     momenting       Sideoplini, fange, manoparkich, addioned     51     momenting     momenting     momenting       Sideoplini, fa	thead     t	1 4 3 2 2 1 1 1	66% 4/6 61% 320/51 59% 88/115 59% 88/115 58% 164/261 58% 125/560 58% 122/35 57% 8/18 53% 111/200	50% 20% 30% 26% 25% 43% 8/A	1/2 26/136 16/38 42/32 18/66 1/4 45/100	
<pre>stdendlif.inano.manosethch.edm.to.co tionali.if.anan.manosethch.edm.to.co tionali.if.anan.manosethch.edm.toriogi. tionali.if.anan.manosethch.edm.toriogi. tionali.if.anan.manosethch.edm.toriogi. tionali.if.anan.manosethch.edm.toria.it.m</pre>	costocola, An enuns costocola, An enuns costocola esta esta esta esta esta esta costocola esta esta esta esta esta esta costocola, esta esta esta esta esta esta esta costocola, esta esta esta esta esta esta esta esta	1 4 3 2 2 1 1 1	61% 228/53 59% 26/48 59% 48/23 58% 245/283 58% 223/260 58% 122/33 57% 8/3 53% 134/208	20% 50% 37% 26% 25% 43% 8/A	18/36 \$2/32 \$8/68 \$7/4 \$5/100	
<pre>stonal.if.fanao.manosatch.combools.comy stonal.if.fanao.manosatch.combools.comy stonal.if.fanao.manosatch.combools.com stonal.if.fanao.manosatch.combools.com stonal.if.fanao.manosatch.combools.com stonal.if.fanao.manosatch.com ston</pre>	estivaris, channel, endoprint conclusion fra ancossi rotocola, acentalis, encoding colina, acempanetata insteation, imal rotocola, altesta 87, encoding rotocola, manifi 2, encoding rotocola, manifi 2, encoding	1 4 3 2 2 1 1 1	59% 28/4 59% 68/115 58% 144/285 58% 121/300 58% 122/235 57% 8/13 53% 111/208	50% 37% 26% 25% 45% N/A	18/36 \$2/32 \$8/68 \$7/4 \$5/100	
discalinfanzo.manosmith.mefactoring discalinfanzo.manosmith.mefactoring discalinfanzo.manosmith.effactoring discalinfanzo.manosmith.effactoring discalinfanzo.manosmith.effactoring discalinfanzo.manosmith.euthorization 50 0 m. 10 0 10 0	solisition fa arosess outina comenenta soutina comenenta sostenion ima rotocola shetab87.encodina rotocola shetab87.encodina rotocola shetab87.encodina	1 4 3 2 2 1 1 1	59% 88/31 58% 364/283 58% 211/960 58% 122/230 57% 8/38 53% 111/200	37% 26% 25% 45% N/A	12/32 18/68 1/4 43/100	
ddnoklifangenmendententer     47     0     public Security/Juscien(String mame, HashNet/Geourity/Farameter)     pocenting       ddnoklifangenmendententer     0     11.1.1name = name     nonenting       ddnoklifangenmendententer     49     0     this_name     nonenting       ddnoklifangenmendententer     49     0     this_parameters = (parameters = mamif     nonenting       ddnoklifangenmendententer     0     1     this_parameters = (parameters = mamif     nonenting       dage_infange_mendententer     50     0     1     nonenting       togething     0     0     1     nonenting       togething     0     0     1     nonenting       togething     1     1     nonenting     nonenting       togething     0     1     nonenting     nonenting       togething     0     1     nonenting     nonenting       togething     1     nonenting     1     nonenting       togething     1     nonenting     1     nonenting       togething     1     1     nonenting     1       togething     1     1     nonenting     1       togething     1     1     1     1       togething     1     1     1	rde anoossa ordenas anticologia outina domadnenta xxeedioni.mel rotocola "heabit", encodina rotocola, negoli 7, encodina rotocola nego	4 3 2 2 1 1	58% 164/281 58% 211/360 58% 122/230 57% 8/18 53% .11/200	26% 25% 45% N/A	18/60 1/4 43/100	
Storket.infange.naroswitch.scurinty     45 0     this.mee = name;     income infange;     income infange; <td>rotocols.posii 7.encoding outina.components voetoon.mo rotocols.shetab87.encoding rotocols.neuin87.encoding rotocols.ase</td> <td>3 2 2 1 1</td> <td>58% 211/360 58% 122/215 57% 8/1 53% 11/208</td> <td>25% 45% N/A</td> <td>45/100</td> <td></td>	rotocols.posii 7.encoding outina.components voetoon.mo rotocols.shetab87.encoding rotocols.neuin87.encoding rotocols.ase	3 2 2 1 1	58% 211/360 58% 122/215 57% 8/1 53% 11/208	25% 45% N/A	45/100	
denduir(hosp.nancdeddh.addienned denduir(hosp.nancdeddh.addienned base.ir(hosp.nanceddh.addienned base.ir(hosp.nanceddh.addienned base.ir(hosp.nanceddh.addienned ) public SecurityFunction(String name) ( biogenth. ) ) ) ) ) ) ) ) ) ) ) ) )	outina.components exception.impl votocols.shetab87.encoding votocols.hesp votocols.hesp	2 2 1 1 1 1	50% 122/235 57% 8/14 53% 111/208	45% N/A	45/100	
<pre>dismain_inspan_encosmitch.terminal.stm.s 50 0 1</pre>	votecois.shetab87.encoding votocois.shetab87.encoding votocois.negin87.encoding votocois.aege	1	57% 8/13 53% 111/208	N/A		
bace_infansp.nerosentch.subbrication bace_infansp.nerosentch.subbrication	votocols.shetab87.encoding votocols.negin87.encoding votocols.base	1	53% 111/200			
bace.ir.fang.manoswitch.subhorgation.co* 52 public SecurityTunction(String mame) ( bace.ir.fang.manoswitch.subhorgation.co* 53 bis(mame, mill) ( docentity ( docen	rotocols.negin87.encoding rotocols.base	1				
m ) 53 0 this (name, null); second, a second by second b	rotocols.base	1		0%	0/4	
Anoswith a			53% 111/208	0%	6/4	
		10	51% 23/64		2/8	
	rotocols.pos87	10	50% 330/695		105/326	_
anap.nanoswitch.security.base	cheduler.base	2	50% 36/22		14/60	
	ersistence	10	45% 54/110		16/26	
	til.encoders	5	45% 46/102		16/48	
	learing.cyclecriteria	3	44% 16/36	0%	0/32	
and the second se		6	41% 22/209		3/10	
	rotocols.ndc.encoding	2	41% 220/526		0/177	
		10	40% 82/205	4%		
63	nessage.components	2	40% 100/270		42/156	
Manager (0%)	learing.consts	5	38% 43/113		11/39	
Store (0%)		16	38% 244/624		46/246	
Money (M(A)	learing.settlement	7	37% 229/604		53,312	
	ersistence.impl	6	36% 62/168		17/44	
and a second s	rotocols fx imp	20	35% 647/1813		108/834	
antirection board (078) 60 million model antilogram (Sering board )		2	35% 00/167		11/24	
untyParameter (21%)	ecurity.securekey	4	34% 38/129	37%	2/0	
undyProfile (12%)	uthorization.component	2	34% 80/232		25/156	
landsmitch.		12	34% 344/996		106/329	_
MODE ALL CONTRACTOR OF	vrotocols.shetab87	8	33% 164/491		52/222	
73 public SetSecurityParameters getParameters() {	alectar	10	31% 47/463	12%	29/219	

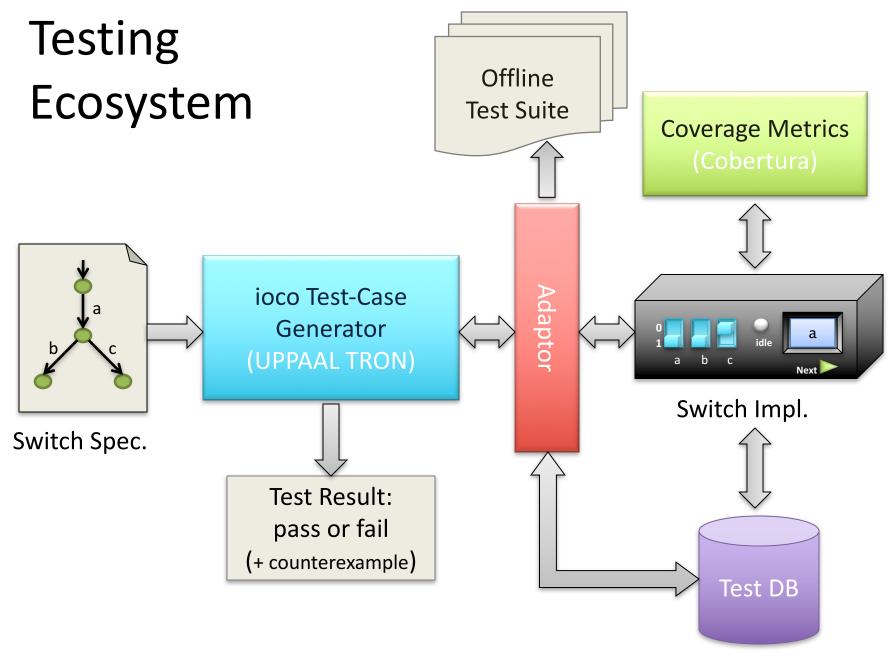
#### What about model-based coverage?

> D. F.

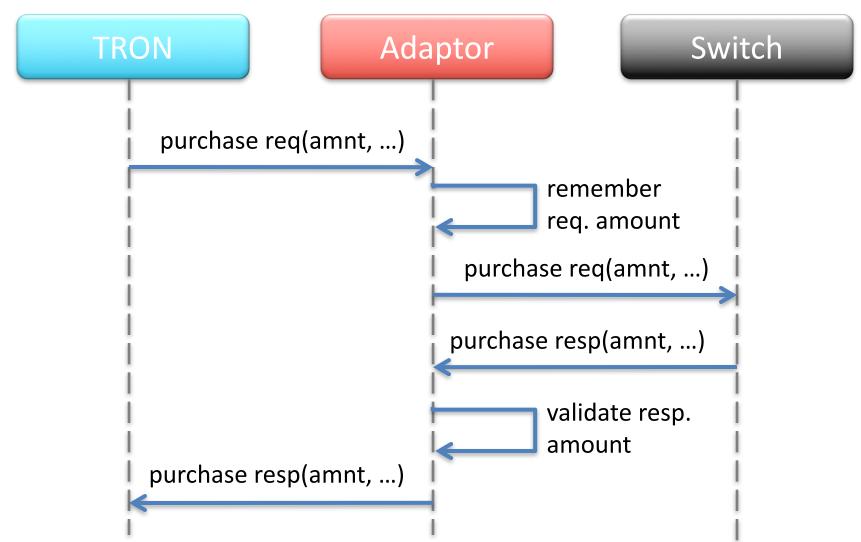
# Logging Offline Test Suite

- The adaptor logs the messages from TRON
- Prioritization based on coverage
- Used for regression testing





# **Testing Business Rules**



# Results

- Have found a number of bugs in the operational system
  - One traced back to null-pointer dereferencing
  - Poor exception handling

• Code coverage of about 40%



## Observations

- Modeling language and tool limitations
  - Component (de-compositional) testing
  - Asynchrony
  - Data specification and selection
  - Variability
- Scalability Issues
  - Running concurrent MBT instances
  - Reducing buffer lengths, pool sizes, etc.



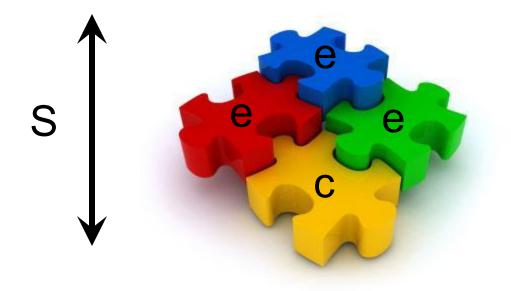
# **Further Reading**

- Asaadi, Khosravi, MRM, and Noroozi. Towards Model-Based Testing of Electronic Funds Transfer Systems. Proc. of FSEN 2011. Models publicly available on Assembla.
- Vishal, Kovacioglu, Kherazi, and MRM.
   Integrating Model-Based and Constraint-Based Testing Using SpecExplorer. Proc. of MoTiP 2012.

# Back to Theory:

**Component Testing** 

### **Decompositional Testing**

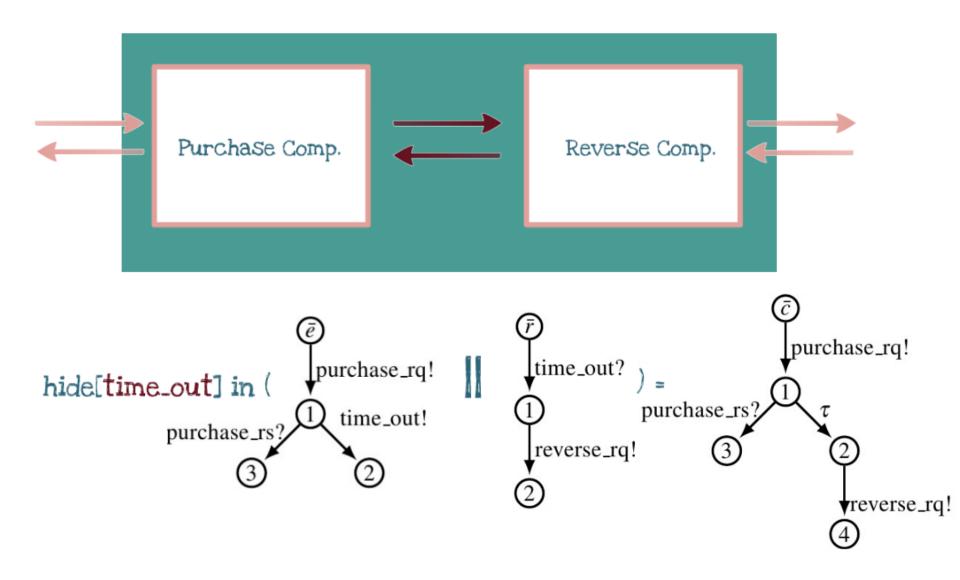


### **Decompositional Testing**



### for all c, (c || e) ioco s iff c ioco S/e

### Compositional Testing with joco [M. van der Bijl, A. Rensink & J. Tretmans -2003]



### **Decompositional properties**

1. Decomposability

$$\exists \vec{s}', \forall \bar{c}, \bar{e}. \ \bar{c} \text{ ioco } \vec{s}' \quad \Rightarrow \quad \bar{c} \| \bar{e} \text{ ioco } \bar{s}$$

2. Strong decomposability  $\exists \vec{s}', \forall \bar{c}, \bar{e}. \ \bar{c} \ \mathbf{ioco} \ \vec{s}' \qquad \Longleftrightarrow \qquad \bar{c} \| \bar{e} \ \mathbf{ioco} \ \bar{s}$ 

### **Decompositional Testing**

for all c, (c || e) ioco s iff c ioco S/e

Construction of S/e:

• Check:

Can S be the composition of e with some c?

• Filter out the behavior of e in S

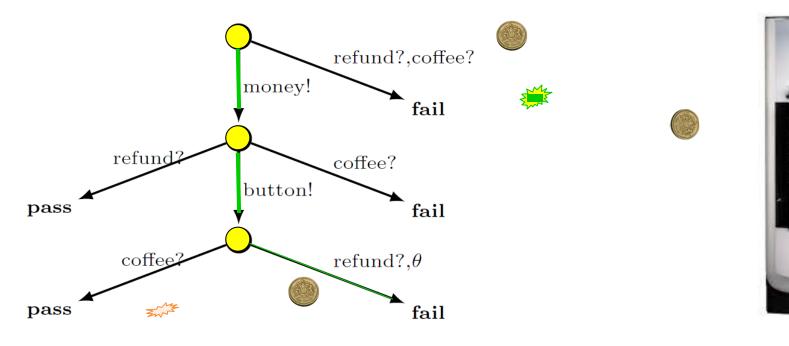
# **Further Reading**

Noroozi, MRM, and Willemse. Decomposability in Input Output Conformance Testing. Proc. of MBT 2013.

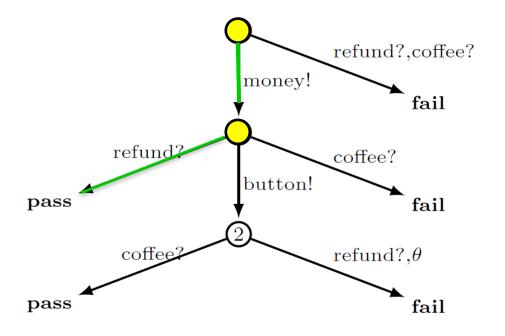
#### Back to Theory:

Asynchrony

#### **Asynchronous Communication**

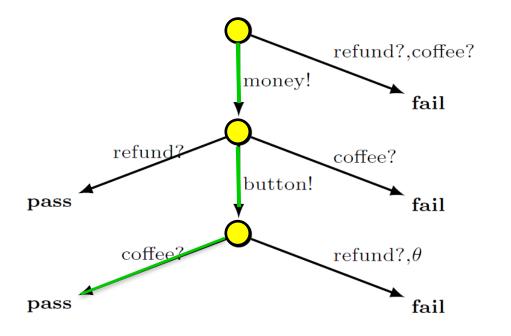


## Synchronous Communication





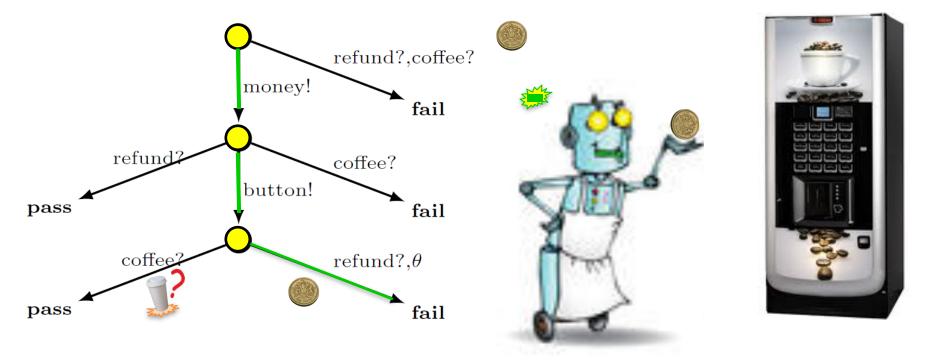
## Synchronous Communication



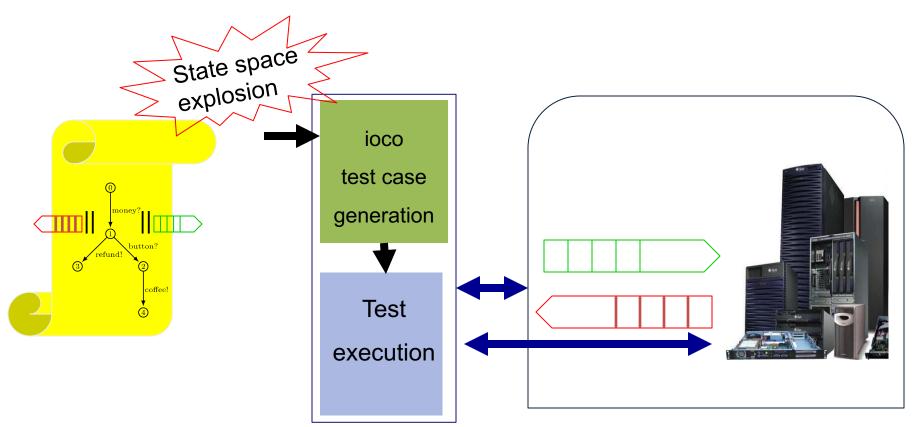


Pasargad School 2018

#### **Asynchronous Communication**

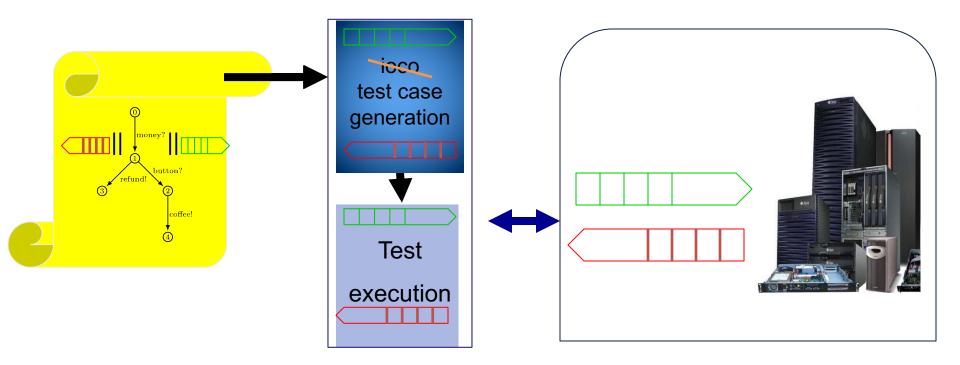


#### Prior Art [Tretmans&Verhaard'92]

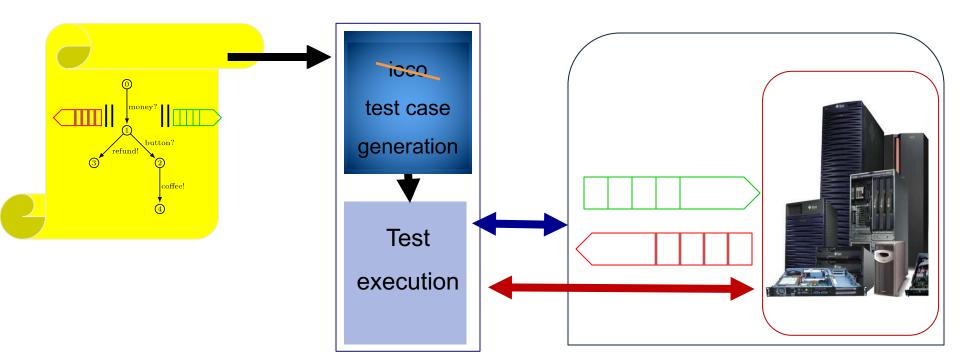


#### State of the Art

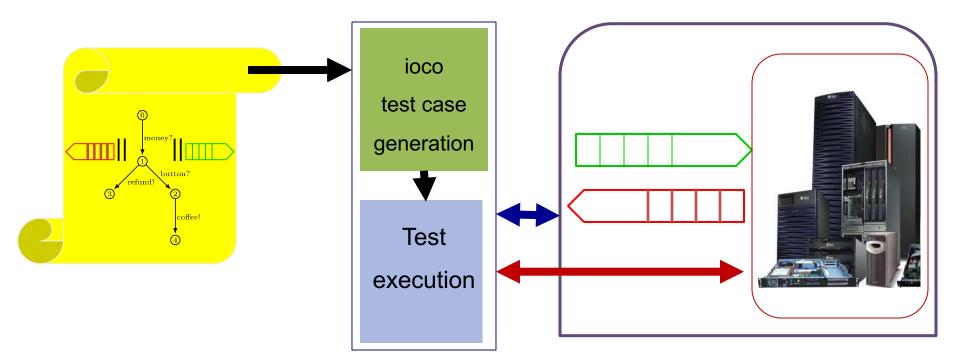
[Petrenko&Yevtushenko'02,03][Simao&Petrenko'10]



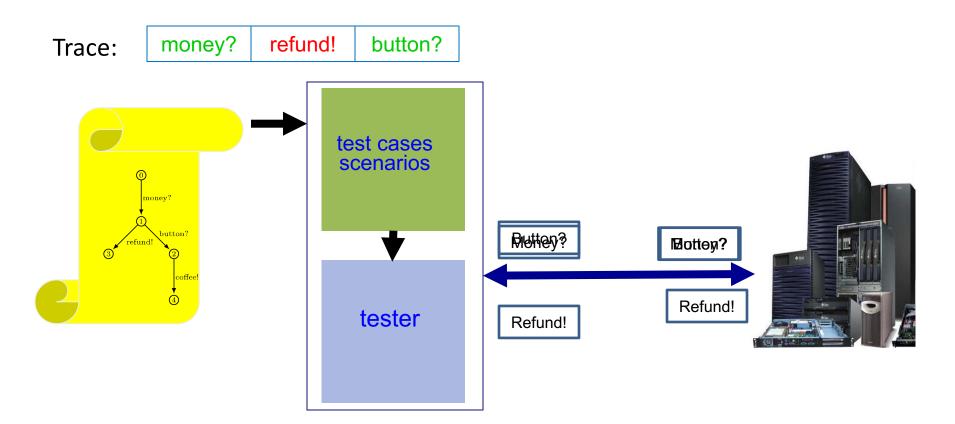
#### State of the Art [Weiglhofer&Wotawa'09]



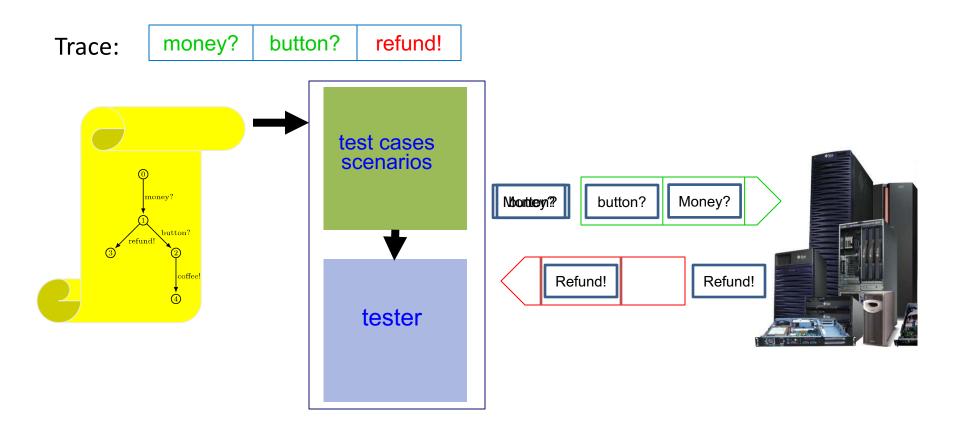
#### What we are after



#### **Delayed Traces**

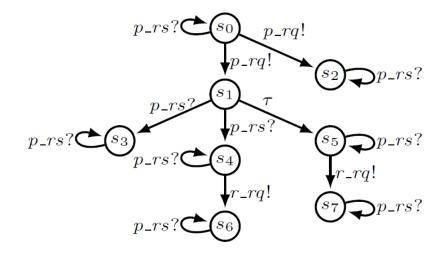


#### **Delayed Traces**



## **Delayed right-closed IOTS**

**Delay right-closed IOTS**, *S* is an IOTS such that  $\forall \sigma \in$  Straces (S) then delayed trace of  $\sigma \in$  Straces (S)



 $\sigma$ .x.a  $\in$  Straces (S) then  $\sigma$ .a.x  $\in$  Straces (S)

#### Theorems

Theorem

If implementation **i** is delay right-closed, then

i ioco Spec if and only if Q(i) ioco Spec

#### Theorem

If  $\forall t \in \text{TestCases}(S)$ , i passes t if and only if Q(i) passes t then implementation i is delay right-closed IOTS

## **Further Reading**

Noroozi, Khosravi, MRM, and Willemse. Synchrony and Asynchrony in Conformance Testing. SoSym J., 2015.



Pasargad School 2018

# Thank You Very Much!

#### mm789@le.ac.uk