Formal testing of timed and probabilistic systems

Manuel Núñez

Universidad Complutense de Madrid

23rd Int. Conf. on Testing of Software and Systems
Outline

1. Motivation
2. The Formalism
3. Conformance Relations
4. Application and Derivation of Test Cases
5. Other Time Domain: Stochastic time
6. Timed Systems with Probabilistic Information
Formal testing is currently a big area

- In fact, there is a ton of workshops, conferences, symposia, special issues on formal testing!
- Focus of this talk: Relations between processes and application of tests to systems.
Formal testing is currently a big area

- In fact, there is a ton of workshops, conferences, symposia, special issues on formal testing!
- Focus of this talk: Relations between processes and application of tests to systems.

Testing can be informal, too...
Formal testing is currently a big area

- In fact, there is a ton of workshops, conferences, symposia, special issues on formal testing!
- Focus of this talk: Relations between processes and application of tests to systems.

Testing can be informal, too... But this is out of the scope of this talk
Introduction

This talk in a nutshell

- Having a model/spec and a (black-box) SUT, try to establish the conformance of the SUT with respect to the spec by applying experiments.
- Add probabilities and/or time to the considered systems.
Temporal Requirements

- The temporal behavior of real-time systems is critical.
- Different time domains for representing conditions over the time consumed by the systems while it performs actions.
Temporal Requirements

- The temporal behavior of real-time systems is critical.
- Different time domains for representing conditions over the time consumed by the systems while it performs actions.

Fix Time

- A time value specifies the amount of time needed to perform an action.
- Not always is easy to precisely establish time requirements.
Temporal Requirements

- The temporal behavior of real-time systems is critical.
- Different time domains for representing conditions over the time consumed by the systems while it performs actions.

Time intervals

- A time interval specifies the range of time values that the system can take to perform an action.
Temporal Requirements

- The temporal behavior of real-time systems is critical.
- Different time domains for representing conditions over the time consumed by the systems while it performs actions.

**Stochastic time**

- Specify that with probability $p \ an \ action \ must \ be \ performed \ before \ t \ time \ units \ have \ passed$.
- Sometimes the specifier either does not have such probabilistic information or considers it unnecessarily complicate the model.
Formal testing methodology

The formalism
Representation of systems that present non-standard requirements.

Notion of conformance
What it means for an implementation to conform to a specification.

Derivation and application of tests
- Algorithms for derivation of tests from the specification.
- Relation between application of tests and conformance.
Common conditions

- Specification and Implementations can be expressed using the same formalism.
- Implementations are input-enabled.
- Both specifications and implementations are observable (we sometimes remove this restriction).

Deterministic  Observable  Non-Deterministic

```
  1
 / \         / \         / \
 a1/b1  a2/b1  a1/b1  a1/b2  a1/b1
 2   3     2   3      2   3
```

Finite State Machines

\[ M = (S, \mathcal{I}, \mathcal{O}, Tr, s_{in}) \]

- \( S \) is a finite set of states.
- \( \mathcal{I} \) is the set of input actions.
- \( \mathcal{O} \) is the set of output actions.
- \( Tr \) is the set of transitions.
- \( s_{in} \) is the initial state.
Finite State Machines
FSMs extended with Time Values
Conformance Relations

Functional Implementation Relation

An implementation $l$ \textit{conforms} to a specification $S$ if for all possible evolution of $S$ the outputs that the implementation $l$ may perform after a given input are a subset of those of the specification.

$l \text{ conf}_{nt} S$, if for all $e = (i_1/o_1, \ldots, i_{r-1}/o_{r-1}, i_r/o_r) \in \text{NTEvol}(S)$, with $r \geq 1$, we have that

$e' = (i_1/o_1, \ldots, i_{r-1}/o_{r-1}, i_r/o'_r) \in \text{NTEvol}(l) \implies e' \in \text{NTEvol}(S)$
Examples of functional conformance

\[
\begin{align*}
S & \xrightarrow{a_1/b_4} 1 \xrightarrow{a_1/b_4} 2 \xrightarrow{a_2/b_3} 1 \\
1 & \xrightarrow{a_1/b_4} 2 \xrightarrow{a_2/b_4} 1
\end{align*}
\]

\[
\begin{align*}
I & \xrightarrow{a_1/b_4} 1 \xrightarrow{a_1/b_4} 2 \xrightarrow{a_2/b_3} 2 \\
1 & \xrightarrow{a_1/b_4} 2 \xrightarrow{a_2/b_4} 1
\end{align*}
\]

\[
\begin{align*}
S & \xrightarrow{a_3/b_3} 1 \xrightarrow{a_1/b_1} 2 \xrightarrow{a_2/b_2} 3 \\
1 & \xrightarrow{a_1/b_1} 2 \xrightarrow{a_2/b_2} 3 \xrightarrow{a_3/b_3} 1 \\
3 & \xrightarrow{a_3/b_3} 2 \xrightarrow{a_2/b_2} 3
\end{align*}
\]

\[
\begin{align*}
I & \xrightarrow{a_3/b_3} 1 \xrightarrow{a_1/b_1} 2 \xrightarrow{a_3/b_1} 3 \\
1 & \xrightarrow{a_1/b_1} 2 \xrightarrow{a_2/b_1} 3 \xrightarrow{a_3/b_3} 1 \\
3 & \xrightarrow{a_3/b_3} 2 \xrightarrow{a_2/b_1} 3
\end{align*}
\]

\[
\begin{align*}
I & \xrightarrow{a_3/b_3} 1 \xrightarrow{a_1/b_1} 2 \xrightarrow{a_3/b_1} 3 \\
1 & \xrightarrow{a_1/b_1} 2 \xrightarrow{a_2/b_1} 3 \xrightarrow{a_3/b_3} 1 \\
3 & \xrightarrow{a_3/b_3} 2 \xrightarrow{a_2/b_1} 3
\end{align*}
\]

Not \( I \) conf\_nt \( S \)
Timed conformance relations: Right choice?

We require $I \text{conf}_{nt} S$ plus something else...
Timed conformance relations: Right choice?

We require $l \text{ conf}_{nt} S$ plus something else...

- $l \text{ conf}_{a} S$: All evolutions in $l$ take time $\equiv$ to an equivalent evolution in $S$. 
Timed conformance relations: Right choice?

We require $I \text{conf}_{nt} S$ plus something else...

- $I \text{conf}_a S$: All evolutions in $I$ take time $= \text{to an equivalent evol}uation in S$.
- $I \text{conf}_w S$: All evolutions in $I$ take time $\leq \text{the slowest equivalent evolution in } S$.
- $I \text{conf}_b S$: All evolutions in $I$ take time $\leq \text{the fastest equivalent evolution in } S$.
- $I \text{conf}_{sw} S$: Some evolution in $I$ takes time $\leq \text{to the slowest equivalent evolution in } S$.
- $I \text{conf}_{sb} S$: Some evolution in $I$ takes time $\leq \text{to the fastest equivalent evolution in } S$. 
Relations among relations

\[ l \text{conf}_b S \Rightarrow l \text{conf}_{sb} S \]
\[ \Downarrow \]
\[ l \text{conf}_a S \Rightarrow l \text{conf}_w S \Rightarrow l \text{conf}_{sw} S \]
Relations among relations

\[ l \text{ conf}_b S \Rightarrow l \text{ conf}_{sb} S \]
\[ \Downarrow \Downarrow \]
\[ l \text{ conf}_a S \Rightarrow l \text{ conf}_w S \Rightarrow l \text{ conf}_{sw} S \]

**time-deterministic specifications**

\[ \text{conf}_w = \text{conf}_b \text{ and } \text{conf}_{sw} = \text{conf}_{sb} \]

**time-deterministic implementations**

\[ \text{conf}_w = \text{conf}_{sw} \text{ and } \text{conf}_b = \text{conf}_{sb}. \]

**time-deterministic specifications and implementations**

\[ \text{conf}_a \text{ is still different from conf}_b. \]
## Timed Test Cases

### What is a test case?
Tests represent sequences of inputs applied to an implementation.

### Checking functional behavior
Once an output is received, the tester checks whether it belongs to the set of expected ones or not.

### Checking temporal behavior
Tests include *time stamps* to check the correct performance of actions.
Applying tests

\[ I \parallel T_1 \checkmark \text{ but } I \parallel T_2 \checkmark \]
Application of Tests to Implementations

Checking functional behavior

Terminal states reached by the composition of implementation and test belong to the set of *passing* states.

After we know that the functional behavior of the implementation is correct with respect to the suite, we check time conditions.

Checking time conditions

Different notions of passing tests corresponding to:

- Domain for representing time requirements.
- Underlying conformance relation.
Test Derivation

Algorithm for generating a *test suite* from a spec $S$. 
Test Derivation

Algorithm for generating a test suite from a spec \( S \).

- Generating one test for each possible trace in \( S \).
- Non-deterministic choice with two possibilities.
  - Close a branch with a pass/fail state.
  - Continue testing in the branch.
- A time stamp is attached to each pass state: Time cumulated so far to perform the sequence (values are taken from \( S \)).
Test Derivation

Algorithm for generating a test suite from a spec $S$.

- Generating one test for each possible trace in $S$.
- Non-deterministic choice with two possibilities.
  - Close a branch with a pass/fail state.
  - Continue testing in the branch.
- A time stamp is attached to each pass state: Time cumulated so far to perform the sequence (values are taken from $S$).

By considering all choices we get a sound and complete test suite.
Uncomplete Completeness

- In general, generated test suites are infinite.
- Completeness in the limit
  - Generating all test of length $n$.
  - Completeness is achieved when $n$ tends to infinite.
Limitations of fixed time values

- Sometimes we cannot simply use a value to represent a time constraint.
- Consider more expressive frameworks: Stochastic time
What is Stochastic Time?

*Combination* of deterministic time and probabilities.
What is Stochastic Time?

*Combination* of deterministic time and probabilities.

What is Deterministic Time?
The message will arrive before 2 seconds.
**What is Stochastic Time?**

*Combination* of deterministic time and probabilities.

**What is Deterministic Time?**

The message will arrive before 2 seconds.

**How do we use Probabilities?**

The message will arrive with probability 0.95.
What is Stochastic Time?

*Combination* of deterministic time and probabilities.

What is Deterministic Time?
The message will arrive before 2 seconds.

How do we use Probabilities?
The message will arrive with probability 0.95.

So, stochastic time is...
What is Stochastic Time?

*Combination* of deterministic time and probabilities.

What is Deterministic Time?
The message will arrive before 2 seconds.

How do we use Probabilities?
The message will arrive with probability 0.95.

So, stochastic time is...
The message will arrive before 2 seconds with probability 0.95.
FSMs extended with Stochastic Time

\[ F_{\xi_1}(x) = \begin{cases} 
0 & x < 0 \\
\frac{x}{2} & 0 \leq x < 2 \\
1 & 2 \leq x 
\end{cases} \]

Uniform distribution in \([0, 2)\)
Random Variables

- A *random variable* $\xi$ takes values according to a *probability distribution function* $F_\xi$.
- $F_\xi(x) = p$ means the probability that $\xi$ takes a value smaller than or equal to $x$ is $p$.

- Random variables can be *added*. We may define $\xi$ distributed as $\xi_1 + \xi_2$.

- In our framework, random variables always denote time: $F_\xi(x) = 0$, for $x < 0$.

- We write $\xi_1 \sim \xi_2$ if for all $x$ we have $F_{\xi_1}(x) = F_{\xi_2}(x)$. 
Simulating Deterministic Time

**Dirac distributions:** The message arrives exactly at time 2

\[
F(x) = \begin{cases} 
0 & x < 2 \\
1 & 2 \leq x 
\end{cases}
\]

**Uniform distributions:** The message arrives in the interval [0, 2], being times *equiprobable*

\[
F(x) = \begin{cases} 
0 & x < 0 \\
\frac{x}{2} & 0 \leq x < 2 \\
1 & 2 \leq x 
\end{cases}
\]
A First Implementation Relation

Stochastic traces of $M$: $\text{STraces}(M)$
A trace plus the random variable denoting time.

$I \text{ conf}_S S$

$I$ stochastically conforms to $S$ if $I \text{ conf}_{nt} S$ and the random variables associated with a given trace in $S$ and in $I$ are equivalent.

\[
(\rho, \xi) \in \text{STraces}(I) \\
\wedge \\
\rho \in \text{NSTraces}(S) \\
\implies (\rho, \xi') \in \text{STraces}(S) \wedge \xi \sim \xi'
\]
Properties of $conf_s$

- From a *theoretical* point of view, $conf_s$ is an appropriate conformance relation.
Properties of $\text{conf}_s$

- From a *theoretical* point of view, $\text{conf}_s$ is an appropriate conformance relation.

**Unfortunately**

**Specification**

We know everything about its probability distribution functions.

**Implementation**

Under *black-box* testing, we don’t have access to them.
Properties of $\text{conf}_s$

Time Intervals and Stochastic Time

- $((i_1/o_1, \ldots, i_n/o_n), t)$ is a timed execution of $I$ if the observation of $I$ shows $(i_1/o_1, \ldots, i_n/o_n)$ performed in time $t$.
- $H = \{(e'_1, t_1), \ldots, (e'_n, t_n)\}$ be a multiset of timed executions:

$$\text{Sampling}_H(e) = \{t \mid (e, t) \in H\}$$
Hypothesis Contrast

- We take from the specification the *expected* performance $\xi_S$.
- We apply the test several times and obtain a set of *time values* $H$.
- We determine whether these are *similar* enough up to a confidence level.
Hypothesis Contrast

- We take from the specification the **expected** performance $F_S$.
- We apply the test several times and obtain **time values**:

<table>
<thead>
<tr>
<th>At time</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.75</td>
<td>21 observations</td>
</tr>
<tr>
<td>1</td>
<td>16 observations</td>
</tr>
<tr>
<td>1.5</td>
<td>25 observations</td>
</tr>
<tr>
<td>2.15</td>
<td>12 observations</td>
</tr>
<tr>
<td>2.3</td>
<td>48 observations</td>
</tr>
<tr>
<td>2.8</td>
<td>4 observations</td>
</tr>
</tbody>
</table>

- From these time values we obtain a function $F_I$. 
Hypothesis Contrast

- We determine whether these two functions are *similar* enough (up to a level of confidence).

\[ F_S \]
\[ F_I \]
Stochastic Time

Formally, $I(\alpha, H)$ stochastic $S$

$$(e, \xi_S) \in \text{TEvol}(S) \implies \gamma(\xi_S, \text{Sampling}_{(H,\phi)}(e)) > \alpha$$
Stochastic Time

- Only difference with previous tests: Random variables are assigned to passing states.
- Test executions are compared with the random variable attached to the pass states.
- Test executions are used by different tests sharing the same observed sequence.
Stochastic Time

Passing a Test

I passes the test $T$ with probability $\alpha$ if after applying the test several times:

- The test always finishes in a passing state.
- Time values observed might have been generated by the random variables of the test.
One step forward...

We might like to quantify non-deterministic choices of systems.
Adding probabilities to FSMs

- Goal: quantify non-determinism.

\[
\begin{align*}
&x/y & x/y' \\
&x/y, p & x/y', 1-p
\end{align*}
\]
Adding probabilities to FSMs

- Goal: quantify non-determinism.
- No probabilistic relation among different inputs.
Adding probabilities to FSMs

- Goal: quantify non-determinism.
- No probabilistic relation among different inputs.
- For each input $x$ and state $s$, the probabilities associated with $s$ and $x$ add up to 1
Probabilistic Stochastic Finite State Machine

A non-deterministic finite state machine in which every transition also has an associated probability and a random variable.

\[ M = (S, s_0, Li, Lo, P_T, P_V) \]

- \( S \) is a finite set of states
- \( s_0 \) is the initial state
- \( Li \) is the set of input actions
- \( Lo \) is the set of output actions
- \( P_T : S \times Li \times Lo \times S \rightarrow [0, 1] \) is the probability-transition function
- \( P_V : S \times Li \times Lo \times S \rightarrow \mathcal{V} \) is the time function.
- For all \( s \in S \) and \( a \in Li \), \( \sum_{p \in P_T(s,a,x,s')} p = 1 \).
Probabilistic Stochastic Finite State Machine

For all \((s, a, x, s') \in S \times Li \times Lo \times S\) if \(P_T(s, a, x, s') = p > 0\) and \(P_Y(s, a, x, s') = \xi\) then \((s, a, x, p, \xi, s')\) is a transition of \(M\).

Transition: \((s, a, x, p, \xi, s')\)

If the machine is in state \(s\) and receives the input \(a\) then

- with probability \(p\)
- it produces the output \(x\), and
- it moves into the state \(s'\).
- before time \(t\) with probability \(F_\xi(t)\)
Execution Time of Input Sequences

- Let $M = (S, s_0, Li, Lo, P_T, P_Y)$ be a PSFSM,
- $\bar{a}/\bar{y}$ be an input/output sequence, and
- $s \in S$ a state.

Time spent to reach the state $s'$ from $s$ performing with $\bar{a}/\bar{y}$

$$P^*_Y(s, \varepsilon, x, s') = \theta$$

$$P^*_Y(s, \bar{a}, \bar{x}, s'') + P_Y(s'', a, x, s') \quad \text{if } \exists s'' \in SP_T(s'', a, x, s') > 0$$

$$P^*_Y(s, \bar{a}, \bar{x}, s'') \quad \text{if otherwise}$$
Example

\[ p_T^*(s_1, x_1x_2, y_1y_1) = \frac{1}{4} \times \frac{3}{4} \]

\[ p_V^*(s_1, x_1x_2, y_1y_1) = \xi_1 + \xi_3 \]
Conformance Relations

Correctness

Require that traces of the specification that can be performed by the implementation have the same associated probability and delay
Conformance Relations

Correctness

Require that traces of the specification that can be performed by the implementation have the *same associated probability and delay*.

The same problem

No access to probabilities and random variables of the implementation.
Conformance Relations

**Correctness**

Require that traces of the specification that can be performed by the implementation have the **same associated probability and delay**.

**The same problem**

No access to probabilities and random variables of the implementation.

**Proposal based on a finite set of observations**

Check that the observed outputs and execution times in the implementation **fit** the probabilities and random variables of the specification.
Notion of *fitting*

**Probabilistic conformance: Interval estimation**

Do not request that the probabilities of the implementation be equal to the ones corresponding to the specification but that this fact happens *up to a certain probability*.

**Temporal conformance: Hypothesis contrast**

Decide whether the sample could be generated by the corresponding random variable *with a certain confidence*.
Let $S$ and $I$ be PSFSM,
$H$ be a multiset of timed executions of $I$
$\Phi = \{\bar{\sigma} \mid \exists t : (\bar{\sigma}, t) \in H\}$, and
$0 \leq \alpha \leq 1$.

\[ (\alpha, H)\text{-probabilistically conforms to } S \]
For all $\bar{\sigma} = \bar{a}/\bar{x}$ such that $P^*_T(s_0, \bar{a}, \bar{x}) > 0$ we have
\[ P^*_T(s_0, \bar{a}, \bar{x}) \in \text{Cl}_\alpha(\text{SeqSampling}_{(H,\Phi)}(\bar{\sigma})) \]

\[ (\alpha, H)\text{-stochastically conformance to } S \]
For all $\bar{\sigma} = \bar{a}/\bar{x}$ we have
\[ \gamma \left( P^*_V(s_0, \bar{a}, \bar{x}), \text{Sampling}_{(H,\Phi)}(\bar{\sigma}) \right) > \alpha \]
Testing Probabilistic and Stochastic Systems

Checking behavior

1. Tests include *verdicts* to determine whether the output observed belongs to the set of expected ones or not.
2. Tests include *probabilities*.
3. Tests include *random variables*. 
Test Case

A test case diagram with nodes labeled as follows:

- **T**
- **a**
- **b**
- **pass, \( \frac{1}{2}, \xi_1 \)**
- **fail**
- **pass, \( \frac{1}{4}, \xi_3 \)**
- **pass, \( \frac{1}{6}, \xi_5 \)**
- **fail**

Connections between nodes are indicated by arrows, with labels such as **x**, **y**, and **z**.
Applying test cases

\[ s_1 \xrightarrow{a/x, \gamma_1, 1} s_2 \]
\[ s_2 \xrightarrow{b/z, \gamma_1, 1} s_1 \]
\[ s_3 \xrightarrow{a/y, \gamma_3, 1 - p_1} s_4 \]
\[ s_4 \xrightarrow{b/z, \gamma_1, p_3} s_3 \]
\[ s_4 \xrightarrow{a/y, \gamma_3, p_1} s_2 \]
\[ s_4 \xrightarrow{b/z, \gamma_2, 1 - p_3} s_3 \]
\[ s_4 \xrightarrow{a/z, \gamma_1, 1} s_3 \]
\[ s_3 \xrightarrow{b/y, \gamma_2, 1} s_4 \]

\[ T \]
\[ a \]
\[ \rightarrow \]
\[ x \]
\[ \rightarrow \]
\[ y \]
\[ \rightarrow \]
\[ z \]
\[ \rightarrow \]
\[ fail \]
\[ pass, \xi_1, \frac{1}{4} \]
\[ pass, \xi_3, \frac{3}{8} \]
\[ fail \]
\[ pass, \xi_1, \frac{1}{8} \]
Applying test cases - Initial state
Applying test cases - We apply the input a
Applying test cases - We apply the input a
Applying test cases - The output $y$ is emitted

Timed test executions $\{(a/y/3)\}$
Applying test cases - State $s_4$ is reached

Timed test executions $\{(a/y/3)\}$
Applying test cases - We apply the input $b$

Timed test executions $\{(a/y/3)\}$
Applying test cases - The output x is emitted

Timed test executions \{(ab/yx/7)\}
Checking probabilities and delays conditions

We need several applications of the test to the implementation.

- To evaluate if a set of test executions match the distribution function associated to the random variable indicated by the corresponding state of the test.
- To evaluate if a set of test executions fit the probabilities associated to the corresponding states of the test.
Checking delays and probabilities

Sample of timed test executions

\((ab/yx/7),(ab/yz/5),(ab/yx/6.9),(ab/yx/7.1),\ldots\)

\(\frac{3}{8} \in Cl_\alpha(\{ab/yx, ab/yz, ab/yx, ab/yx, \ldots\})\)

\(\gamma(\{7, 6.9, 7.1, \ldots\}, \xi_3)\)
Some references


Other work


Thanks for you attention!