Challenges in Enforcing Behaviours on Systems
Challenges in System Realisation from Property-based Specifications.

Eric Rothstein  Daniel Schreckling

IT Security Group
University of Passau
Passau, Germany

Second BIOMICS Summer Workshop
University of St. Andrews.
18-20 June 2014
Outline

1. Motivation
2. Systems, Behaviours and Properties
3. Challenges
4. Summary and Discussion
Motivation
Realising Systems

Realising a system from a specification:

- What is your specification?
- What is the type of the system being realised?
BIOMICS: Behaviour-based specifications of interaction machines.

- What is your specification?
  Combination of (biologically inspired) interaction properties.

- What is the type of the system being realised?
  Interaction machine.
Desired System

Environment

Actions

Observer

Desired System

Desired External Behaviour
Reference Point: Theory of Enforcement

- What is your specification?
  An (enforceable) security property.
- What is the type of the system being realised?
  Enforcers and Monitors.

Enforcers and monitors are coupled to the system that needs to satisfy the security property.
Behaviour of a System

- **Environment**
- **System**
- **Observer**
- **Actions**
- **External Behaviour**
Enforcing a Behaviour in a System

- Environment
- Actions
- Enforcer
- Edited Actions
- System
- Observer
- Desired External Behaviour
Systems, Behaviours and Properties
Systems: Deterministic Sequence Recognisers

- Environment sends a sequence of $a$'s and $b$'s.
- External behaviour: colour of current state.

\[
\langle a, ab, aba, abab, ababb, ababba \rangle \mapsto \langle \bullet, \bullet, \bullet, \bullet, \bullet, \bullet, \bullet \rangle
\]
Properties and Behaviours

A behaviour is a partition of the set of all input sequences.

- Classes ● and ○ are properties.
- Complementary: never ● and ○, always ● or ○.

Example: a healthy heart should always be in a ● state. Suppose a is diastole and b is systole.
- abababa is good for now.
- abb is not good already.
Desired System with “Always ●”

Environment

Observer

abaababaab
What is really going on?

Environment

Filter

Observer

System
Other Properties

Different properties, same structure: $2^{\mid States\mid}$ binary behaviours.
Desired Behaviours and the Desired Property

- Always remain in desired states: $\bullet$.
- Avoid undesired states: $\bullet^C$.

For the heart: $\bullet = \bullet$, and $\bullet^C = \bullet$.

Environment may try to put the system in a $\bullet^C$ state, and the system should “fight” it, and remain in/return to $\bullet$. 
Behaviour Realisation – List of Requirements.

<table>
<thead>
<tr>
<th>Beh. Req.</th>
<th>Realised System</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$ $\mapsto$ $\bullet$</td>
<td></td>
</tr>
<tr>
<td>$a$ $\mapsto$ $\bullet$</td>
<td></td>
</tr>
<tr>
<td>$b$ $\mapsto$ $\bullet$</td>
<td></td>
</tr>
</tbody>
</table>

Diagram:

- $a$ from $\bullet$ to $\bullet$
- $b$ from $\bullet$ to $\bullet$
- $a$ from $\bullet$ to $\bullet$
## Behaviour Realisation – List of Requirements.

<table>
<thead>
<tr>
<th>Beh. Req.</th>
<th>Realised System</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$</td>
<td><img src="image" alt="Realised System Diagram" /></td>
</tr>
<tr>
<td>$a$</td>
<td><img src="image" alt="Realised System Diagram" /></td>
</tr>
<tr>
<td>$b$</td>
<td><img src="image" alt="Realised System Diagram" /></td>
</tr>
<tr>
<td>$aa$</td>
<td><img src="image" alt="Realised System Diagram" /></td>
</tr>
<tr>
<td>$ab$</td>
<td><img src="image" alt="Realised System Diagram" /></td>
</tr>
</tbody>
</table>
Behaviour Realisation – List of Requirements.

<table>
<thead>
<tr>
<th>Beh. Req.</th>
<th>Realised System</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon \mapsto \bullet$</td>
<td></td>
</tr>
<tr>
<td>$a \mapsto \bullet$</td>
<td></td>
</tr>
<tr>
<td>$b \mapsto \bullet$</td>
<td></td>
</tr>
<tr>
<td>$aa \mapsto \bullet$</td>
<td></td>
</tr>
<tr>
<td>$ab \mapsto \bullet$</td>
<td></td>
</tr>
<tr>
<td>$ba \mapsto \bullet$</td>
<td></td>
</tr>
<tr>
<td>$bb \mapsto \bullet$</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
## Behaviour Realisation – List of Requirements.

<table>
<thead>
<tr>
<th>Beh. Req.</th>
<th>Realised System</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$ $\mapsto q_0$</td>
<td></td>
</tr>
<tr>
<td>$a$ $\mapsto q_1$</td>
<td></td>
</tr>
<tr>
<td>$b$ $\mapsto q_2$</td>
<td></td>
</tr>
<tr>
<td>$aa$ $\mapsto q_2$</td>
<td></td>
</tr>
<tr>
<td>$ab$ $\mapsto q_0$</td>
<td></td>
</tr>
<tr>
<td>$ba$ $\mapsto q_2$</td>
<td></td>
</tr>
<tr>
<td>$bb$ $\mapsto q_2$</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Diagram:

- $q_0$ to $q_2$ with $b$ transition
- $q_0$ to $q_1$ with $a$ transition
- $q_1$ to $q_2$ with $a$ transition
- $q_2$ to $q_0$ with $b$ transition
Behaviour Realisation – List of Requirements (2)

\[ \Delta : \{a, b\}^* \rightarrow \{q_0, q_1, q_2\} \]

\[ C : \{q_0, q_1, q_2\} \rightarrow \{\bullet, \bullet\} \]

\[
\begin{align*}
q_0 & \mapsto \bullet \\
q_1 & \mapsto \bullet \\
q_2 & \mapsto \bullet \\
\end{align*}
\]

\[ B_E = C \circ \Delta \]
Behaviour Specification with Sequence Recognisers

A sequence recogniser is both a system and a behaviour description.
Meta-behaviours – (for completeness)

Meta behaviours are functions from sets of input sequences of any length to some set $X$:

$$\{a, b\}^\infty \rightarrow X$$
$$\mathcal{P}(\{a, b\}^\infty) \rightarrow X$$
$$\mathcal{P}^2(\{a, b\}^\infty) \rightarrow X$$
$$\ldots$$
$$\mathcal{P}^n(\{a, b\}^\infty) \rightarrow X$$
Realising Property-based Specifications

Challenges in Realising Property-Based Specifications
Challenge #1: Description of the Desired Property.

- \( = a.b.aa.bb.aaa.bbb.aaaa.bbbb \ldots \)

Machines do not understand Ellipsis i.e. “…”

**What is your property description language?**

**Why that one and not another?**

Choose among:

- State machines: ASM, Büchi and Streett automata, etc.
- Modal logic: LTL, CTL, \( \mu \)-calculus, etc.
- (Co)algebras and category theory.
- Domain-specific languages.
Challenge #2: What was the Desired Property?

1. Choose 1 from the $2^n$ different behaviours.
2. Choose 1 of the colours to be your desired property.

Is that really the property you wanted?

“Do as I think, not as I say!”

Hard challenge: you have to ask the experts!
Challenge #3: Avoiding Incompatible Properties.

The desired property may be written as the composition of other properties.

Properties \( \bullet, \circ, \bullet \circ \) and \( \bullet \circ \) allow:

- Intersection: “\( \bullet \) and \( \circ \)”. \( \bullet \cap \circ \)
- Union: “\( \bullet \) or \( \circ \)”. \( (\bullet^C \cap \circ^C)^C \)
- Difference: “\( \circ \), but not \( \bullet \)”. \( \circ \cap \bullet^C \)

In general, it is hard to determine whether two properties are complementary.

Worst case: every state is \( \bullet \) or every state is \( \bullet^C \)

Trivial behaviour
Challenge #4: Realising the Desired Property.

Suppose we overcame Ch. 1, 2 and 3, and we have a desired property ●.

Is it possible to realise a System that “defends” itself from the environment?

Can System force its next state to be ●?

- Some actions of the environment cannot be controlled, and force a $C^\bullet$ state.
  For example: a meteor falls.
- IRL we cannot consider all the actions of the environment: the state of the system may unexpectedly be changed!
as the Desired Property of the Heart.
Summary and Discussion
Summary

- Challenges #1, #2 and #3 split the problem of defining the desired property.
- Challenge #4 presents theoretical limitations for realisation of systems with desired behaviours.
Discussion: Reacting Back.

We want to:

- Always remain in desired states: $\bullet$.
- Avoid undesired states: $\bullet^C$.

For some reason, we are in a $\bullet^C$ state: environment.

**Can we return to a $\bullet$ state?**

We need an “immune system”.

**Equilibrium: $\bullet^C$ sink? $\bullet$ sink? Both?**
Question and Answers

Questions?

Thank you for your attention!