~~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

Motivation

Realising Systems

Realising a system from a specification:

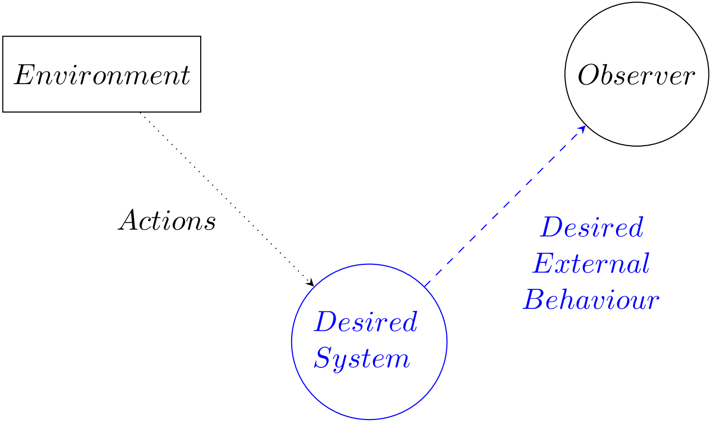
- What is your specification?
- What is the type of the system being realised?

BIOMICS

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



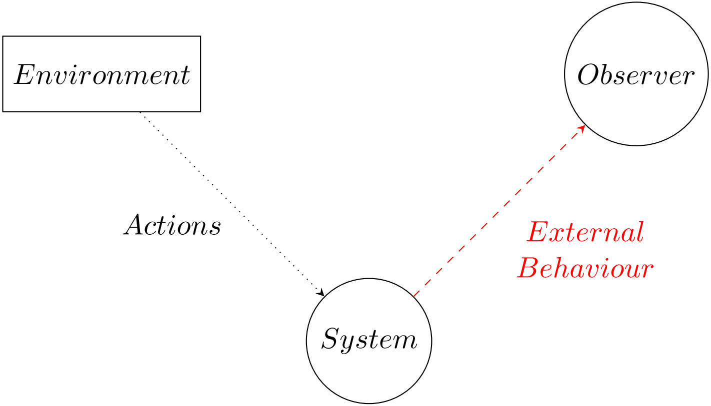
BIOMICS

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.

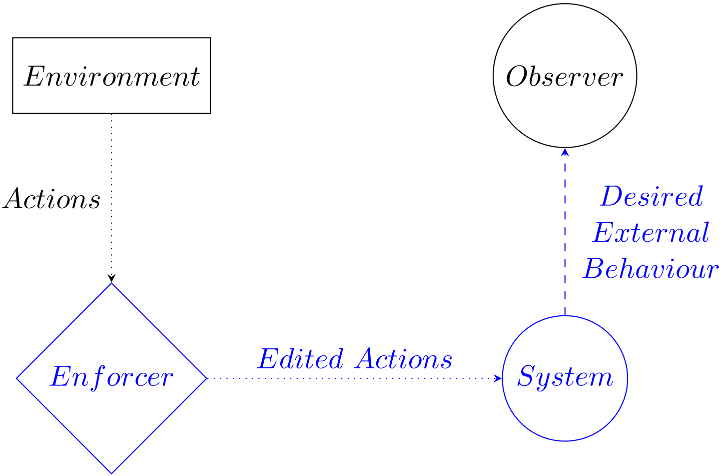
Enforcers and monitors are coupled to the system that needs to satisfy the security property.

Behaviour of a System



BIOMICS

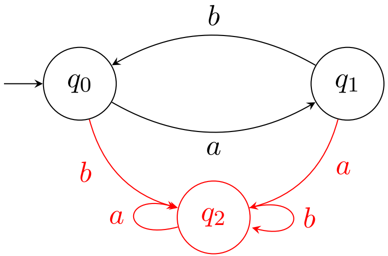
Enforcing a Behaviour in a System



Systems, Behaviours and Properties

Systems, Behaviours and Properties

Systems: Deterministic Sequence Recognisers



- Enviroment sends a sequence of a's and b's.
- External behaviour: colour of current state.
- $\langle \underline{a}, a\underline{b}, ab\underline{a}, aba\underline{b}, abab\underline{b}, ababb\underline{a} \rangle \mapsto \langle \bullet, \bullet, \bullet, \bullet, \bullet, \bullet \rangle$

Properties and Behaviours

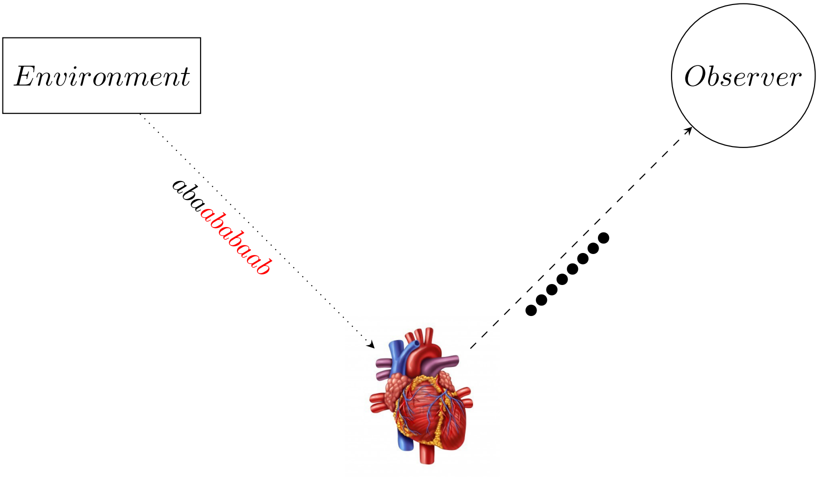
A behaviour is a partition of the set of all input sequences.

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

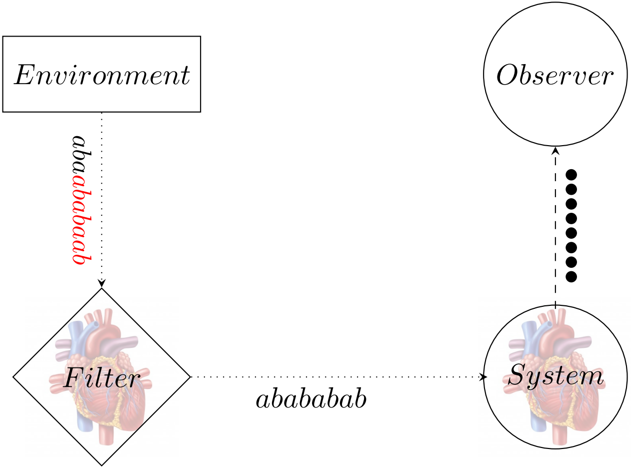
Example: a **healthy heart** should always be in a • state.
Suppose a is diastole and b is systole.

- $abababa$ is good for now.
- abb is not good already.

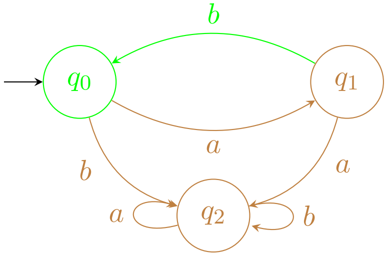
Desired System with "Always •"



What is really going on?



Other Properties



Different properties, same structure: $2^{|States|}$ binary behaviours.

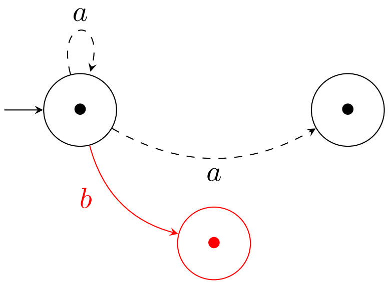
Desired Behaviours and the Desired Property

- Always remain in desired states: $\color{blue}{\bullet}$.
- Avoid undesired states: $\color{blue}{\bullet}^C$.

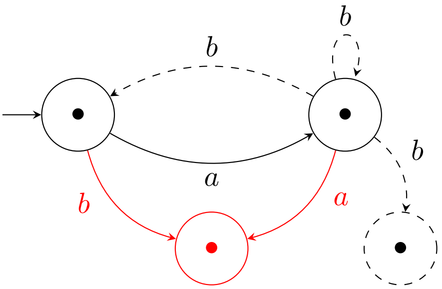
For the heart: $\color{blue}{\bullet} = \bullet$, and $\color{blue}{\bullet}^C = \color{red}{\bullet}$.

Environment may try to put the system in a $\color{blue}{\bullet}^C$ state, and the system should "fight" it, and remain in/return to $\color{blue}{\bullet}$.

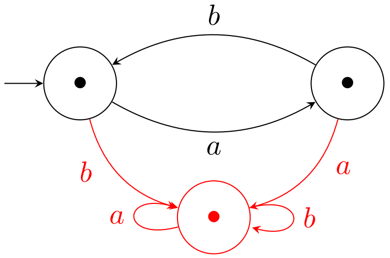
Behaviour Realisation – List of Requirements.

Beh. Req.	Realised System
$\varepsilon \mapsto \bullet$ $a \mapsto \bullet$ $b \mapsto \bullet$	

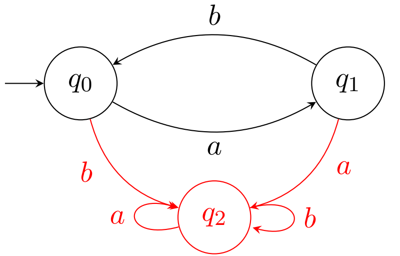
Behaviour Realisation – List of Requirements.

Beh. Req.	Realised System
$\varepsilon \mapsto \bullet$ $a \mapsto \bullet$ $b \mapsto \bullet$ $aa \mapsto \bullet \quad (*)$ $ab \mapsto \bullet \quad (*)$	

Behaviour Realisation – List of Requirements.

Beh. Req.	Realised System
$\varepsilon \mapsto$ ●	
$a \mapsto$ ●	
$b \mapsto$ ●	
$aa \mapsto$ ●	
$ab \mapsto$ ●	
$ba \mapsto$ ●	
$bb \mapsto$ ●	
$\dots$	

Behaviour Realisation – List of Requirements.

Beh. Req.	Realised System
$\varepsilon \mapsto q_0$	
$a \mapsto q_1$	
$b \mapsto q_2$	
$aa \mapsto q_2$	
$ab \mapsto q_0$	
$ba \mapsto q_2$	
$bb \mapsto q_2$	
$\dots$	

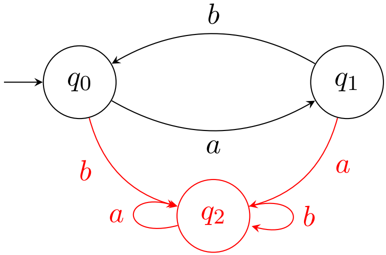
Behaviour Realisation – List of Requirements (2)

$$\Delta\colon \{a, b\}^* \rightarrow \{q_0, q_1, q_2\}$$

$$\begin{aligned} C : \{q_0, q_1, q_2\} &\rightarrow \{\bullet, \color{red}{\bullet}\} \\ q_0 &\mapsto \bullet \\ q_1 &\mapsto \bullet \\ q_2 &\mapsto \color{red}{\bullet} \end{aligned}$$

$$B_E = C \circ \Delta$$

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:

$$
\begin{aligned}
\{a,b\}^{\infty} &\rightarrow X \\
\mathcal{P}(\{a,b\}^{\infty}) &\rightarrow X \\
\mathcal{P}^2(\{a,b\}^{\infty}) &\rightarrow X \\
&\dots \\
\mathcal{P}^n(\{a,b\}^{\infty}) &\rightarrow X
\end{aligned}
$$

Realising Property-based Specifications

Challenges in Realising Property-Based Specifications

Challenge #1: Description of the Desired Property.

$$\bullet = a.b.aa.bb.aaa.bbb.aaaa.bbbb\ldots$$

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, μ-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 $\bullet$,$\color{red}\bullet$,$\color{green}\bullet$ and $\color{brown}\bullet$ allow:

- Intersection: "$\color{red}\bullet$ and $\color{brown}\bullet$". $\color{red}\bullet \color{black}\cap \color{brown}\bullet$
- Union: "$\bullet$ or $\color{green}\bullet$". $(\bullet^C \cap \color{green}\bullet\color{black}^C)^C$
- Difference: "$\color{green}\bullet$, but not $\color{red}\bullet$". $\color{green}\bullet \color{black}\cap \color{red}\bullet\color{black}^C$

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

Worst case: every state is $\color{blue}\bullet$ or every state is $\color{blue}\bullet\color{black}^C$
Trivial behaviour

Challenge #4: Realising the Desired Property.

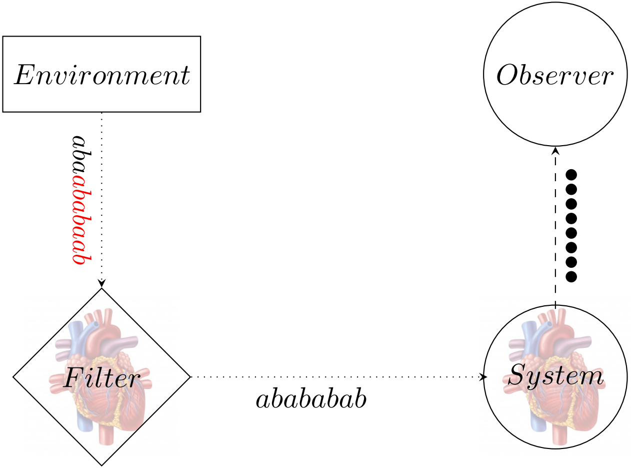
Suppose we overcame Ch. 1, 2 and 3, and we have a desired property $\bullet$.

Is it possible to realise a $System$ that "defends" itself from the environment?

Can $System$ force its next state to be $\bullet$?

- Some actions of the environment cannot be controlled, and force a $\bullet^C$ 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 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!