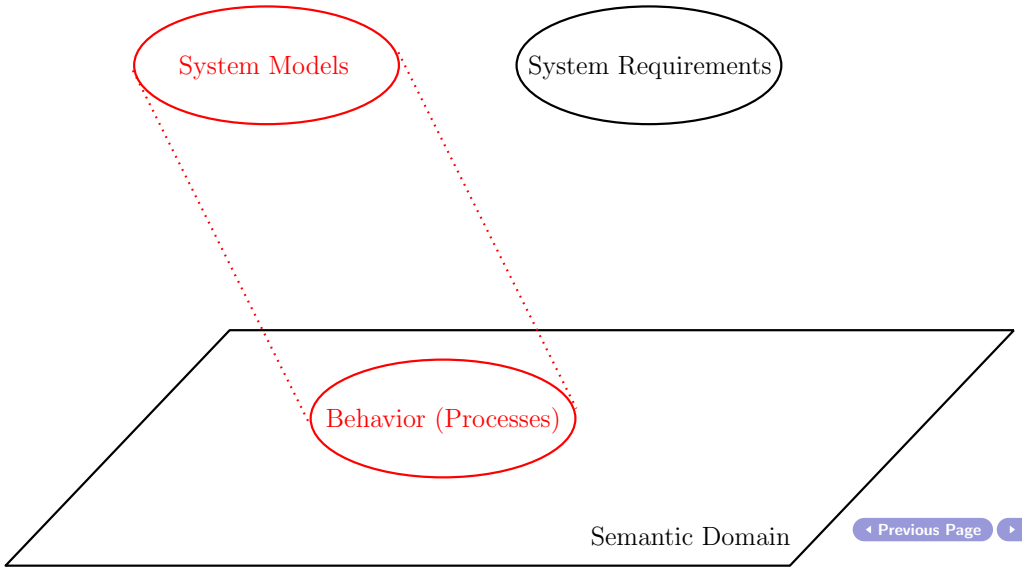
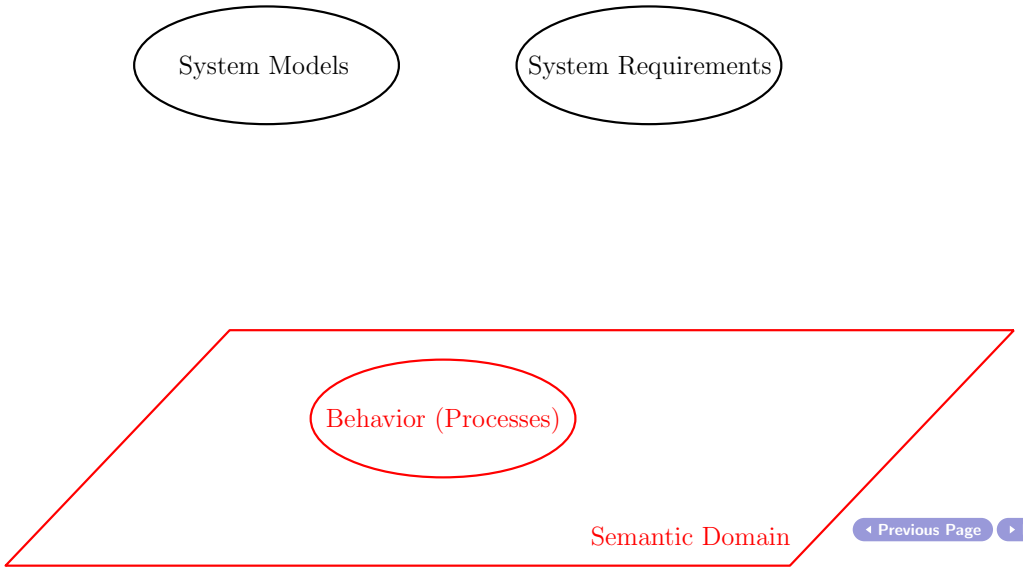
Formal Modelling and Analysis of Concurrent Systems: Actions, Behaviour and Abstraction

Mohammad Mousavi and Jeroen Keiren

General Overview



General Overview



Actions

- Focus on interaction between components

Actions

- Focus on interaction between components
- (Inter)actions are atomic building blocks of models

Actions

- Focus on interaction between components
- (Inter)actions are atomic building blocks of models
- Action can happen internally in a system, or externally with the environment

Actions

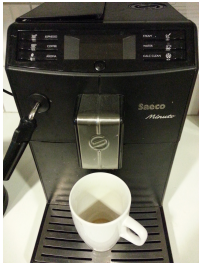
- Focus on interaction between components
- (Inter)actions are atomic building blocks of models
- Action can happen internally in a system, or externally with the environment
- Depends on level of abstraction

Actions

- Focus on interaction between components
- (Inter)actions are atomic building blocks of models
- Action can happen internally in a system, or externally with the environment
- Depends on level of abstraction
- Actions can be composed to obtain behaviour

Actions: An Example

What (inter)actions does a coffee machine have?



Actions: An Example

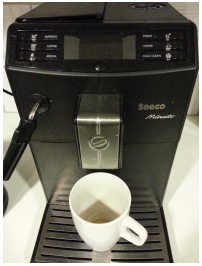
What (inter)actions does a coffee machine have?

- coffee, tea, coin

Actions: An Example

What (inter)actions does a coffee machine have?

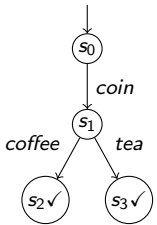
- coffee, tea, coin
- grind



Composing Actions Into Behaviour

- Actions alone do not describe the behaviour of a system!
- We need to add structure and dependencies

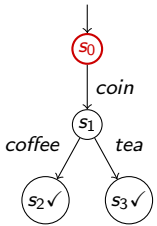
Example (Coffee Machine)



Composing Actions Into Behaviour

- Actions alone do not describe the behaviour of a system!
- We need to add structure and dependencies

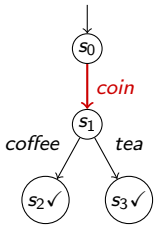
Example (Coffee Machine)



Composing Actions Into Behaviour

- Actions alone do not describe the behaviour of a system!
- We need to add structure and dependencies

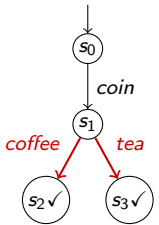
Example (Coffee Machine)



Composing Actions Into Behaviour

- Actions alone do not describe the behaviour of a system!
- We need to add structure and dependencies

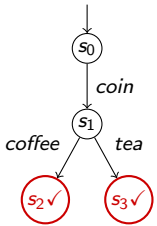
Example (Coffee Machine)



Composing Actions Into Behaviour

- Actions alone do not describe the behaviour of a system!
- We need to add structure and dependencies

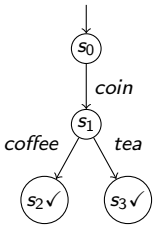
Example (Coffee Machine)

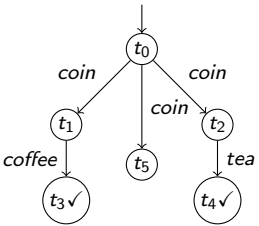


Composing Actions Into Behaviour

- Actions alone do not describe the behaviour of a system!
- We need to add structure and dependencies

Example (Coffee Machine)

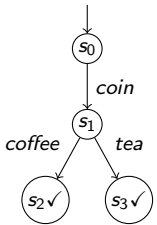


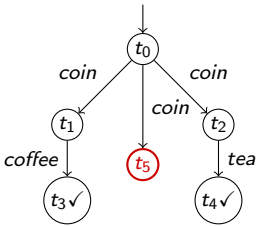


Composing Actions Into Behaviour

- Actions alone do not describe the behaviour of a system!
- We need to add structure and dependencies

Example (Coffee Machine)





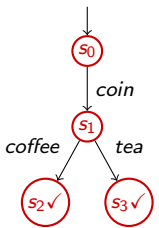
Labelled Transition Systems

An LTS is a 5-tuple $\langle S, Act, \rightarrow, s, T \rangle$:

Labelled Transition Systems

An LTS is a 5-tuple $\langle S, Act, \rightarrow, s, T \rangle$:

- S is a set of *states*

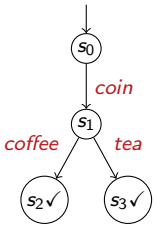


- $S = \{s_0, s_1, s_2, s_3\}$

Labelled Transition Systems

An LTS is a 5-tuple $\langle S, Act, \rightarrow, s, T \rangle$:

- S is a set of *states*
- Act is a set of *(multi-)actions*



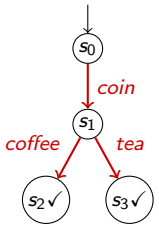
- $S = \{s_0, s_1, s_2, s_3\}$
- $Act = \{coin, coffee, tea\}$

Labelled Transition Systems

An LTS is a 5-tuple $\langle S, Act, \rightarrow, s, T \rangle$:

- S is a set of *states*
- Act is a set of *(multi-)actions*
- $\rightarrow \subseteq S \times Act \times S$ is the *transition relation*

Write $t \xrightarrow{a} t'$ for $(t, a, t') \in \rightarrow$



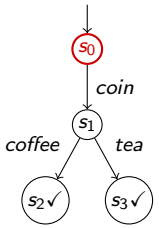
- $S = \{s_0, s_1, s_2, s_3\}$
- $Act = \{coin, coffee, tea\}$
- $\rightarrow = \{(s_0, coin, s_1), (s_1, coffee, s_2), (s_1, tea, s_3)\}$

Labelled Transition Systems

An LTS is a 5-tuple $\langle S, Act, \rightarrow, s, T \rangle$:

- S is a set of *states*
- Act is a set of *(multi-)actions*
- $\rightarrow \subseteq S \times Act \times S$ is the *transition relation*
- $s \in S$ is the *initial* state

Write $t \xrightarrow{a} t'$ for $(t, a, t') \in \rightarrow$, $Act_{\surd}$ for $Act \cup \{\surd\}$



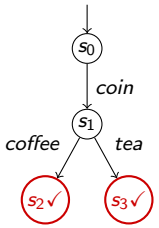
- $S = \{s_0, s_1, s_2, s_3\}$
- $Act = \{coin, coffee, tea\}$
- $\rightarrow = \{(s_0, coin, s_1), (s_1, coffee, s_2), (s_1, tea, s_3)\}$
- $s = s_0$

Labelled Transition Systems

An LTS is a 5-tuple $\langle S, Act, \rightarrow, s, T \rangle$:

- S is a set of *states*
- Act is a set of *(multi-)actions*
- $\rightarrow \subseteq S \times Act \times S$ is the *transition relation*
- $s \in S$ is the *initial* state
- $T \subseteq S$ is the set of *terminating* states

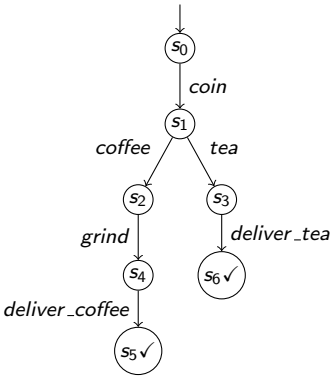
Write $t \xrightarrow{a} t'$ for $(t, a, t') \in \rightarrow$, $Act_{\surd}$ for $Act \cup \{\surd\}$



- $S = \{s_0, s_1, s_2, s_3\}$
- $Act = \{coin, coffee, tea\}$
- $\rightarrow = \{(s_0, coin, s_1), (s_1, coffee, s_2), (s_1, tea, s_3)\}$
- $s = s_0$
- $T = \{s_2, s_3\}$

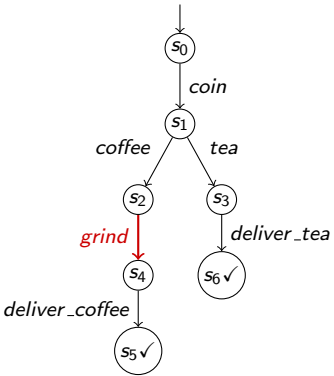
Internal Behaviour

- Some behaviour is internal to the system



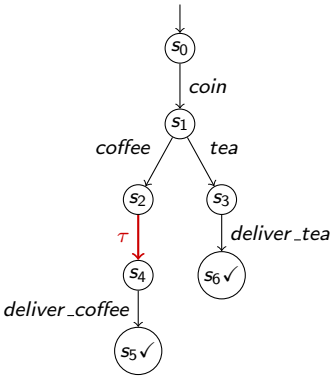
Internal Behaviour

- Some behaviour is internal to the system



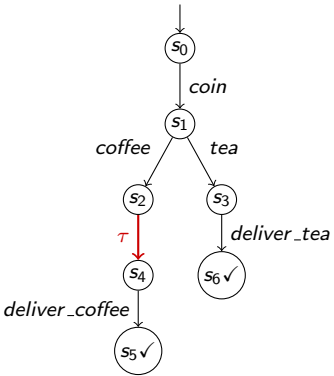
Internal Behaviour

- Some behaviour is internal to the system
- Abstract from it using internal action



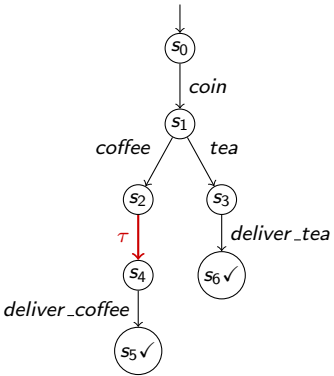
Internal Behaviour

- Some behaviour is internal to the system
- Abstract from it using internal action
- Denoted τ



Internal Behaviour

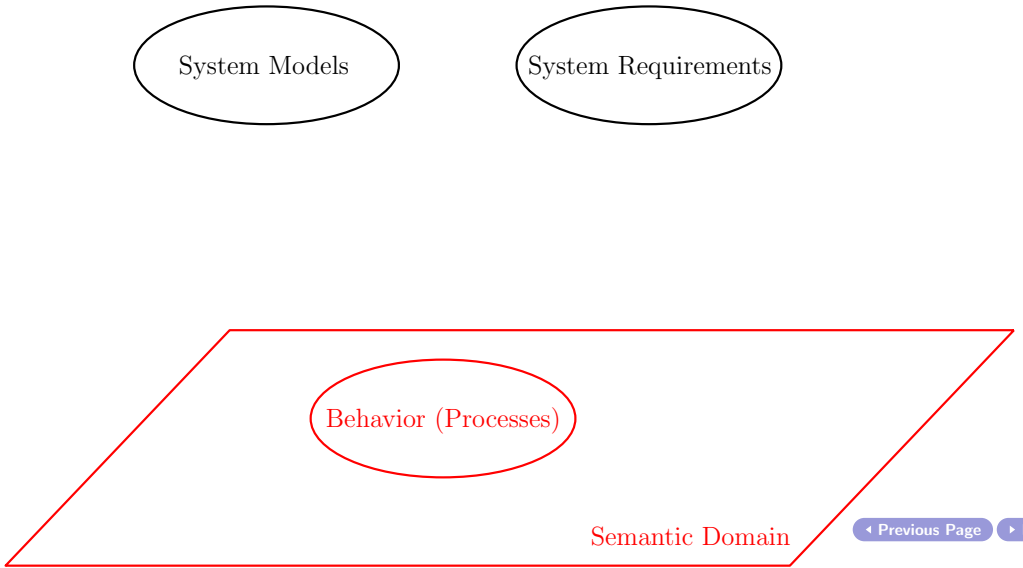
- Some behaviour is internal to the system
- Abstract from it using internal action
- Denoted τ
- $Act_\tau = Act \cup \{\tau\}$



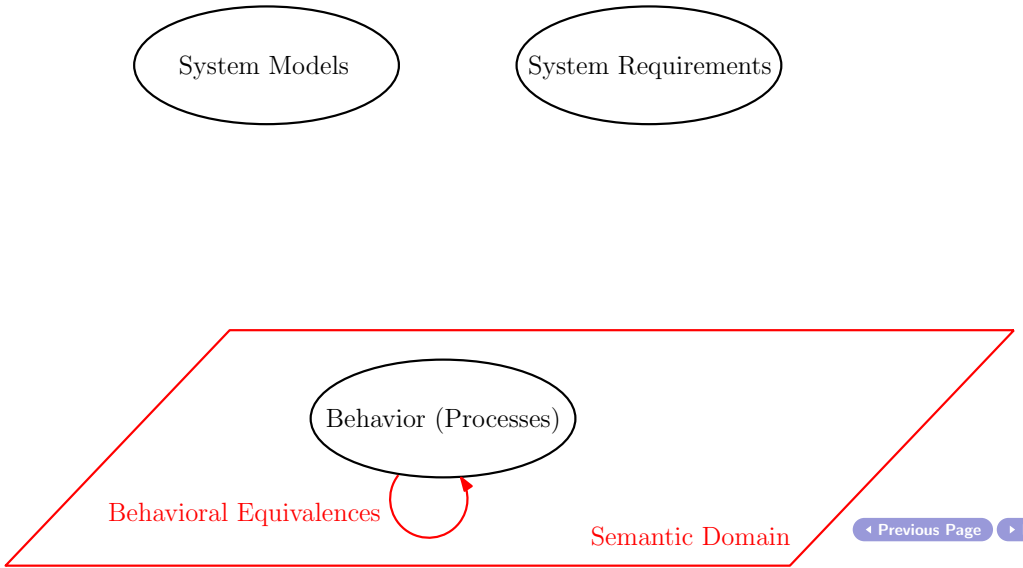
Summary

- Actions main building block
- Behaviour described as Labelled Transition System
- Abstraction of internal actions: τ

General Overview



General Overview



Thank you very much.