Compositional dynamic modelling: a Computer Science perspective

Jane Hillston
The University of Edinburgh

encounters
INSPACE

Models of Computer Systems

Many disciplines recognise the benefits of modelling — in the context of computer science models are generally constructed during the design phase of systems.

Models of Computer Systems

Many disciplines recognise the benefits of modelling — in the context of computer science models are generally constructed during the design phase of systems.

Motivations include:

Models of Computer Systems

Many disciplines recognise the benefits of modelling — in the context of computer science models are generally constructed during the design phase of systems.

Motivations include:

- Checking the system will behave correctly.

Models of Computer Systems

Many disciplines recognise the benefits of modelling — in the context of computer science models are generally constructed during the design phase of systems.

Motivations include:

- Checking the system will behave correctly.
- Analysing the capacity of the system.

Models of Computer Systems

Many disciplines recognise the benefits of modelling — in the context of computer science models are generally constructed during the design phase of systems.

Motivations include:

- Checking the system will behave correctly.
- Analysing the capacity of the system.
- Predicting the performance of the system.

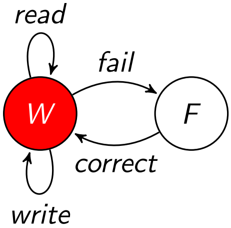
Models of Computer Systems

Many disciplines recognise the benefits of modelling — in the context of computer science models are generally constructed during the design phase of systems.

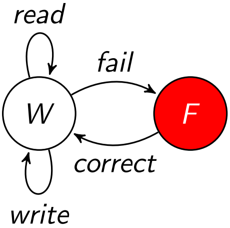
Motivations include:

- Checking the system will behave correctly.
- Analysing the capacity of the system.
- Predicting the performance of the system.

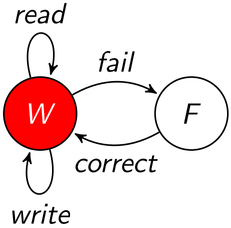
Simple model of Disk Behaviour



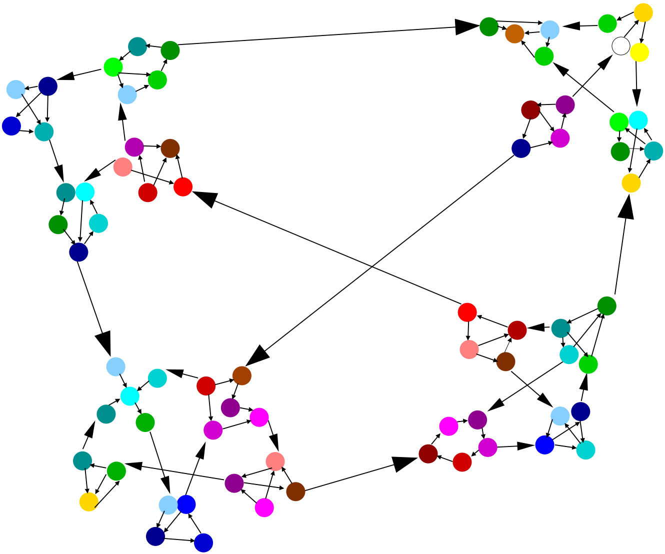
Simple model of Disk Behaviour



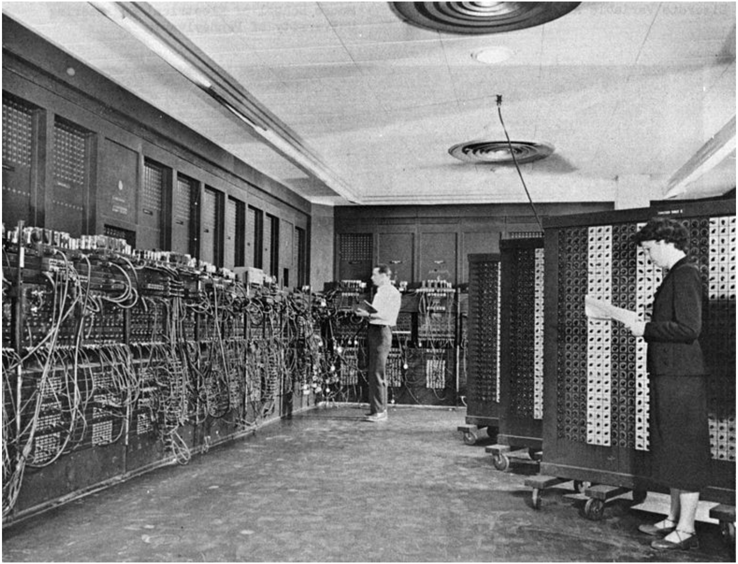
Simple model of Disk Behaviour



Detailed view of Behaviour — State Machine



Advancing Computer Technology

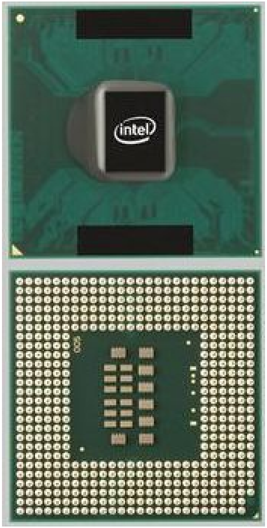


ENIAC c. 1946

Advancing Computer Technology

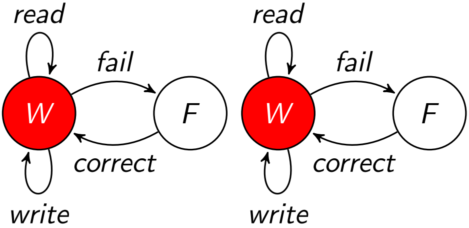
standard laptop c. 2006

Advancing Computer Technology

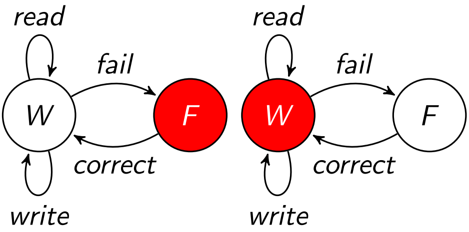


Intel dual core

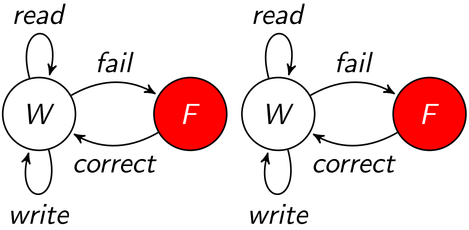
Disks revisited



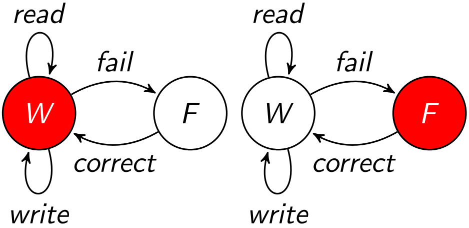
Disks revisited



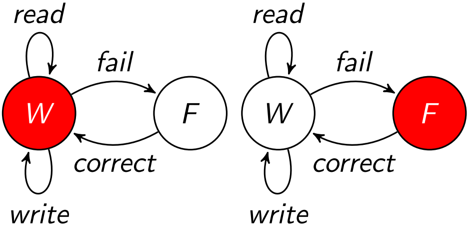
Disks revisited



Disks revisited



Disks revisited



Constructing a model directly in terms of the possible states rapidly becomes a daunting prospect

Advancing Modelling Technology

The complexity and concurrency of a modern computer system would make it infeasible to construct the state machine to model its possible behaviours.

Advancing Modelling Technology

The complexity and concurrency of a modern computer system would make it infeasible to construct the state machine to model its possible behaviours.

Instead language-based modelling techniques which focus on the compositionality and interaction within a system were developed.

Advancing Modelling Technology

The complexity and concurrency of a modern computer system would make it infeasible to construct the state machine to model its possible behaviours.

Instead language-based modelling techniques which focus on the compositionality and interaction within a system were developed.

These are called process algebras and were first developed in the 1980s by Robin Milner and Tony Hoare.

Advancing Modelling Technology

The complexity and concurrency of a modern computer system would make it infeasible to construct the state machine to model its possible behaviours.

Instead language-based modelling techniques which focus on the compositionality and interaction within a system were developed.

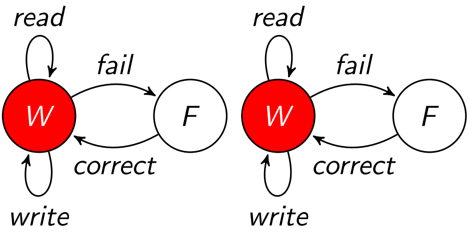
These are called process algebras and were first developed in the 1980s by Robin Milner and Tony Hoare.

Later version of the formalisms included information about the timing and resource use associated with the represented processes: these are stochastic process algebras.

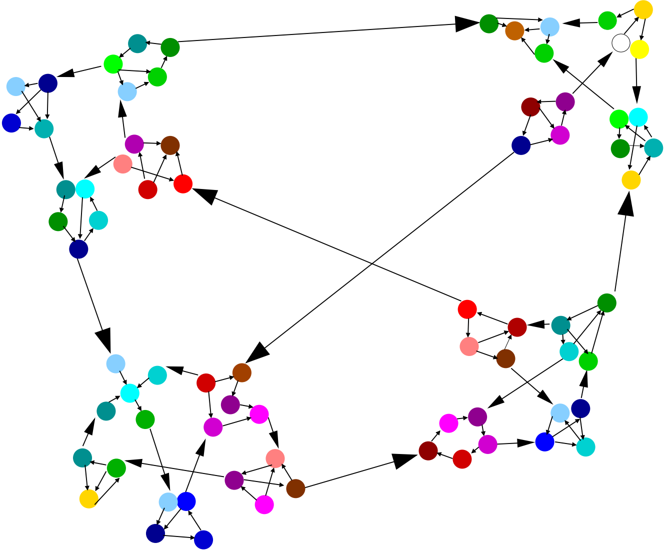
Disks revisited

Disk

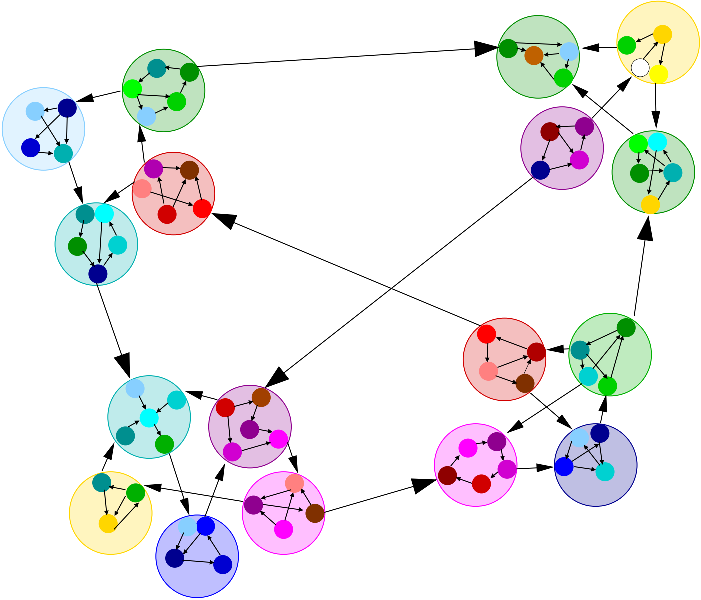
$$
\begin{aligned}
Working &\stackrel{def}{=} (read, r).Working \\
&+ (write, w).Working \\
&+ (fail, f).Failed \\
Failed &\stackrel{def}{=} (correct, c).Working
\end{aligned}
$$



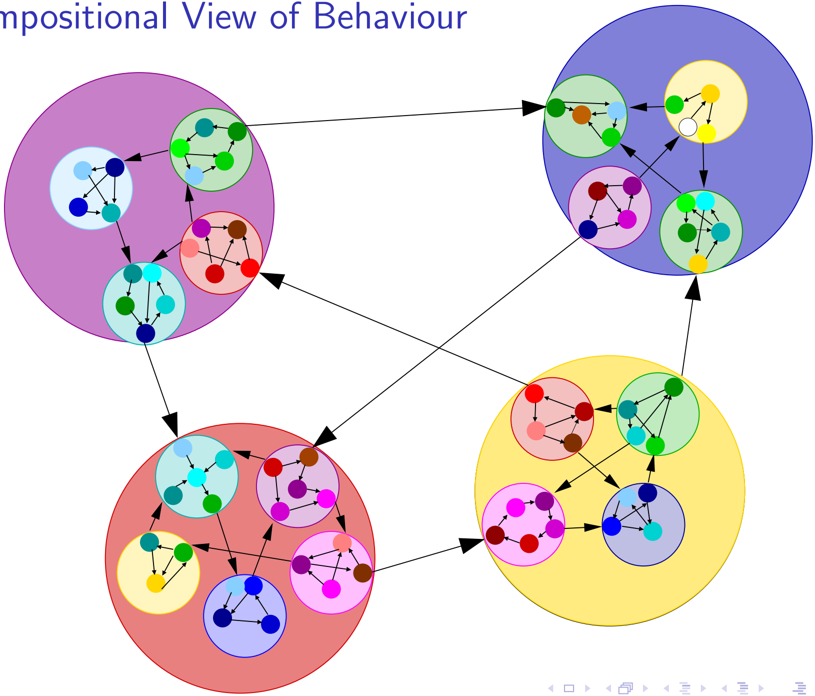
Compositional View of Behaviour



Compositional View of Behaviour



Compositional View of Behaviour



Constructing models from simple components

Client

$$Client_{idle} \stackrel{def}{=} (request, \lambda).Client_{waiting}$$
$$Client_{waiting} \stackrel{def}{=} (response, \top).Client_{idle}$$

Server

$$Server_{idle} \stackrel{def}{=} (request, \top).Server_{computing}$$
$$Server_{computing} \stackrel{def}{=} (compute, \pi).Server_{responding}$$
$$Server_{responding} \stackrel{def}{=} (response, \rho).Server_{idle}$$

System

$$System \stackrel{def}{=} Client_{idle}[3] \underset{\mathcal{L}}{\bowtie} Server_{idle}[2]$$
$$\text{where } \mathcal{L} = \{request, response\}$$

Other applications

The compositional modelling style of process algebras has also been applied to modelling other things as well as computers, most notably biochemical pathways in systems biology.

Other applications

The compositional modelling style of process algebras has also been applied to modelling other things as well as computers, most notably biochemical pathways in systems biology.

There has been substantial work recognising that within a cell there are many molecules interacting through reactions which may be occurring simultaneously.

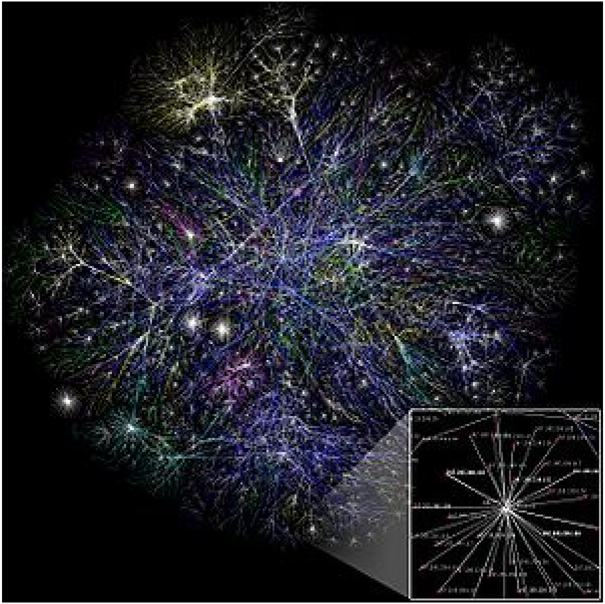
Other applications

The compositional modelling style of process algebras has also been applied to modelling other things as well as computers, most notably biochemical pathways in systems biology.

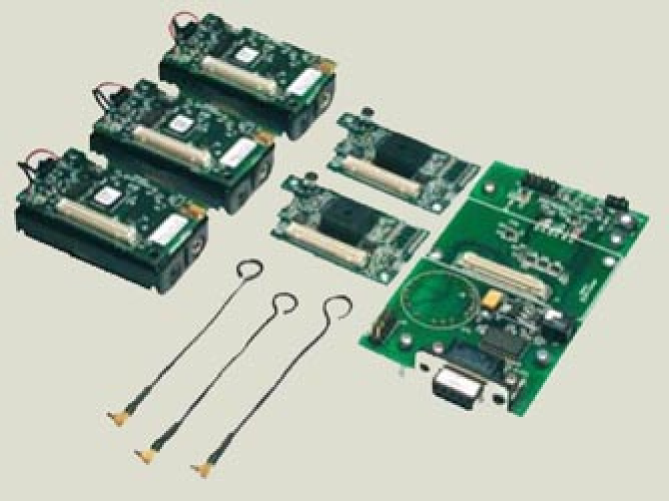
There has been substantial work recognising that within a cell there are many molecules interacting through reactions which may be occurring simultaneously.

In general stochastic process algebras provide a good framework for modelling systems in which the collective behaviour emerges as the result of a large number of individuals, acting and interacting in a predefined, but stochastic, manner.

Internet scale



Mote scale



Smaller and smaller devices create the possibily of a network around each person.

Challenges of modelling ubiquitous systems

Populations of computing entities will be a significant part of our environment, performing tasks that support us, and we shall be largely unaware of them.

Mark Weiser 1994

Challenges of modelling ubiquitous systems

> *Populations of computing entities will be a significant part of our environment, performing tasks that support us, and we shall be largely unaware of them.*

Mark Weiser 1994

The technology is now available to make this vision a reality but as we enter this new informatic era there is a greater need to model computer systems to understand and predict their behaviour.

Challenges of modelling ubiquitous systems

> *Populations of computing entities will be a significant part of our environment, performing tasks that support us, and we shall be largely unaware of them.*

Mark Weiser 1994

The technology is now available to make this vision a reality but as we enter this new informatic era there is a greater need to model computer systems to understand and predict their behaviour.

Unfortunately the scale of the systems is such that our existing modelling techniques are severely challenged by ubiquitous systems.

State-space explosion

Disks	States
1	2

State-space explosion

Disks	States
1	2
2	4

State-space explosion

Disks	States
1	2
2	4
6	64

State-space explosion

Disks	States
1	2
2	4
6	64
10	1024

State-space explosion

Disks	States
1	2
2	4
6	64
10	1024
20	1048576

State-space explosion

Disks	States
1	2
2	4
6	64
10	1024
20	1048576
50	1125899906842624

State-space explosion

Disks	States
1	2
2	4
6	64
10	1024
20	1048576
50	1125899906842624
100	1267650600228229401496703205376

State-space explosion

Disks	States
1	2
2	4
6	64
10	1024
20	1048576
50	1125899906842624
100	1267650600228229401496703205376
150	1427247692705959881058285969449495136382746624

State-space explosion

Disks	States
1	2
2	4
6	64
10	1024
20	1048576
50	1125899906842624
100	1267650600228229401496703205376
150	1427247692705959881058285969449495136382746624

2^{150} states

State-space explosion

Disks	States
1	2
2	4
6	64
10	1024
20	1048576
50	1125899906842624
100	1267650600228229401496703205376
150	1427247692705959881058285969449495136382746624

2^{150} states $= 2^{152}$ bytes

State-space explosion

Disks	States
1	2
2	4
6	64
10	1024
20	1048576
50	1125899906842624
100	1267650600228229401496703205376
150	1427247692705959881058285969449495136382746624

2^{150} states $= 2^{152}$ bytes $= 2^{82} \times 2^{70}$ bytes

State-space explosion

Disks	States
1	2
2	4
6	64
10	1024
20	1048576
50	1125899906842624
100	1267650600228229401496703205376
150	1427247692705959881058285969449495136382746624

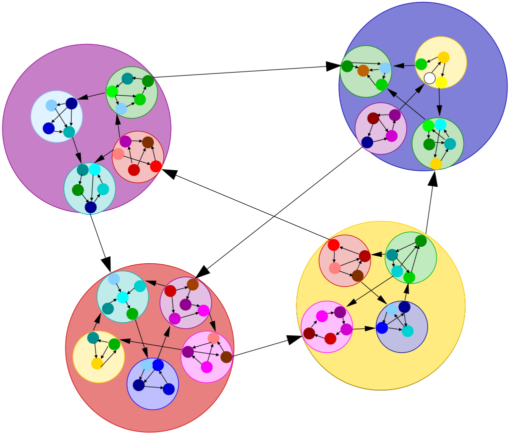
$$2^{150} \text{ states} = 2^{152} \text{ bytes} = 2^{82} \times 2^{70} \text{ bytes} = 2^{82} \text{ zettabytes}$$

Compositionality and Abstraction

One approach is to make use of the compositional structure and abstract away detail of the internal behaviour of components.

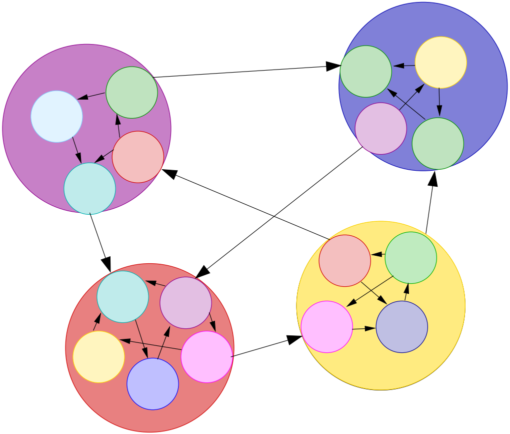
Compositionality and Abstraction

One approach is to make use of the compositional structure and abstract away detail of the internal behaviour of components.



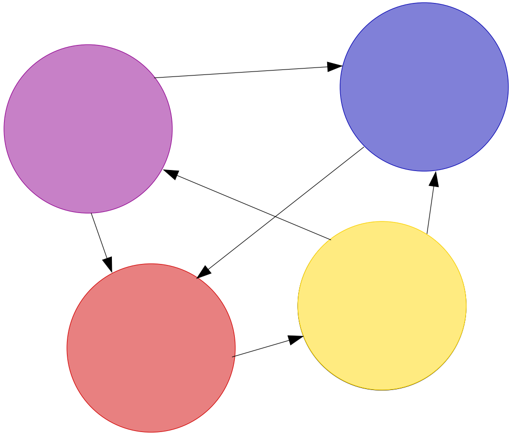
Compositionality and Abstraction

One approach is to make use of the compositional structure and abstract away detail of the internal behaviour of components.



Compositionality and Abstraction

One approach is to make use of the compositional structure and abstract away detail of the internal behaviour of components.



Identity and Individuality

Many of the systems we consider are constructed from many instances of the a set of components.

Identity and Individuality

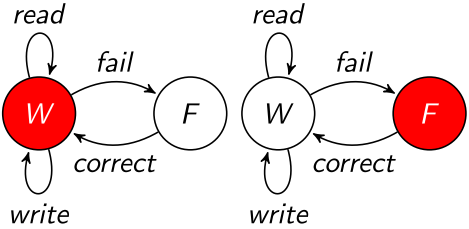
Many of the systems we consider are constructed from many instances of the a set of components.

If we cease to distinguish between instances of components we can form an aggregation which can reduce the state space.

Identity and Individuality

Many of the systems we consider are constructed from many instances of the a set of components.

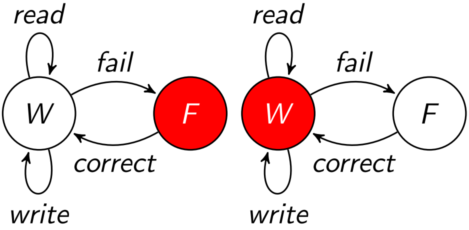
If we cease to distinguish between instances of components we can form an aggregation which can reduce the state space.



Identity and Individuality

Many of the systems we consider are constructed from many instances of the a set of components.

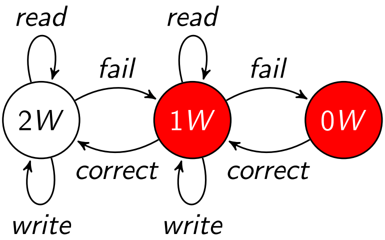
If we cease to distinguish between instances of components we can form an aggregation which can reduce the state space.



Identity and Individuality

Many of the systems we consider are constructed from many instances of the a set of components.

If we cease to distinguish between instances of components we can form an aggregation which can reduce the state space.

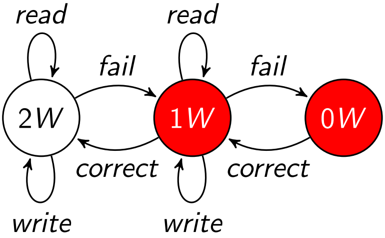


We may choose to disregard the identity of components.

Identity and Individuality

Many of the systems we consider are constructed from many instances of the a set of components.

If we cease to distinguish between instances of components we can form an aggregation which can reduce the state space.



We may choose to disregard the identity of components.

Even better reductions can be achieved when we no longer regard the components as individuals.

Collective Behaviour

In the natural world there are many instances of collective behaviour and its consequences:

Collective Behaviour

In the natural world there are many instances of collective behaviour and its consequences:

Collective Behaviour

In the natural world there are many instances of collective behaviour and its consequences:

Performance as an emergent behaviour

A shift in perspective allows us to model the interactions between individual components but then only the consider the system as a whole as an interaction of populations.

Performance as an emergent behaviour

A shift in perspective allows us to model the interactions between individual components but then only the consider the system as a whole as an interaction of populations.

This allows us to model much larger systems than previously possible but in making the shift we are no longer able to collect any information about individuals in the system.

Performance as an emergent behaviour

A shift in perspective allows us to model the interactions between individual components but then only the consider the system as a whole as an interaction of populations.

This allows us to model much larger systems than previously possible but in making the shift we are no longer able to collect any information about individuals in the system.

We must instead think about the performance of the collective point of view. Service providers often want to do this in any case. For example making contracts in terms of service level agreements.

Performance as an emergent behaviour

A shift in perspective allows us to model the interactions between individual components but then only the consider the system as a whole as an interaction of populations.

This allows us to model much larger systems than previously possible but in making the shift we are no longer able to collect any information about individuals in the system.

We must instead think about the performance of the collective point of view. Service providers often want to do this in any case. For example making contracts in terms of service level agreements.

Example Service Level Agreement

90% of requests receive a response within 3 seconds.

Disk model in PEPA

Disk

$$\begin{aligned} \mathit{Working} &\stackrel{\mathit{def}}{=} (\mathit{read}, r).\mathit{Working} \\ &+ (\mathit{write}, w).\mathit{Working} \\ &+ (\mathit{fail}, f).\mathit{Failed} \\ \mathit{Failed} &\stackrel{\mathit{def}}{=} (\mathit{correct}, c).\mathit{Working} \end{aligned}$$

- We have W working disks and F failed ($W + F = N$).

Disk model in PEPA

Disk

$$
\begin{aligned}
Working &\stackrel{def}{=} (read, r).Working \\
&+ (write, w).Working \\
&+ (fail, f).Failed \\
Failed &\stackrel{def}{=} (correct, c).Working
\end{aligned}
$$

- We have W working disks and F failed $(W + F = N)$.
- Working disks fail at rate $f \times W$.

Disk model in PEPA

Disk

$$
\begin{aligned}
Working &\stackrel{def}{=} (read, r).Working \\
&+ (write, w).Working \\
&+ (fail, f).Failed \\
Failed &\stackrel{def}{=} (correct, c).Working
\end{aligned}
$$

- We have W working disks and F failed $(W + F = N)$.
- Working disks fail at rate $f \times W$.
- Failures are corrected at rate $c \times F$.

Disk model in PEPA

Disk

$$
\begin{aligned}
Working &\stackrel{def}{=} (read, r).Working \\
&+ (write, w).Working \\
&+ (fail, f).Failed \\
Failed &\stackrel{def}{=} (correct, c).Working
\end{aligned}
$$

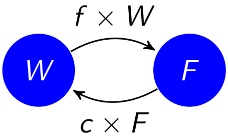
- We have W working disks and F failed ($W + F = N$).
- Working disks fail at rate $f \times W$.
- Failures are corrected at rate $c \times F$.

Disk model in PEPA

Disk

$$
\begin{aligned}
Working &\stackrel{def}{=} (read, r).Working \\
&+ (write, w).Working \\
&+ (fail, f).Failed \\
Failed &\stackrel{def}{=} (correct, c).Working
\end{aligned}
$$

- We have W working disks and F failed $(W + F = N)$.
- Working disks fail at rate $f \times W$.
- Failures are corrected at rate $c \times F$.



$$
\begin{aligned}
\mathrm{d}W/\mathrm{d}t &= -f \times W + c \times F \\
\mathrm{d}F/\mathrm{d}t &= f \times W - c \times F
\end{aligned}
$$

Conclusions

Over the last decade stochastic process algebra have been successful in modelling a wide range of computer systems and their performance.

Conclusions

Over the last decade stochastic process algebra have been successful in modelling a wide range of computer systems and their performance.

The compositional style supports representation of large systems.

Conclusions

Over the last decade stochastic process algebra have been successful in modelling a wide range of computer systems and their performance.

The compositional style supports representation of large systems.

The shift to the collective dynamic view supports analysis of very large systems.

Conclusions

Over the last decade stochastic process algebra have been successful in modelling a wide range of computer systems and their performance.

The compositional style supports representation of large systems.

The shift to the collective dynamic view supports analysis of very large systems.

Nevertheless there are significant challenges ahead as ubiquitous systems become a reality.

Acknowledgements

Stochastic process algebra group and collaborators at Edinburgh

Allan Clark, Jie Ding, Adam Duguid, Vashti Galpin, Stephen Gilmore, Maria Luisa Guerriero, Jane Hillston, Michael Smith, Mirco Tribastone.

Stochastic process algebra community and collaborators elsewhere

Soufiene Benkrine, Jeremy Bradley, Federica Ciocchetta, Nick Dingle, Peter Harrison, Richard Hayden, William Knottenbelt, Carron Shankland, Nigel Thomas, Yishi Zhao.