Challenges for Quantitative Analysis of Collective Adaptive Systems

Jane Hillston
LFCS, University of Edinburgh

August 31st 2013

Outline

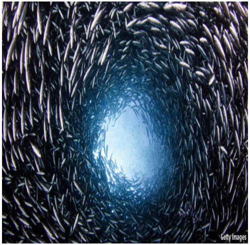
Collective Systems

We are surrounded by examples of collective systems:

Collective Systems

We are surrounded by examples of collective systems:

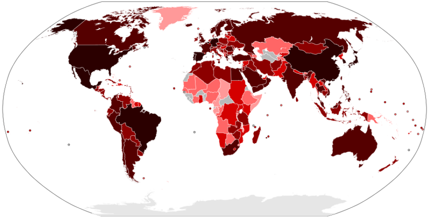
in the natural world ....



Collective Systems

We are surrounded by examples of collective systems:

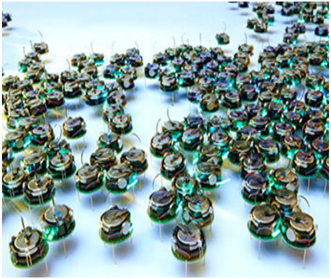
.... and in the man-made world



Collective Systems

We are surrounded by examples of collective systems:

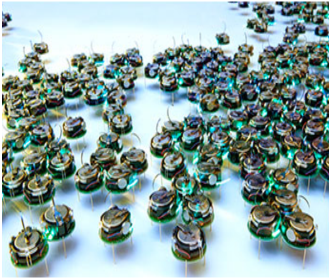
.... and in the man-made world



Collective Systems

We are surrounded by examples of collective systems:

.... and in the man-made world



Most of these systems are also adaptive to their environment

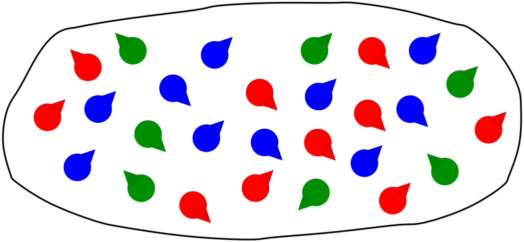
Collective Systems

We are surrounded by examples of collective systems:

an informatic environment

Collective Adaptive Systems

From a computer science perspective these systems can be viewed as being made up of a large number of interacting entities.

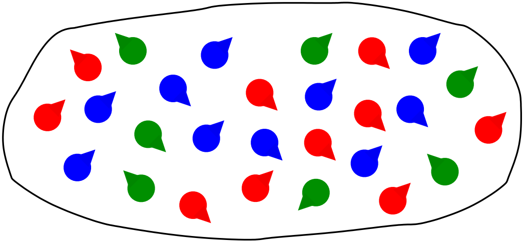


Each entity may have its own properties, objectives and actions.

At the system level these combine to create the collective behaviour.

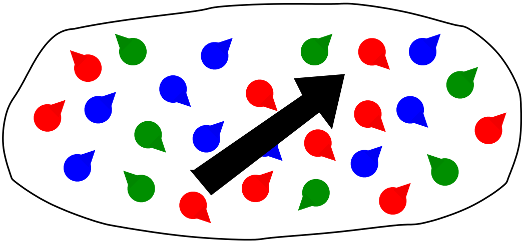
Collective Adaptive Systems

The behaviour of the system is thus dependent on the behaviour of the individual entities.



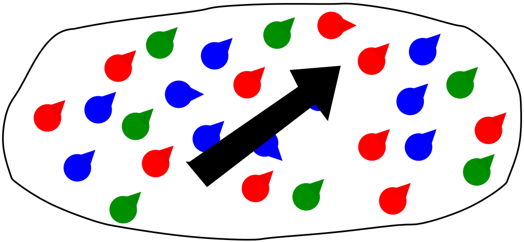
Collective Adaptive Systems

The behaviour of the system is thus dependent on the behaviour of the individual entities.



Collective Adaptive Systems

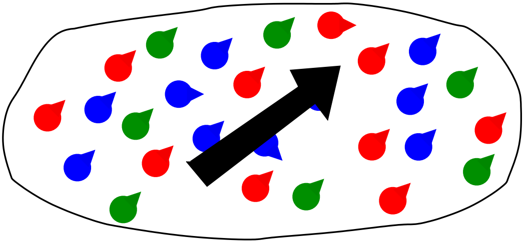
The behaviour of the system is thus dependent on the behaviour of the individual entities.



And the behaviour of the individuals will be influenced by the state of the overall system.

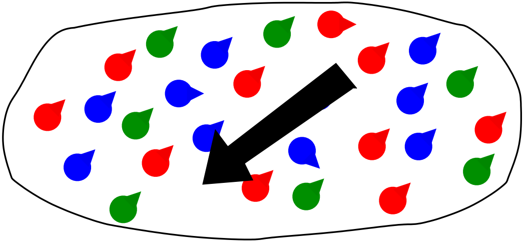
Collective Adaptive Systems

Such systems are often embedded in our environment and need to operate without centralised control or direction.



Collective Adaptive Systems

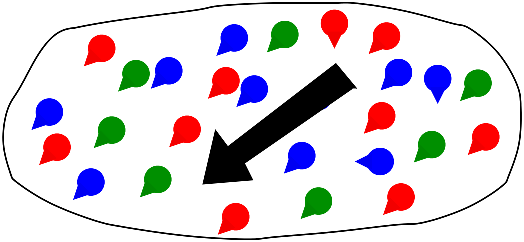
Such systems are often embedded in our environment and need to operate without centralised control or direction.



Moreover when conditions within the system change it may not be feasible to have human intervention to adjust behaviour appropriately.

Collective Adaptive Systems

Such systems are often embedded in our environment and need to operate without centralised control or direction.



Moreover when conditions within the system change it may not be feasible to have human intervention to adjust behaviour appropriately.

Thus systems must be able to autonomously adapt.

The Informatic Environment

Robin Milner coined the term of informatics environment, in which pervasive computing elements are embedded in the human environment, invisibly providing services and responding to requirements.

The Informatic Environment

Robin Milner coined the term of informatics environment, in which pervasive computing elements are embedded in the human environment, invisibly providing services and responding to requirements.

Such systems are now becoming the reality, and many form collective adaptive systems, in which large numbers of computing elements collaborate to meet the human need.

The Informatic Environment

Robin Milner coined the term of informatics environment, in which pervasive computing elements are embedded in the human environment, invisibly providing services and responding to requirements.

Such systems are now becoming the reality, and many form collective adaptive systems, in which large numbers of computing elements collaborate to meet the human need.

For instance, may examples of such systems can be found in components of Smart Cities, such as smart urban transport and smart grid electricity generation and storage.

Quantitative Modelling

Performance modelling aims to construct models of the dynamic behaviour of systems in order to support the fair and efficient sharing of resources.

Quantitative Modelling

Performance modelling aims to construct models of the dynamic behaviour of systems in order to support the fair and efficient sharing of resources.

Markovian-based discrete event models have been applied to computer systems since the mid-1960s and communication systems since the early 20th century.

Quantitative Modelling

Performance modelling aims to construct models of the dynamic behaviour of systems in order to support the fair and efficient sharing of resources.

Markovian-based discrete event models have been applied to computer systems since the mid-1960s and communication systems since the early 20th century.

Originally queueing networks were primarily used to construct models, and sophisticated analysis techniques were developed.

Quantitative Modelling

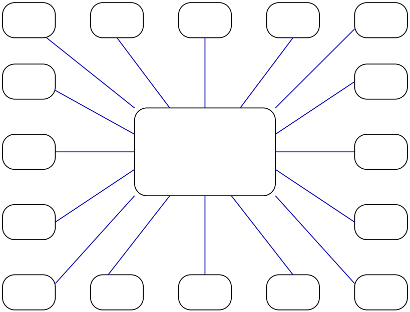
Performance modelling aims to construct models of the dynamic behaviour of systems in order to support the fair and efficient sharing of resources.

Markovian-based discrete event models have been applied to computer systems since the mid-1960s and communication systems since the early 20th century.

Originally queueing networks were primarily used to construct models, and sophisticated analysis techniques were developed.

These techniques are no longer widely applicable for expressing the dynamic behaviour observed in distributed systems, and this is even more true of systems with collective adaptive behaviour.

Performance Modelling: Motivation



Capacity planning

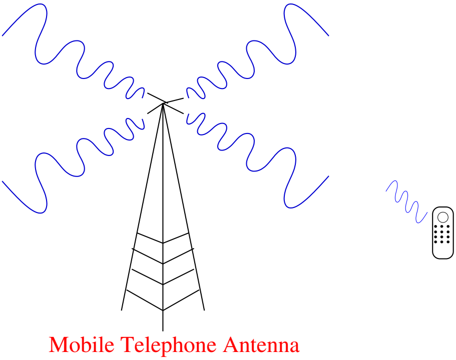
- How many clients can the existing server support and maintain reasonable response times?

Performance Modelling: Motivation

Capacity planning

- How many buses do I need to maintain service at peak time in a smart urban transport system?

Performance Modelling: Motivation



System Configuration

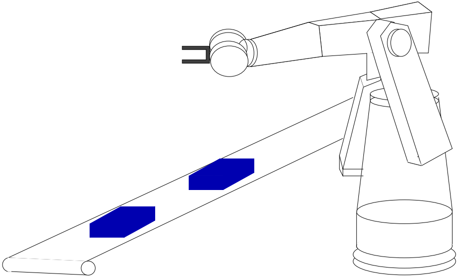
- How many frequencies do you need to keep blocking probabilities low?

Performance Modelling: Motivation

System Configuration

- What capacity do I need at bike stations to minimise the movement of bikes by truck?

Performance Modelling: Motivation



System Tuning

- What speed of conveyor belt will minimize robot idle time and maximize throughput whilst avoiding lost widgets?

Performance Modelling: Motivation

System Tuning

- What strategy can I use to maintain supply-demand balance within a smart electricity grid?

Performance Modelling

The size and complexity of real systems makes the direct construction of discrete state models costly and error-prone.

Performance Modelling

The size and complexity of real systems makes the direct construction of discrete state models costly and error-prone.

For the last three decades there has been substantial interest in applying formal modelling techniques enhanced with information about timing and probability.

Performance Modelling

The size and complexity of real systems makes the direct construction of discrete state models costly and error-prone.

For the last three decades there has been substantial interest in applying formal modelling techniques enhanced with information about timing and probability.

From these high-level system descriptions the underlying mathematical model (Continuous Time Markov Chain (CTMC)) can be automatically generated.

Performance Modelling

The size and complexity of real systems makes the direct construction of discrete state models costly and error-prone.

For the last three decades there has been substantial interest in applying formal modelling techniques enhanced with information about timing and probability.

From these high-level system descriptions the underlying mathematical model (Continuous Time Markov Chain (CTMC)) can be automatically generated.

Primary examples include:

- Stochastic Petri Nets and
- Stochastic/Markovian Process Algebras.

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

$$(\alpha, r).P$$

action type
or name

activity rate
(parameter of an
exponential distribution)

component/
derivative

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

$$(\alpha,\ r).P$$

action type
or name

activity rate
(parameter of an
exponential distribution)

component/
derivative

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

$$(\alpha,\ r).P$$

action type or name

component/ derivative

activity rate (parameter of an exponential distribution)

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

$$(\alpha,\ r).P$$

action type
or name

activity rate
(parameter of an
exponential distribution)

component/
derivative

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

$$(\alpha, r).P$$

action type
or name

activity rate
(parameter of an
exponential distribution)

component/
derivative

- The language is used to generate a CTMC for performance modelling.

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

$$(\alpha, r).P$$

action type or name

activity rate (parameter of an exponential distribution)

component/ derivative

- The language is used to generate a CTMC for performance modelling.

SPA MODEL

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

$$(\alpha,\ r).P$$

action type
or name

activity rate
(parameter of an
exponential distribution)

component/
derivative

- The language is used to generate a CTMC for performance modelling.

SPA
MODEL $\xrightarrow{\text{SOS rules}}$

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

$$(\alpha, r).P$$

action type or name

activity rate (parameter of an exponential distribution)

component/ derivative

- The language is used to generate a CTMC for performance modelling.

$$\text{SPA MODEL} \xrightarrow{\text{SOS rules}} \text{LABELLED TRANSITION SYSTEM}$$

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

$$(\alpha, r).P$$

action type or name

activity rate (parameter of an exponential distribution)

component/ derivative

- The language is used to generate a CTMC for performance modelling.

SPA MODEL $\xrightarrow{\text{SOS rules}}$ LABELLED TRANSITION SYSTEM $\xrightarrow[\text{diagram}]{\text{state transition}}$

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

$$(\alpha, r).P$$

action type or name

activity rate (parameter of an exponential distribution)

component/ derivative

- The language is used to generate a CTMC for performance modelling.

SPA MODEL $\xrightarrow{\text{SOS rules}}$ LABELLED TRANSITION SYSTEM $\xrightarrow[\text{diagram}]{\text{state transition}}$ CTMC $\mathbf{Q}$

Performance Evaluation Process Algebra

$(\alpha, f).P$	Prefix
$P_1 + P_2$	Choice
$P_1 \bowtie_L P_2$	Co-operation
P/L	Hiding
X	Constant

Performance Evaluation Process Algebra

$(\alpha, f).P$	Prefix
$P_1 + P_2$	Choice
$P_1 \underset{L}{\bowtie} P_2$	Co-operation
P/L	Hiding
X	Constant

Performance Evaluation Process Algebra

$(\alpha, f).P$	Prefix
$P_1 + P_2$	Choice
$P_1 \underset{L}{\bowtie} P_2$	Co-operation
P/L	Hiding
X	Constant

Performance Evaluation Process Algebra

$(\alpha, f).P$	Prefix
$P_1 + P_2$	Choice
$P_1 \bowtie_{L} P_2$	Co-operation
P/L	Hiding
X	Constant

Performance Evaluation Process Algebra

$(\alpha, f).P$	Prefix
$P_1 + P_2$	Choice
$P_1 \underset{L}{\bowtie} P_2$	Co-operation
P/L	Hiding
X	Constant

Performance Evaluation Process Algebra

$(\alpha, f).P$	Prefix
$P_1 + P_2$	Choice
$P_1 \bowtie_{L} P_2$	Co-operation
P/L	Hiding
X	Constant

Integrated analysis

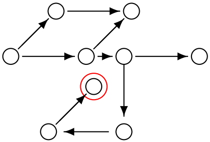
Qualitative verification can now be complemented by quantitative verification.

Integrated analysis

Qualitative verification can now be complemented by quantitative verification.

Reachability analysis

How long will it take for the system to arrive in a particular state?

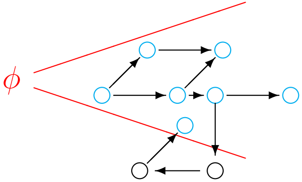


Integrated analysis

Qualitative verification can now be complemented by quantitative verification.

Model checking

Does a given property ϕ hold within the system with a given probability?

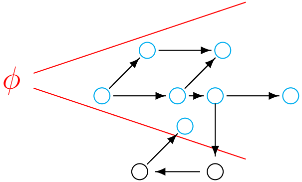


Integrated analysis

Qualitative verification can now be complemented by quantitative verification.

Model checking

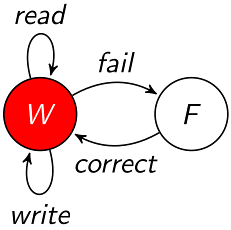
For a given starting state
how long is it until
a given property ϕ holds?



Discrete event modelling

At the basic level a discrete event model captures the states of the system and the events that cause transitions between states.

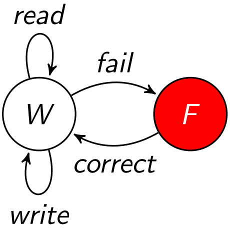
Consider a simple model of a disk:



Discrete event modelling

At the basic level a discrete event model captures the states of the system and the events that cause transitions between states.

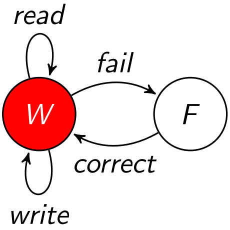
Consider a simple model of a disk:



Discrete event modelling

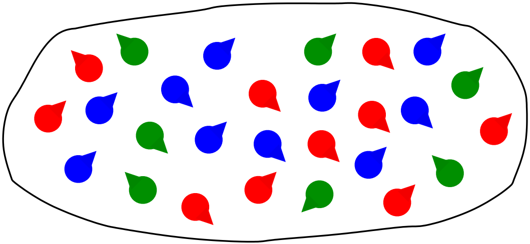
At the basic level a discrete event model captures the states of the system and the events that cause transitions between states.

Consider a simple model of a disk:



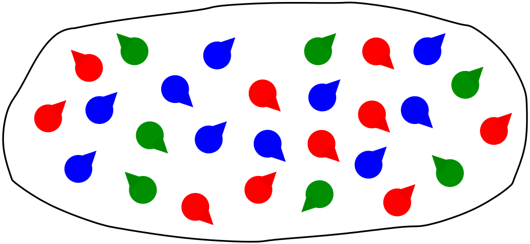
Modelling collective behaviour

A key feature of collective systems is the existence of populations of entities who share certain characteristics.



Modelling collective behaviour

A key feature of collective systems is the existence of populations of entities who share certain characteristics.

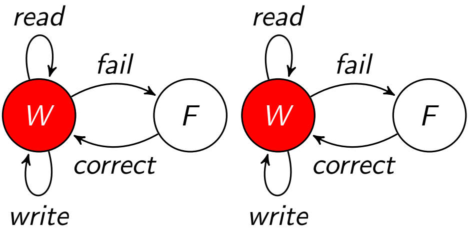


High-level modelling formalisms allow this repetition to be captured at the high-level rather than explicitly.

Disks revisited

Disk

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



Process Algebra for CAS

Process algebra are well-suited for constructing models of CAS:

Process Algebra for CAS

Process algebra are well-suited for constructing models of CAS:

- Developed to represent concurrent behaviour compositionally;

Process Algebra for CAS

Process algebra are well-suited for constructing models of CAS:

- Developed to represent concurrent behaviour compositionally;
- Represent the interactions between individuals explicitly;

Process Algebra for CAS

Process algebra are well-suited for constructing models of CAS:

- Developed to represent concurrent behaviour compositionally;
- Represent the interactions between individuals explicitly;
- Stochastic extensions allow the dynamics of system behaviour to be captured;

Process Algebra for CAS

Process algebra are well-suited for constructing models of CAS:

- Developed to represent concurrent behaviour compositionally;
- Represent the interactions between individuals explicitly;
- Stochastic extensions allow the dynamics of system behaviour to be captured;
- State-dependent functional rates allow adaptation to be captured;

Process Algebra for CAS

Process algebra are well-suited for constructing models of CAS:

- Developed to represent concurrent behaviour compositionally;
- Represent the interactions between individuals explicitly;
- Stochastic extensions allow the dynamics of system behaviour to be captured;
- State-dependent functional rates allow adaptation to be captured;
- Incorporate formal apparatus for reasoning about the behaviour of systems.

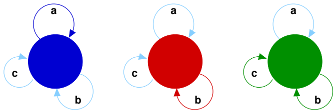
Process Algebra for CAS

Process algebra are well-suited for constructing models of CAS:

- Developed to represent concurrent behaviour compositionally;
- Represent the interactions between individuals explicitly;
- Stochastic extensions allow the dynamics of system behaviour to be captured;
- State-dependent functional rates allow adaptation to be captured;
- Incorporate formal apparatus for reasoning about the behaviour of systems.

Recent advances in analysis techniques for process algebras have made it possible to study such systems even when the number of entities and activities become huge.

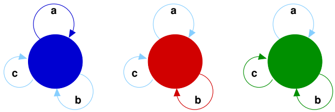
Solving discrete state models

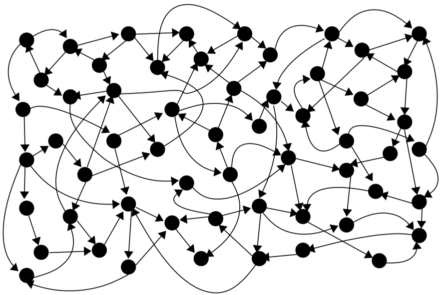


Under the SOS semantics a SPA model is mapped to a CTMC with global states determined by the local states of all the participating components.

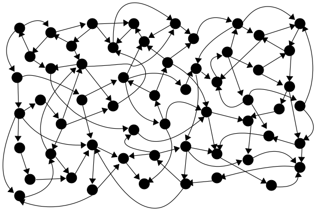
Solving discrete state models

Under the SOS semantics a SPA model is mapped to a CTMC with global states determined by the local states of all the participating components.



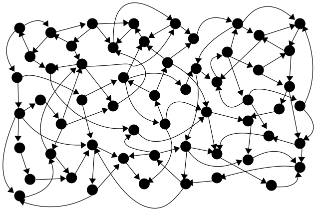


Solving discrete state models



When the size of the state space is not too large they are amenable to numerical solution (linear algebra) to determine a steady state or transient probability distribution.

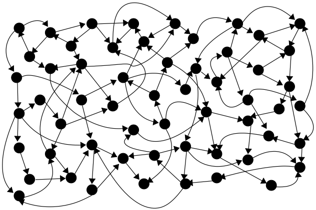
Solving discrete state models



When the size of the state space is not too large they are amenable to numerical solution (linear algebra) to determine a steady state or transient probability distribution.

$$Q = \begin{pmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,N} \\ q_{2,1} & q_{2,2} & \cdots & q_{2,N} \\ \vdots & \vdots & & \vdots \\ q_{N,1} & q_{N,2} & \cdots & q_{N,N} \end{pmatrix}$$

Solving discrete state models



When the size of the state space is not too large they are amenable to numerical solution (linear algebra) to determine a steady state or transient probability distribution.

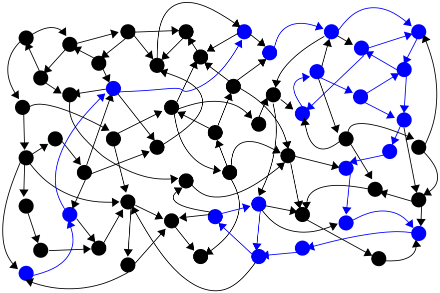
$$Q = \begin{pmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,N} \\ q_{2,1} & q_{2,2} & \cdots & q_{2,N} \\ \vdots & \vdots & & \vdots \\ q_{N,1} & q_{N,2} & \cdots & q_{N,N} \end{pmatrix}$$

$$\pi(t) = (\pi_1(t), \pi_2(t), \ldots, \pi_N(t))$$

$$\pi(\infty)Q = 0$$

Solving discrete state models

Alternatively they may be studied using stochastic simulation. Each run generates a single trajectory through the state space. Many runs are needed in order to obtain average behaviours.



State space explosion

As the size of the state space becomes large it becomes infeasible to carry out numerical solution and extremely time-consuming to conduct stochastic simulation.

Analysing collective behaviour

Process algebra models typically give rise to discrete state mathematical representations, where a state of the whole system is defined in terms of the state of each of the individual entities in the system.

Analysing collective behaviour

Process algebra models typically give rise to discrete state mathematical representations, where a state of the whole system is defined in terms of the state of each of the individual entities in the system.

This rapidly leads to enormous numbers of states which are computationally expensive, or even prohibitive, to explore.

Analysing collective behaviour

Process algebra models typically give rise to discrete state mathematical representations, where a state of the whole system is defined in terms of the state of each of the individual entities in the system.

This rapidly leads to enormous numbers of states which are computationally expensive, or even prohibitive, to explore.

Our approach to analysing collective behaviour is to make a counting abstraction and view the system not in terms of the individual components but in terms of proportions within the subpopulations.

Identity and Individuality

Collective systems are constructed from many instances of a set of components.

Identity and Individuality

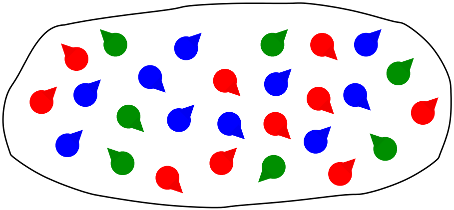
Collective systems are constructed from many instances of a set of components.

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

Identity and Individuality

Collective systems are constructed from many instances of a set of components.

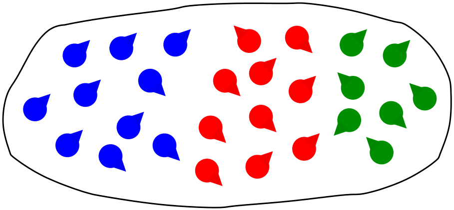
If we cease to distinguish between instances of components we can form an aggregation which can reduce the state space.



Identity and Individuality

Collective systems are constructed from many instances of a set of components.

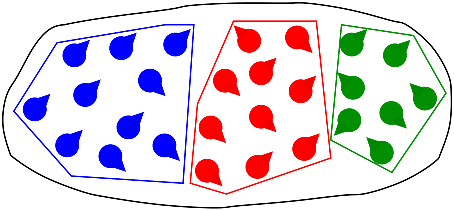
If we cease to distinguish between instances of components we can form an aggregation which can reduce the state space.



Identity and Individuality

Collective systems are constructed from many instances of a set of components.

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

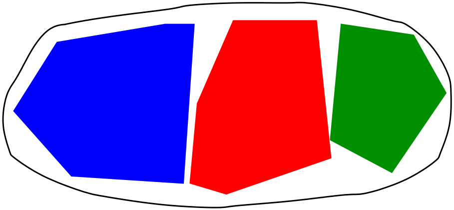


We may choose to disregard the identity of components.

Identity and Individuality

Collective systems are constructed from many instances of 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.

Population statistics: emergent behaviour

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

Population statistics: emergent behaviour

A shift in perspective allows us to model the interactions between individual components but then only 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.

Population statistics: emergent behaviour

A shift in perspective allows us to model the interactions between individual components but then only 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.

To characterise the behaviour of a population we calculate the proportion of individuals within the population that are exhibiting certain behaviours rather than tracking individuals directly.

Population statistics: emergent behaviour

A shift in perspective allows us to model the interactions between individual components but then only 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.

To characterise the behaviour of a population we calculate the proportion of individuals within the population that are exhibiting certain behaviours rather than tracking individuals directly.

Furthermore we make a continuous approximation of how the proportions vary over time.

Continuous Approximation

This means shifting to a different mathematical representation, where we no longer keep track of the individual states of each entity.

Continuous Approximation

This means shifting to a different mathematical representation, where we no longer keep track of the individual states of each entity.

As we are focussed instead of proportions within populations we now treat these variables as continuous rather than discrete.

Continuous Approximation

This means shifting to a different mathematical representation, where we no longer keep track of the individual states of each entity.

As we are focussed instead of proportions within populations we now treat these variables as continuous rather than discrete.

Continuous Approximation

This means shifting to a different mathematical representation, where we no longer keep track of the individual states of each entity.

As we are focussed instead of proportions within populations we now treat these variables as continuous rather than discrete.

Continuous Approximation

This means shifting to a different mathematical representation, where we no longer keep track of the individual states of each entity.

As we are focussed instead of proportions within populations we now treat these variables as continuous rather than discrete.

Continuous Approximation

This means shifting to a different mathematical representation, where we no longer keep track of the individual states of each entity.

As we are focussed instead of proportions within populations we now treat these variables as continuous rather than discrete.

○ ○ ○ ○ ○

Continuous Approximation

This means shifting to a different mathematical representation, where we no longer keep track of the individual states of each entity.

As we are focussed instead of proportions within populations we now treat these variables as continuous rather than discrete.

Continuous Approximation

This means shifting to a different mathematical representation, where we no longer keep track of the individual states of each entity.

As we are focussed instead of proportions within populations we now treat these variables as continuous rather than discrete.

○ ○ ○ ○ ○ ○ ○ ○ ○

Continuous Approximation

This means shifting to a different mathematical representation, where we no longer keep track of the individual states of each entity.

As we are focussed instead of proportions within populations we now treat these variables as continuous rather than discrete.

Continuous Approximation

This means shifting to a different mathematical representation, where we no longer keep track of the individual states of each entity.

As we are focussed instead of proportions within populations we now treat these variables as continuous rather than discrete.

○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○

Continuous Approximation

This means shifting to a different mathematical representation, where we no longer keep track of the individual states of each entity.

As we are focussed instead of proportions within populations we now treat these variables as continuous rather than discrete.

Continuous Approximation

This means shifting to a different mathematical representation, where we no longer keep track of the individual states of each entity.

As we are focussed instead of proportions within populations we now treat these variables as continuous rather than discrete.

Continuous Approximation

This means shifting to a different mathematical representation, where we no longer keep track of the individual states of each entity.

As we are focussed instead of proportions within populations we now treat these variables as continuous rather than discrete.

Continuous Approximation

This means shifting to a different mathematical representation, where we no longer keep track of the individual states of each entity.

As we are focussed instead of proportions within populations we now treat these variables as continuous rather than discrete.

Use ordinary differential equations to represent the evolution of those variables over time.

Disk model in PEPA

Disk

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

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

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 ($W + F = N$).
- Working disks fail at rate $f \times W$.

Disk model in PEPA

Disk

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

- 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

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

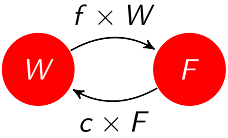
- We have W working disks and F failed $(W + F = N)$.
- Working disks fail at rate $f \times W$.
- Failures are corrected at rate $c \times F$.

Disk model in PEPA

Disk

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

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



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

Deriving a Fluid Approximation of a SPA model

The aim is to represent the CTMC implicitly (avoiding state space explosion), and to generate the set of ODEs which are the fluid limit of that CTMC.

Deriving a Fluid Approximation of a SPA model

The aim is to represent the CTMC implicitly (avoiding state space explosion), and to generate the set of ODEs which are the fluid limit of that CTMC.

The existing (CTMC) SOS semantics is not suitable for this purpose because it constructs the state space of the CTMC explicitly.

Deriving a Fluid Approximation of a SPA model

The aim is to represent the CTMC implicitly (avoiding state space explosion), and to generate the set of ODEs which are the fluid limit of that CTMC.

The existing (CTMC) SOS semantics is not suitable for this purpose because it constructs the state space of the CTMC explicitly.

SPA MODEL $\xrightarrow{\text{SOS rules}}$ LABELLED TRANSITION SYSTEM $\xrightarrow[\text{diagram}]{\text{state transition}}$ CTMC $\mathbf{Q}$

Deriving a Fluid Approximation of a SPA model

The aim is to represent the CTMC implicitly (avoiding state space explosion), and to generate the set of ODEs which are the fluid limit of that CTMC.

The existing (CTMC) SOS semantics is not suitable for this purpose because it constructs the state space of the CTMC explicitly.

We define a structured operational semantics which defines the possible transitions of an arbitrary abstract state and from this derive the ODEs.

Deriving a Fluid Approximation of a SPA model

The aim is to represent the CTMC implicitly (avoiding state space explosion), and to generate the set of ODEs which are the fluid limit of that CTMC.

The existing (CTMC) SOS semantics is not suitable for this purpose because it constructs the state space of the CTMC explicitly.

We define a structured operational semantics which defines the possible transitions of an arbitrary abstract state and from this derive the ODEs.

$$\text{SPA MODEL} \xrightarrow{\text{SOS rules}} \text{SYMBOLIC LABELLED TRANSITION SYSTEM} \xrightarrow[\text{functions}]{\text{generator}} \text{ABSTRACT CTMC } \mathbf{Q} \text{ or ODEs } F_{\mathcal{M}}(x)$$

Fluid Structured Operational Semantics

In order to get to the implicit representation of the CTMC we need to:

Fluid Structured Operational Semantics

In order to get to the implicit representation of the CTMC we need to:

1. Remove excess components to identify the counting abstraction of the process (Context Reduction)

Fluid Structured Operational Semantics

In order to get to the implicit representation of the CTMC we need to:

1. Remove excess components to identify the counting abstraction of the process (Context Reduction)
2. Collect the transitions of the reduced context as symbolic updates on the state representation (Jump Multiset)

Fluid Structured Operational Semantics

In order to get to the implicit representation of the CTMC we need to:

1. Remove excess components to identify the counting abstraction of the process (Context Reduction)
2. Collect the transitions of the reduced context as symbolic updates on the state representation (Jump Multiset)
3. Calculate the rate of the transitions in terms of an arbitrary state of the CTMC.

Fluid Structured Operational Semantics

In order to get to the implicit representation of the CTMC we need to:

1. Remove excess components to identify the counting abstraction of the process (Context Reduction)
2. Collect the transitions of the reduced context as symbolic updates on the state representation (Jump Multiset)
3. Calculate the rate of the transitions in terms of an arbitrary state of the CTMC.

Once this is done we can extract the vector field $F_{\mathcal{M}}(x)$ from the jump multiset, under the assumption that the population size tends to infinity.

M.Tribastone, S.Gilmore and J.Hillston. Scalable Differential Analysis of Process Algebra Models. IEEE TSE 2012.

Consistency results

- The vector field $\mathcal{F}(x)$ is Lipschitz continuous i.e. all the rate functions governing transitions in the process algebra satisfy local continuity conditions.

Consistency results

- The vector field $\mathcal{F}(x)$ is Lipschitz continuous i.e. all the rate functions governing transitions in the process algebra satisfy local continuity conditions.

- The generated ODEs are the fluid limit of the family of CTMCs: this family forms a sequence as the initial populations are scaled by a variable n.

Consistency results

- The vector field $\mathcal{F}(x)$ is Lipschitz continuous i.e. all the rate functions governing transitions in the process algebra satisfy local continuity conditions.
- The generated ODEs are the fluid limit of the family of CTMCs: this family forms a sequence as the initial populations are scaled by a variable n.
- We can prove this using Kurtz's theorem:
  Solutions of Ordinary Differential Equations as Limits of Pure Jump Markov Processes, T.G. Kurtz, J. Appl. Prob. (1970).

Consistency results

- The vector field $\mathcal{F}(x)$ is Lipschitz continuous i.e. all the rate functions governing transitions in the process algebra satisfy local continuity conditions.
- The generated ODEs are the fluid limit of the family of CTMCs: this family forms a sequence as the initial populations are scaled by a variable n.
- We can prove this using Kurtz's theorem: Solutions of Ordinary Differential Equations as Limits of Pure Jump Markov Processes, T.G. Kurtz, J. Appl. Prob. (1970).
- Moreover Lipschitz continuity of the vector field guarantees existence and uniqueness of the solution to the initial value problem.

M.Tribastone, S.Gilmore and J.Hillston. Scalable Differential Analysis of Process Algebra Models. IEEE TSE 2012.

Example: crowd dynamics

- Human crowds may be considered as a population of interacting entities.

Example: crowd dynamics

- Human crowds may be considered as a population of interacting entities.
- In modern buildings there may also be populations of ICT components — the informatic environment.

Example: crowd dynamics

- Human crowds may be considered as a population of interacting entities.
- In modern buildings there may also be populations of ICT components — the informatic environment.
- Interactions between these two types of agents could, for example, help people to navigate in unfamiliar locations.

Example: crowd dynamics

- Human crowds may be considered as a population of interacting entities.
- In modern buildings there may also be populations of ICT components — the informatic environment.
- Interactions between these two types of agents could, for example, help people to navigate in unfamiliar locations.
- Recently process algebra models have been used to capture the behaviour of such systems and study their dynamic behaviour.

A.Bracciali, J.Hillston, M.Massink and D.Latella. Modelling Non-linear Crowd Dynamics in Bio-PEPA. FASE 2011.

Emergency egress

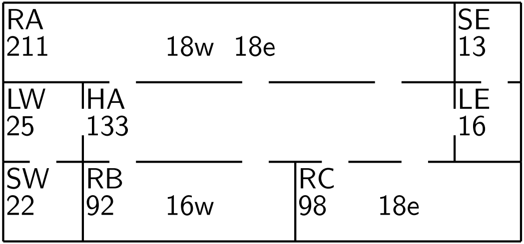
Emergency egress can be regarded as a particular case of crowd dynamics, when the location may be familiar but circumstances may alter the usual topology and make efficient movement particularly important.

Emergency egress

Emergency egress can be regarded as a particular case of crowd dynamics, when the location may be familiar but circumstances may alter the usual topology and make efficient movement particularly important.

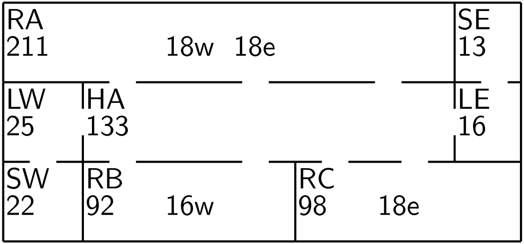
Here technology mediation may mean that information about the best routes (possibly contradicting signage) can be supplied dynamically.

Example scenario



The layout of the building is described in terms of the arrangement of the rooms, hallways, landing and stairs. Each has a capacity and may have an initial occupancy.

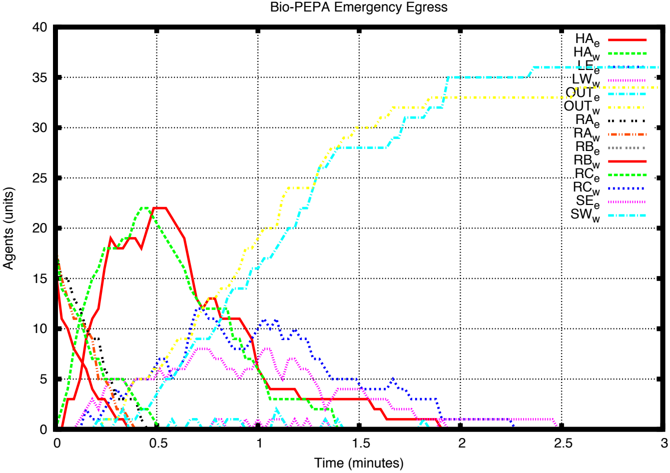
Example scenario



The layout of the building is described in terms of the arrangement of the rooms, hallways, landing and stairs. Each has a capacity and may have an initial occupancy.

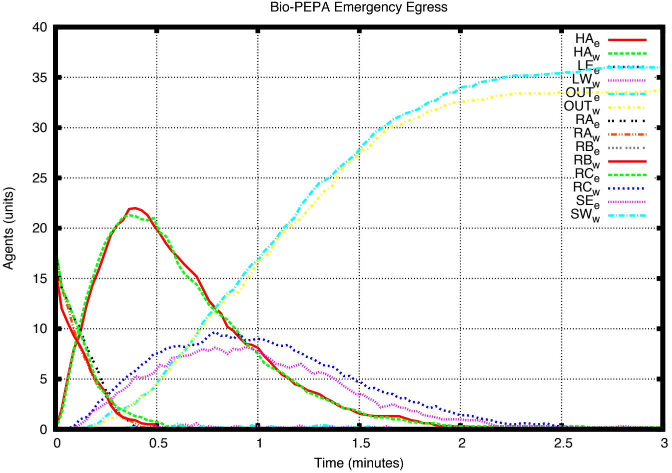
Process algebra components describe the behaviours of individuals, but also rooms and information dissemination.

Example results: room occupancy



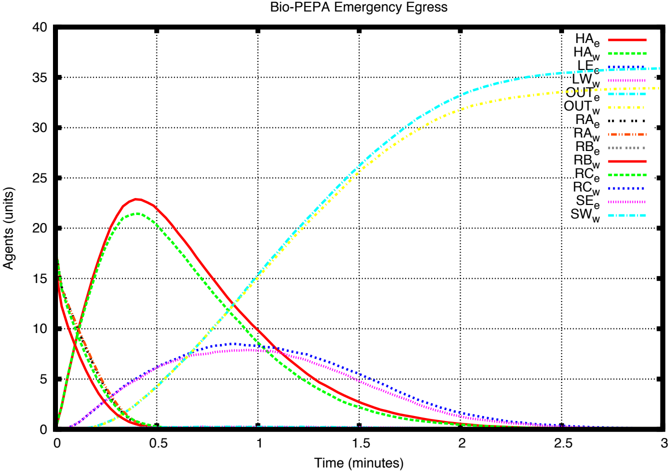
One stochastic simulation run

Example results: room occupancy



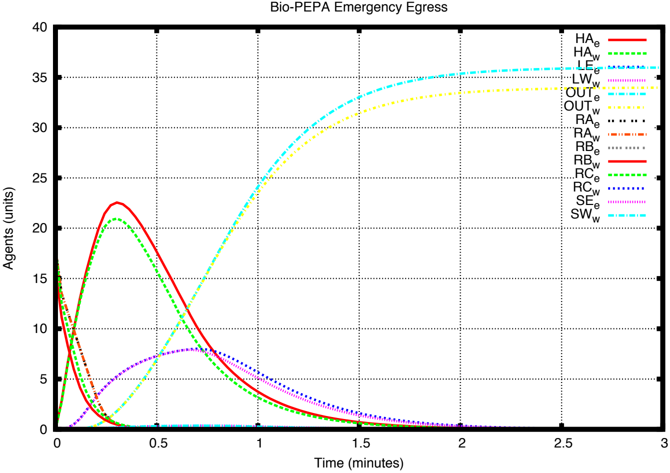
10 stochastic simulation runs

Example results: room occupancy



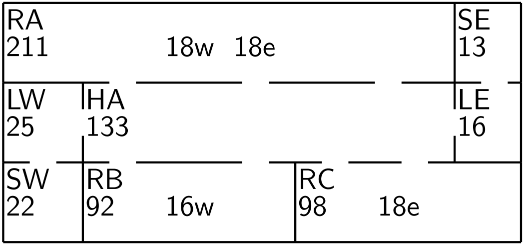
500 stochastic simulation runs

Example results: room occupancy



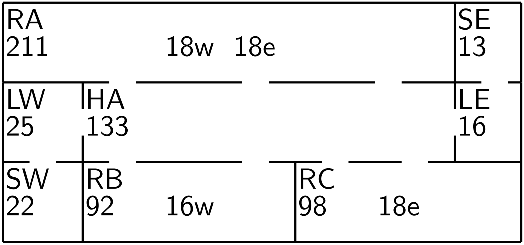
ODE numerical simulation

Adaptive behaviour: rerouting through mediation



During the course of the evacuation one of the stairwells may become blocked but the static signage will continue to send people towards the problems.

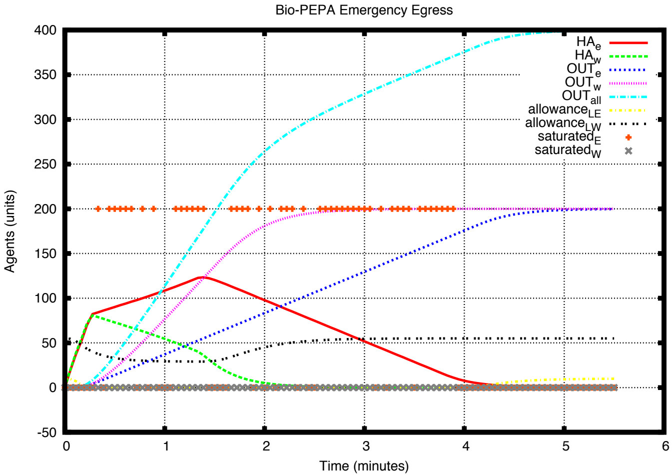
Adaptive behaviour: rerouting through mediation



During the course of the evacuation one of the stairwells may become blocked but the static signage will continue to send people towards the problems.

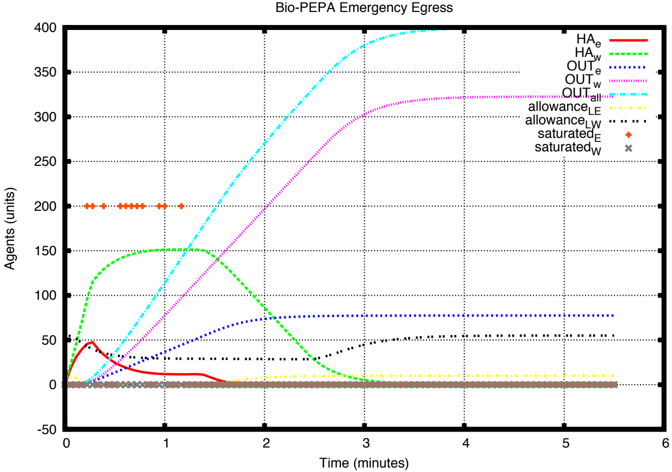
WIth technology mediation (smart signage, SMS messages) people can be rerouted dynamically.

Adaptive behaviour: rerouting through mediation



Room occupancy over time without rerouting capability

Adaptive behaviour: rerouting through mediation



Room occupancy over time with rerouting capability

Quantitative properties

The derived vector field $\mathcal{F}(x)$, gives an approximation of the expected count for each population over time.

Quantitative properties

The derived vector field $\mathcal{F}(x)$, gives an approximation of the expected count for each population over time.

This has been extended in a number of ways:

- Fluid rewards which can be safely calculated from the fluid expectation trajectories.

M.Tribastone, J.Ding, S.Gilmore and J.Hillston. Fluid Rewards for a Stochastic Process Algebra. IEEE TSE 2012.

Quantitative properties

The derived vector field $\mathcal{F}(x)$, gives an approximation of the expected count for each population over time.

This has been extended in a number of ways:

- Fluid rewards which can be safely calculated from the fluid expectation trajectories.

M.Tribastone, J.Ding, S.Gilmore and J.Hillston. Fluid Rewards for a Stochastic Process Algebra. IEEE TSE 2012.

- Vector fields have been defined to approximate higher moments.

R.A.Hayden and J.T.Bradley. A fluid analysis framework for a Markovian process algebra. TCS 2010.

Quantitative properties

The derived vector field $\mathcal{F}(x)$, gives an approximation of the expected count for each population over time.

This has been extended in a number of ways:

- Fluid rewards which can be safely calculated from the fluid expectation trajectories.

M.Tribastone, J.Ding, S.Gilmore and J.Hillston. Fluid Rewards for a Stochastic Process Algebra. IEEE TSE 2012.

- Vector fields have been defined to approximate higher moments.

R.A.Hayden and J.T.Bradley. A fluid analysis framework for a Markovian process algebra. TCS 2010.

- Fluid approximation of passage times have been defined.

R.A.Hayden, A.Stefanek and J.T.Bradley. Fluid computation of passage-time distributions in large Markov models. TCS 2012.

Fluid model checking

Since the vector field records only deterministic behaviour, LTL model checking can be used over a trace to give boolean results.

Fluid model checking

Since the vector field records only deterministic behaviour, LTL model checking can be used over a trace to give boolean results.

But for the systems we are interested in we would like some more quantified answers, in the style of stochastic model checking.

Fluid model checking

Since the vector field records only deterministic behaviour, LTL model checking can be used over a trace to give boolean results.

But for the systems we are interested in we would like some more quantified answers, in the style of stochastic model checking.

Work on this is on-going but there are initial results for:

- CSL properties of a single agent within a population.

L.Bortolussi and J.Hillston. Fluid model checking. CONCUR 2012.

- The fraction of a population that satisfies a property expressed as a one-clock deterministic timed automaton.

L.Bortolussi and R.Lanciani. Central Limit Approximation for Stochastic Model Checking. QEST 2013.

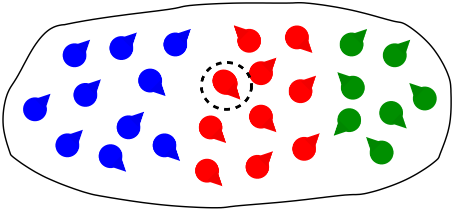
CSL model checking of a single agent

We consider properties of a single agent within a population, expressed in the Continuous Stochastic Logic (CSL), usually used for model checking CTMCs.

CSL model checking of a single agent

We consider properties of a single agent within a population, expressed in the Continuous Stochastic Logic (CSL), usually used for model checking CTMCs.

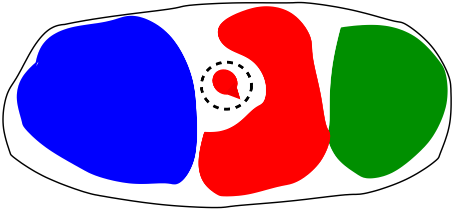
We consider an arbitrary member of the population.



CSL model checking of a single agent

We consider properties of a single agent within a population, expressed in the Continuous Stochastic Logic (CSL), usually used for model checking CTMCs.

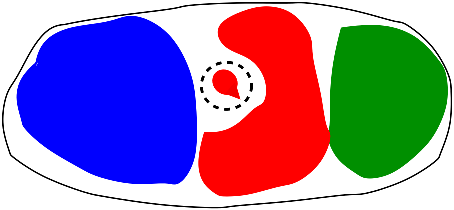
This agent is kept discrete, making transitions between its discrete states, but all other agents are treated as a mean-field influencing the behaviour of this agent.



CSL model checking of a single agent

We consider properties of a single agent within a population, expressed in the Continuous Stochastic Logic (CSL), usually used for model checking CTMCs.

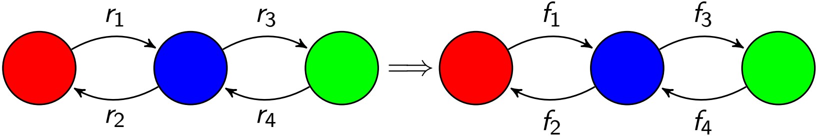
Essentially we keep a detailed discrete-event representation of the one agent and make a fluid approximation of the rest of the population.

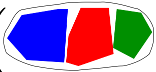


Inhomogeneous CTMC

The transition rates within the discrete-event representation will depend on the rest of the population.

i.e. it will depend on the vector field capturing the behaviour of the residual population.

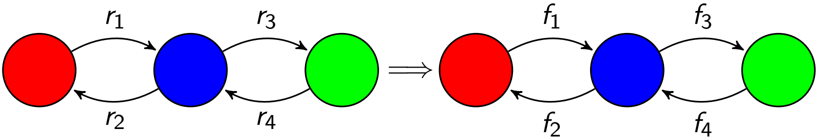


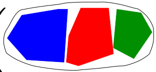
where $f_i = f\Bigg($  $\Bigg)$

Inhomogeneous CTMC

The transition rates within the discrete-event representation will depend on the rest of the population.

i.e. it will depend on the vector field capturing the behaviour of the residual population.



where $f_i = f\Big($  $\Big)$

It is an inhomogeneous continuous time Markov chain.

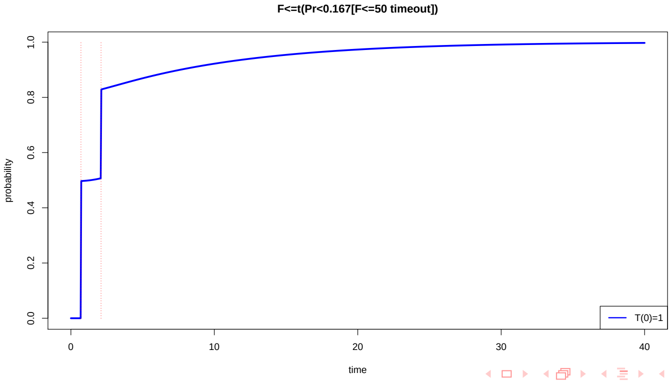
Model checking the ICTMC

Care is needed to model check the ICTMC, which proceeds by explicitly calculating the reachability probability probabilities for states of interest (analogously to CSL model checking on CTMCs).

Model checking the ICTMC

Care is needed to model check the ICTMC, which proceeds by explicitly calculating the reachability probability probabilities for states of interest (analogously to CSL model checking on CTMCs).

The inhomogeneous time within the model means that truth values may change with respect to time.



Challenges for modelling CAS

The work so far demonstrates provides a solid basic framework for modelling systems with collective behaviour but there remain a number of challenges:

- Spatial aspects
- Richer forms of interaction and adaptation
- Extending model checking capabilities

Modelling space

Whilst fluid approximation of stochastic process algebra models can be used to model collective systems there is an implicit assumption that all components are co-located.

Modelling space

Whilst fluid approximation of stochastic process algebra models can be used to model collective systems there is an implicit assumption that all components are co-located.

We can impose the effects of space by encoding it into the behaviour of the actions of components and distinguishing the same component in different location as distinct types, but this is modelling space implicitly.

Modelling space

Whilst fluid approximation of stochastic process algebra models can be used to model collective systems there is an implicit assumption that all components are co-located.

We can impose the effects of space by encoding it into the behaviour of the actions of components and distinguishing the same component in different location as distinct types, but this is modelling space implicitly.

Given the important role that location and movement play within many CAS, e.g. smart cities, it would be preferable to model space explicitly.

Modelling space

Whilst fluid approximation of stochastic process algebra models can be used to model collective systems there is an implicit assumption that all components are co-located.

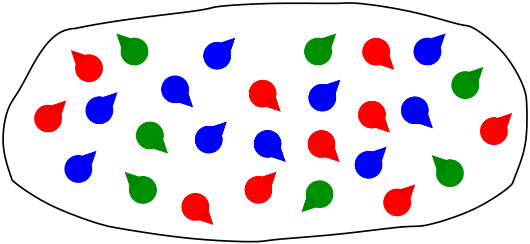
We can impose the effects of space by encoding it into the behaviour of the actions of components and distinguishing the same component in different location as distinct types, but this is modelling space implicitly.

Given the important role that location and movement play within many CAS, e.g. smart cities, it would be preferable to model space explicitly.

This poses significant challenges both of model expression and model solution.

Fluid approximation and space

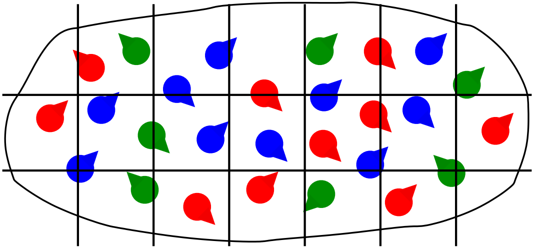
There is a danger that as we distinguish subpopulations by their location, we no longer have a large enough population to justify the fluid approximation.



Initial work is exploring the use of time scale decompositions, partial differential equations and diffusion models but much more work is needed.

Fluid approximation and space

There is a danger that as we distinguish subpopulations by their location, we no longer have a large enough population to justify the fluid approximation.



Initial work is exploring the use of time scale decompositions, partial differential equations and diffusion models but much more work is needed.

Richer forms of interaction and adaptation

If we consider real collective adaptive systems, especially those with emergent behaviour, they embody rich forms of interaction, often based on asynchronous communication.

Richer forms of interaction and adaptation

If we consider real collective adaptive systems, especially those with emergent behaviour, they embody rich forms of interaction, often based on asynchronous communication.

For example, pheromone trails left by social insects.

Richer forms of interaction and adaptation

If we consider real collective adaptive systems, especially those with emergent behaviour, they embody rich forms of interaction, often based on asynchronous communication.

For example, pheromone trails left by social insects.

Languages like SCEL offer these richer communication patterns, with components which include a knowledge store which can be manipulated by other components and attribute-based communication.

R.De Nicola, G.Ferrari, M.Loreti, R.Pugliese. A Language-Based Approach to Autonomic Computing. FMCO 2011.

Richer forms of interaction and adaptation

If we consider real collective adaptive systems, especially those with emergent behaviour, they embody rich forms of interaction, often based on asynchronous communication.

For example, pheromone trails left by social insects.

Languages like SCEL offer these richer communication patterns, with components which include a knowledge store which can be manipulated by other components and attribute-based communication.

R.De Nicola, G.Ferrari, M.Loreti, R.Pugliese. A Language-Based Approach to Autonomic Computing. FMCO 2011.

Developing scalable analysis techniques, such a fluid approximation, for such languages remains an open problem.

Conclusions

- Collective Adaptive Systems are an interesting and challenging class of systems to design and construct.

Conclusions

- Collective Adaptive Systems are an interesting and challenging class of systems to design and construct.
- Their role within infrastructure, such as within smart cities, make it essential that quantitive aspects of behaviour is taken into consideration, as well as functional correctness.

Conclusions

- Collective Adaptive Systems are an interesting and challenging class of systems to design and construct.
- Their role within infrastructure, such as within smart cities, make it essential that quantitive aspects of behaviour is taken into consideration, as well as functional correctness.
- Fluid approximation based analysis offers hope for scalable quantitative analysis techniques, but there remain many interesting and challenging problems to be solved.

Thank you!

Thanks to the other members of the QUANTICOL project

www.quanticol.eu