Challenges for Quantitative Analysis of Collective Adaptive Systems

Jane Hillston
LFCS, University of Edinburgh

15th December 2014

Outline

1. Introduction
   - Collective Adaptive Systems
   - Quantitative Analysis
2. Quantitative Analysis of CAS
   - Mathematical analysis: fluid approximation
   - Deriving properties: fluid model checking
3. Challenges and future prospects

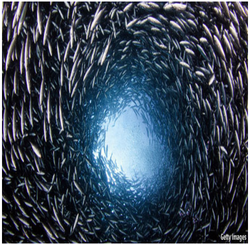
Collective Systems

We are surrounded by examples of collective systems:

Collective Systems

We are surrounded by examples of collective systems:

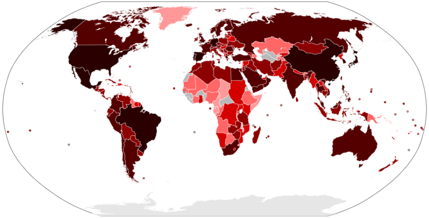
in the natural world ....

Collective Systems

We are surrounded by examples of collective systems:

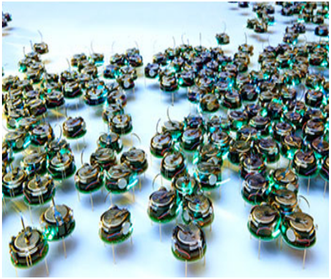
.... and in the man-made world



Collective Systems

We are surrounded by examples of collective systems:

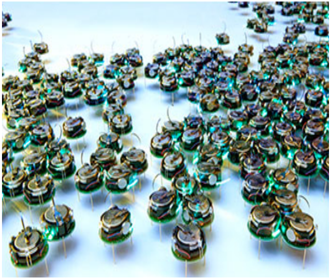
.... and in the man-made world



Collective Systems

We are surrounded by examples of collective systems:

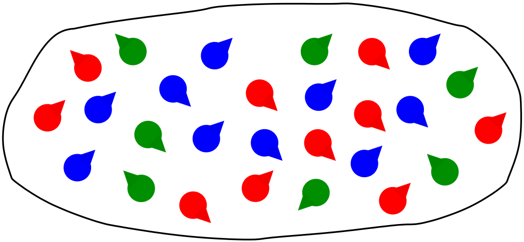
.... and in the man-made world



Most of these systems are also adaptive to their environment.

Collective Adaptive Systems

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

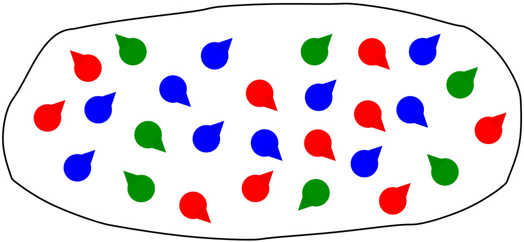


Each entity may have its own properties, objectives and actions.

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

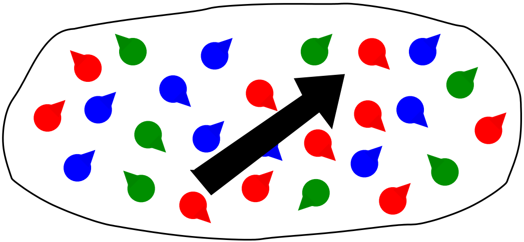
Collective Adaptive Systems

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



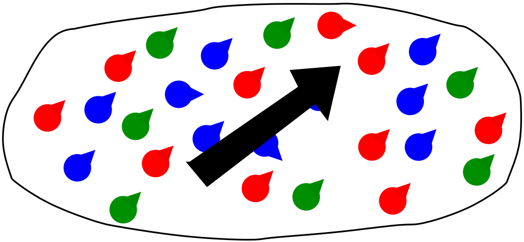
Collective Adaptive Systems

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



Collective Adaptive Systems

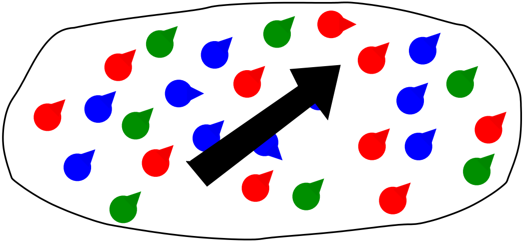
The behaviour of the system is thus dependent on the behaviour of the individual entities.



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

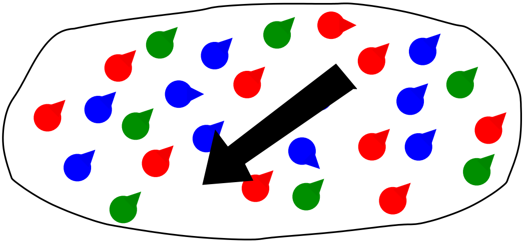
Collective Adaptive Systems

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



Collective Adaptive Systems

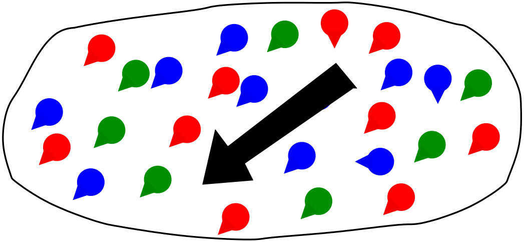
Such systems are often embedded in our environment and need to operate without centralised control or direction.



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

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

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.

The Informatic Environment

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.

Their transparency to the end-user means that it is paramount that the designers of such systems seek to ensure that their behaviour in terms of both qualitative and quantitative properties will be as anticipated.

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.

Dependability modelling aims to predict the availability and reliability of systems in use from the users' perspective.

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.

Dependability modelling aims to predict the availability and reliability of systems in use from the users' perspective.

Markovian-based discrete event models have been applied to computer and communication systems for these purposes 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.

Dependability modelling aims to predict the availability and reliability of systems in use from the users' perspective.

Markovian-based discrete event models have been applied to computer and communication systems for these purposes since the early 20th century.

Unfortunately like all discrete state modelling techniques, Markovian models are prone to the problem of state space explosion.

Does timeliness matter...?

There is sometimes a perception in software development that performance does not matter much, or that it is easily fixed later by buying a faster machine.

Does timeliness matter...?

There is sometimes a perception in software development that performance does not matter much, or that it is easily fixed later by buying a faster machine.

On the contrary — studies have shown that response time is a key feature in user satisfaction and trust in systems.

Does timeliness matter...?

There is sometimes a perception in software development that performance does not matter much, or that it is easily fixed later by buying a faster machine.

On the contrary — studies have shown that response time is a key feature in user satisfaction and trust in systems.

In a 2009 study by Amazon they artificially delayed page loading times in increments of 100 milliseconds. Even such very small delays were observed to result in substantial and costly drops in revenue.

Gary Linden, Amazon, quoted on `http://perspectives.mvdirona.com/2009/10/31/TheCostOfLatency`

Does timeliness matter...?

There is sometimes a perception in software development that performance does not matter much, or that it is easily fixed later by buying a faster machine.

On the contrary — studies have shown that response time is a key feature in user satisfaction and trust in systems.

In a 2009 study by Amazon they artificially delayed page loading times in increments of 100 milliseconds. Even such very small delays were observed to result in substantial and costly drops in revenue.

Gary Linden, Amazon, quoted on `http://perspectives.mvdirona.com/2009/10/31/TheCostOfLatency`

AOL, Bing and Google report similar findings.

Quantitative Analysis in a Smart City Scenario

Capacity planning

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

Quantitative Analysis in a Smart City Scenario

System Configuration

- What capacity do I need at bike stations to minimise the movement of bikes by truck and/or the dissatisfaction of users?

Quantitative Analysis in a Smart City Scenario

System Tuning

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

Quantitative Analysis of CAS

To support the development of CAS which meet quantitative objectives we need an innovative formal design framework:

- an unambiguous way of describing the behaviour of the systems we are interested in;
- a logic or requirements language which allows us to express the behaviours we wish our designed system to have;
- automatic ways to check the description against the requirements, captured in software tools;
- rules that help us construct systems to meet commonly occurring requirements

Quantitative Analysis of CAS

To support the development of CAS which meet quantitative objectives we need an innovative formal design framework:

- an unambiguous way of describing the behaviour of the systems we are interested in; a process algebra-based language
- a logic or requirements language which allows us to express the behaviours we wish our designed system to have;
- automatic ways to check the description against the requirements, captured in software tools;
- rules that help us construct systems to meet commonly occurring requirements

Quantitative Analysis of CAS

To support the development of CAS which meet quantitative objectives we need an innovative formal design framework:

- an unambiguous way of describing the behaviour of the systems we are interested in; a process algebra-based language
- a logic or requirements language which allows us to express the behaviours we wish our designed system to have; a logic
- automatic ways to check the description against the requirements, captured in software tools;
- rules that help us construct systems to meet commonly occurring requirements

Quantitative Analysis of CAS

To support the development of CAS which meet quantitative objectives we need an innovative formal design framework:

- an unambiguous way of describing the behaviour of the systems we are interested in; a process algebra-based language
- a logic or requirements language which allows us to express the behaviours we wish our designed system to have; a logic
- automatic ways to check the description against the requirements, captured in software tools; model checking
- rules that help us construct systems to meet commonly occurring requirements

Quantitative Analysis of CAS

To support the development of CAS which meet quantitative objectives we need an innovative formal design framework:

- an unambiguous way of describing the behaviour of the systems we are interested in; a process algebra-based language
- a logic or requirements language which allows us to express the behaviours we wish our designed system to have; a logic
- automatic ways to check the description against the requirements, captured in software tools; model checking
- rules that help us construct systems to meet commonly occurring requirements a design methodology.

Quantitative Analysis of CAS

To support the development of CAS which meet quantitative objectives we need an innovative formal design framework:

- an unambiguous way of describing the behaviour of the systems we are interested in; a process algebra-based language
- a logic or requirements language which allows us to express the behaviours we wish our designed system to have; a logic
- automatic ways to check the description against the requirements, captured in software tools; model checking
- rules that help us construct systems to meet commonly occurring requirements a design methodology.

Such a framework is being developed in the EU-funded QUANTICOL project.

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.

For this purpose we use a Stochastic/Markovian Process Algebra.

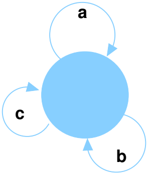
Process Algebra

- Models consist of agents which engage in actions.

$$\alpha.P$$

action type or name

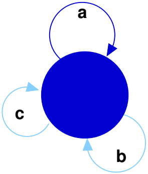
agent/ component



Process Algebra

- Models consist of agents which engage in actions.

$$\underbrace{\alpha}_{\text{action type or name}}.\underbrace{P}_{\text{agent/ component}}$$



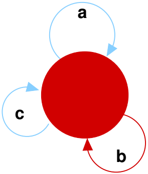
Process Algebra

- Models consist of agents which engage in actions.

$$\alpha.P$$

action type
or name

agent/
component



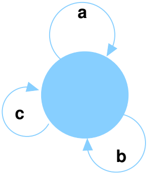
Process Algebra

- Models consist of agents which engage in actions.

$$\alpha.P$$

action type or name

agent/ component



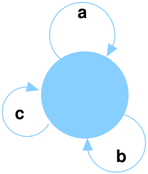
Process Algebra

- Models consist of agents which engage in actions.

$$\alpha.P$$

action type or name

agent/ component



- The structured operational (interleaving) semantics of the language is used to generate a labelled transition system.

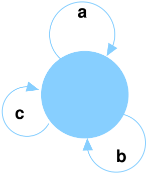
Process Algebra

- Models consist of agents which engage in actions.

$$\alpha.P$$

action type or name

agent/ component



- The structured operational (interleaving) semantics of the language is used to generate a labelled transition system.

Process algebra model

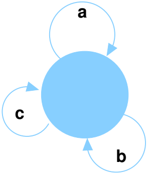
Process Algebra

- Models consist of agents which engage in actions.

$$\alpha.P$$

action type or name

agent/ component



- The structured operational (interleaving) semantics of the language is used to generate a labelled transition system.

Process algebra model $\xrightarrow{\text{SOS rules}}$

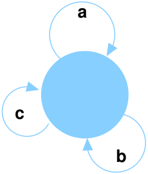
Process Algebra

- Models consist of agents which engage in actions.

$$\alpha.P$$

action type or name

agent/ component

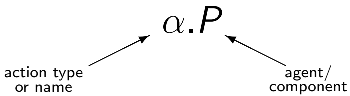


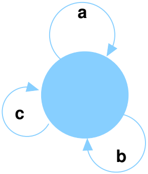
- The structured operational (interleaving) semantics of the language is used to generate a labelled transition system.

Process algebra model $\xrightarrow{\text{SOS rules}}$ Labelled transition system

Process Algebra

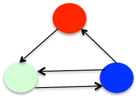
- Models consist of agents which engage in actions.





- The structured operational (interleaving) semantics of the language is used to generate a labelled transition system.

Process algebra model $\xrightarrow{\text{SOS rules}}$ Labelled transition system



Stochastic process algebras

Process algebras where models are decorated with quantitative information used to generate a stochastic process are stochastic process algebras (SPA).

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 Continuous Time Markov Chain (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 Continuous Time Markov Chain (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 Continuous Time Markov Chain (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 Continuous Time Markov Chain (CTMC) for performance modelling.

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

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 Continuous Time Markov Chain (CTMC) for performance modelling.

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

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 Continuous Time Markov Chain (CTMC) for performance modelling.

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

Integrated analysis

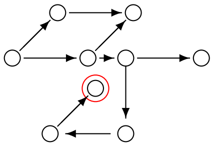
Qualitative verification can now be complemented by quantitative verification.

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?

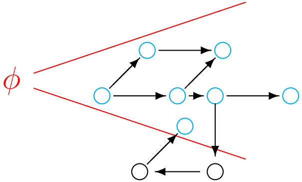


Integrated analysis

Qualitative verification can now be complemented by quantitative verification.

Model checking

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

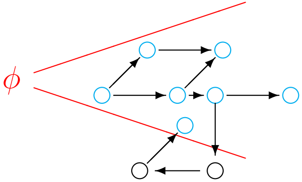


Integrated analysis

Qualitative verification can now be complemented by quantitative verification.

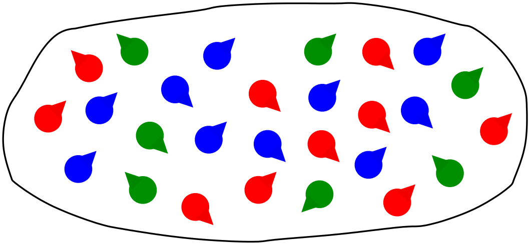
Model checking

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



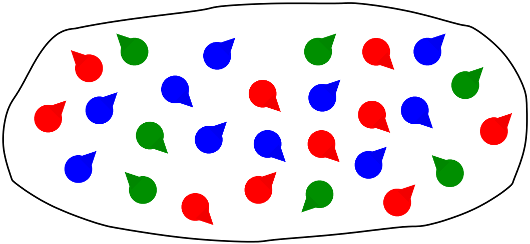
Modelling collective behaviour

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.

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

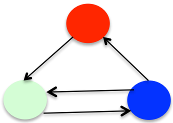
Process algebra are well-suited for constructing models of CAS:

- 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.

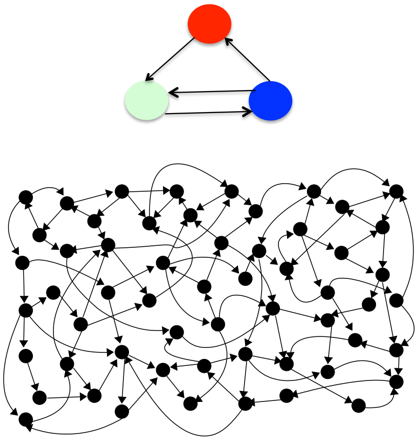
Solving discrete state models

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.

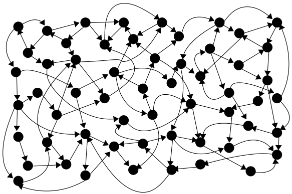


Solving discrete state models

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.

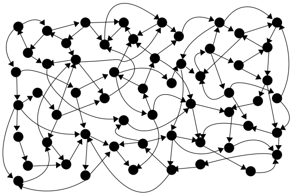


Solving discrete state models



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.

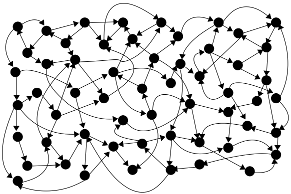
Solving discrete state models



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



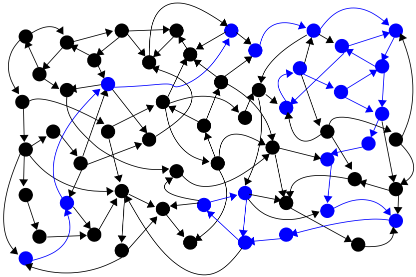
When the size of the state space is not too large they are amenable to numerical solution (linear algebra) to determine a steady state or transient probability distribution.

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

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

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.

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.

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.

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.

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.

This constitutes a continuous, fluid or mean field approximation.

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

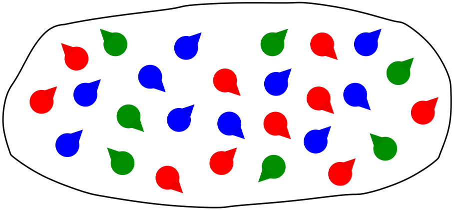
Collective systems are constructed from many instances of a set of components.

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.

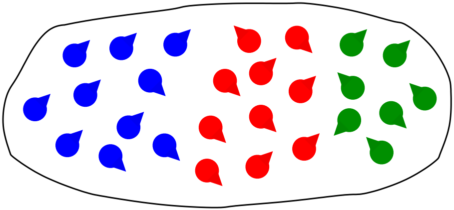
If we cease to distinguish between instances of components we can form an aggregation which can reduce the state space.



Identity and Individuality

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

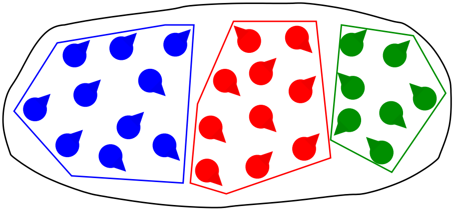
If we cease to distinguish between instances of components we can form an aggregation which can reduce the state space.



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.

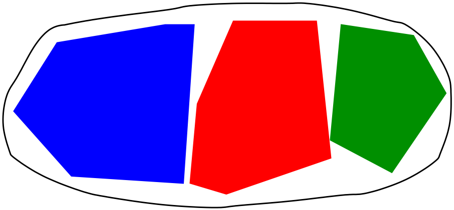


We may choose to disregard the identity of components.

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.

Deriving a Fluid Approximation of a SPA model

The approach is based on a semantics which represents the CTMC implicitly (avoiding state space explosion), and which generates the set of ODEs which are the fluid limit of that CTMC.

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

Deriving a Fluid Approximation of a SPA model

The approach is based on a semantics which represents the CTMC implicitly (avoiding state space explosion), and which generates 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.

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

Deriving a Fluid Approximation of a SPA model

The approach is based on a semantics which represents the CTMC implicitly (avoiding state space explosion), and which generates 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 **Q**

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

Deriving a Fluid Approximation of a SPA model

The approach is based on a semantics which represents the CTMC implicitly (avoiding state space explosion), and which generates 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.

The scalable semantics defines the possible transitions of an arbitrary abstract state in terms of generator functions which can then be used to derive the ODEs.

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

Deriving a Fluid Approximation of a SPA model

The approach is based on a semantics which represents the CTMC implicitly (avoiding state space explosion), and which generates 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.

The scalable semantics defines the possible transitions of an arbitrary abstract state in terms of generator functions which can then be used to derive the ODEs.

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

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.

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.

Deriving properties: fluid model checking

Just as with Markovian-based quantitative analysis, the direct study of the behaviour of the model in terms of the exhibited behaviour, whilst valuable, is sometimes not sufficient to assess the system properties which we wish to ensure.

Deriving properties: fluid model checking

Just as with Markovian-based quantitative analysis, the direct study of the behaviour of the model in terms of the exhibited behaviour, whilst valuable, is sometimes not sufficient to assess the system properties which we wish to ensure.

Thus we seek the analogy of the stochastic model checking as supported by tools such as PRISM or MRMC, but without the dependence on an explicit discrete state space of the whole system.

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.

We would like more quantified answers, as in stochastic model checking — work on this is on-going but there are initial results:

- 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.

- Approximating the probability that a given set of states in the population state space will be reached.

  L.Bortolussi and R.Lanciani.
  Stochastic Approximation of Global Reachability Probabilities of Markov Population Models. EPEW 2014.

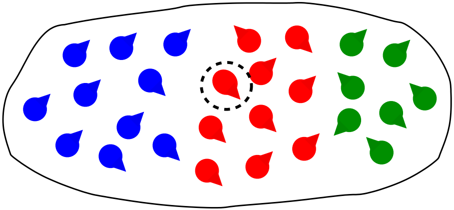
CSL model checking of a single agent

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.

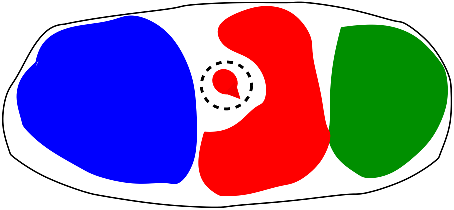
We consider an arbitrary member of the population.



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.

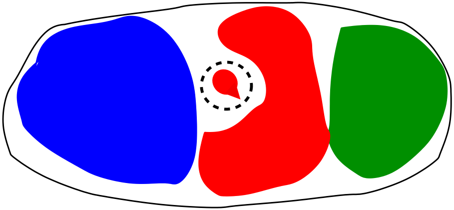
This agent is kept discrete, making transitions between its discrete states, but all other agents are treated as a mean-field influencing the behaviour of this agent.



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.

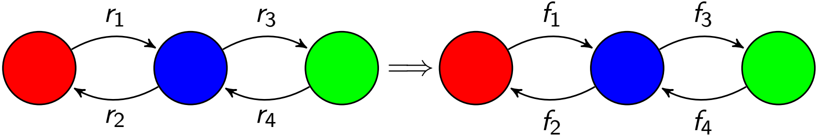
Essentially we keep a detailed discrete-event representation of the one agent and make a fluid approximation of the rest of the population.

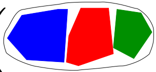


Inhomogeneous CTMC

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.

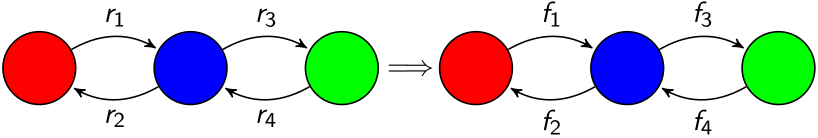


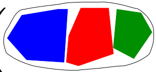
where $f_i = f\Big($  $\Big)$

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.

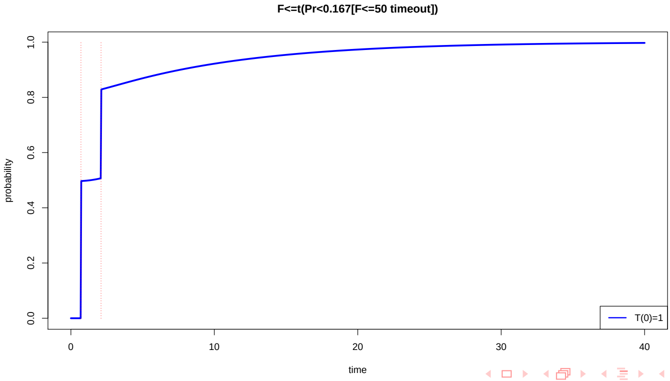
Model checking the ICTMC

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 has demonstrated the potential power of the fluid approximation approach and its validity as an approach to the quantitative analysis of (some) CAS.

Challenges for modelling CAS

The work so far has demonstrated the potential power of the fluid approximation approach and its validity as an approach to the quantitative analysis of (some) CAS.

Whilst this provides a solid basic framework for modelling systems with collective behaviour 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 the system is well-mixed, i.e. 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 the system is well-mixed, i.e. 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 the system is well-mixed, i.e. 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 the system is well-mixed, i.e. all components are co-located.

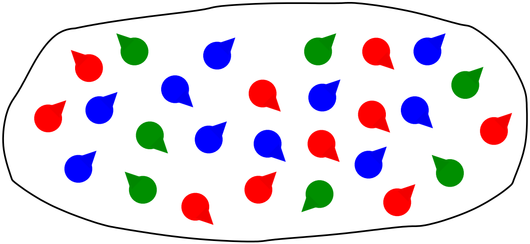
We can impose the effects of space by encoding it into the behaviour of the actions of components and distinguishing the same component in different location as distinct types, but this is modelling space implicitly.

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.

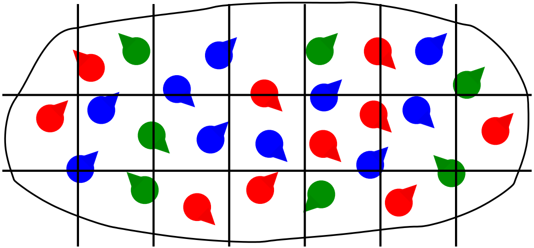
Fluid approximation and space

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.



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.



Richer forms of interaction

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

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

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 (and CASPA) 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

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 (and CASPA) 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 is challenging as again the individuals within the population are being differentiated by their attributes.

Adaptation

Current stochastic process algebras model adaptation in the sense of allowing functional rates which depend on the current state of the system.

Adaptation

Current stochastic process algebras model adaptation in the sense of allowing functional rates which depend on the current state of the system.

In this case the possible states of the system remain the same, but the dynamics may change how likely some of those states are.

Adaptation

Current stochastic process algebras model adaptation in the sense of allowing functional rates which depend on the current state of the system.

In this case the possible states of the system remain the same, but the dynamics may change how likely some of those states are.

- Is this enough?
- Do we need more radical adaptation and if so, how do we capture it during model construction and evaluate it during model analysis?

Extending model checking capabilities

Model checking offers the possibility of detailed interrogation of models via sophisticated properties rigorously expressed in logic.

This can be used to used to check to satisfaction of regulatory requirements are as well as user requirements.

Extending model checking capabilities

Model checking offers the possibility of detailed interrogation of models via sophisticated properties rigorously expressed in logic.

This can be used to used to check to satisfaction of regulatory requirements are as well as user requirements.

But the scale of CAS remains a challenge, and the incorporation of space into models necessitates the definition of spatial-temporal logics for the expression of properties.

Thank you!

Thank you!

quanticol

www.quanticol.eu