Process algebras for quantitative analysis: Lecture 1 — Introduction

Jane Hillston

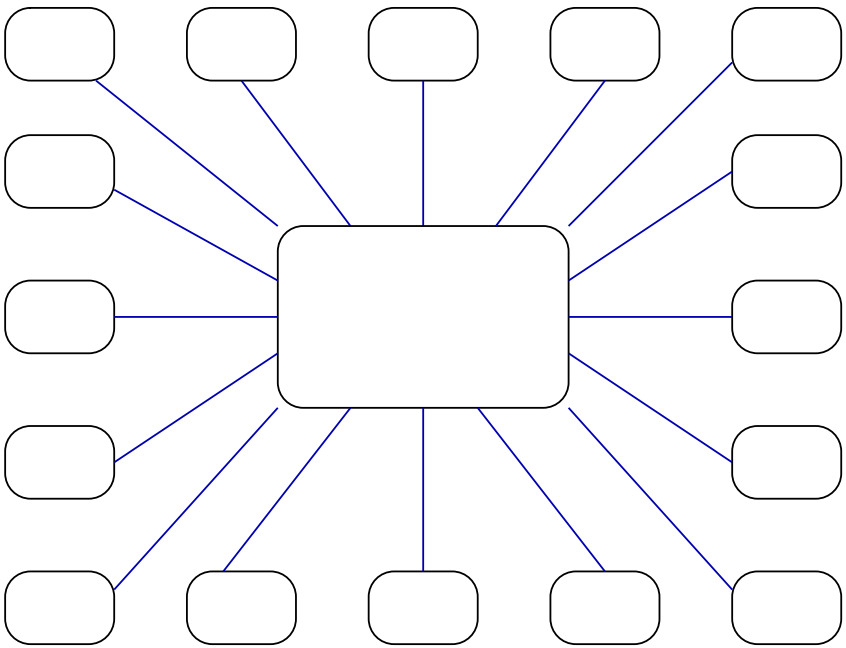
School of Informatics
The University of Edinburgh
Scotland

8th February 2010

Outline

Outline

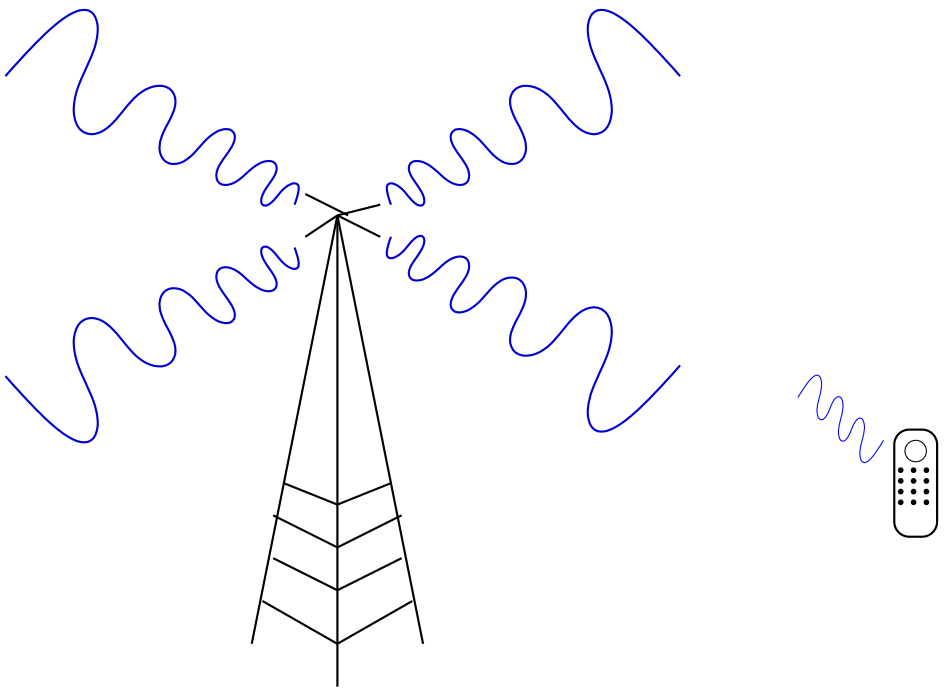
Performance Modelling: Motivation



Capacity planning

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

Performance Modelling: Motivation

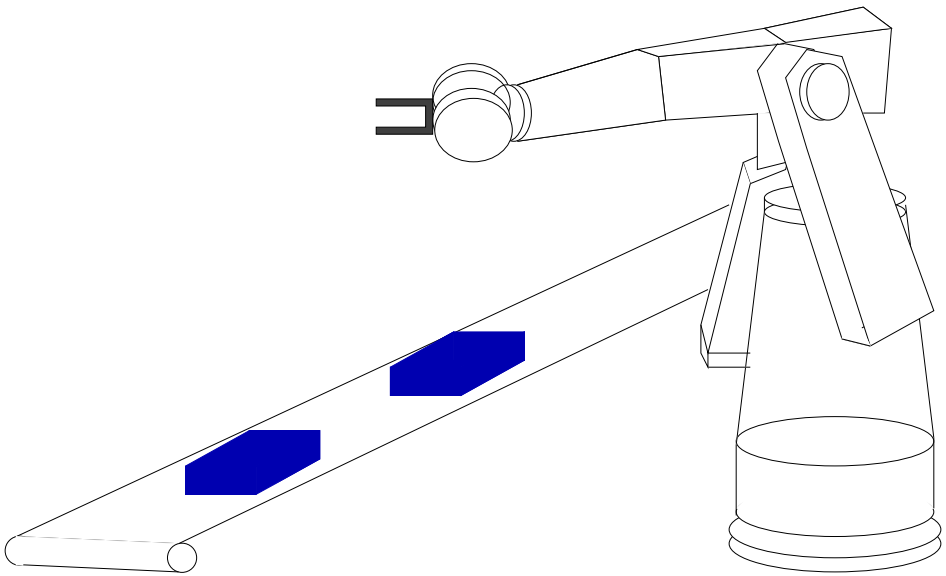


Mobile Telephone Antenna

System Configuration

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

Performance Modelling: Motivation



System Tuning

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

Modelling computer systems: the challenges

- Physical distance
  - Network latency

Modelling computer systems: the challenges

- Physical distance
  - Network latency
- Partial failures
  - Server may be down
  - Routers may be down

Modelling computer systems: the challenges

- Physical distance
  - Network latency
- Partial failures
  - Server may be down
  - Routers may be down
- Scale
  - Workload characterisation

Modelling computer systems: the challenges

- Physical distance
  - Network latency
- Partial failures
  - Server may be down
  - Routers may be down
- Scale
  - Workload characterisation
- Resource sharing
  - Network contention
  - CPU load

Modelling computer systems: the challenges

- Physical distance — need to represent time
  - Network latency
- Partial failures
  - Server may be down
  - Routers may be down
- Scale
  - Workload characterisation
- Resource sharing
  - Network contention
  - CPU load

Modelling computer systems: the challenges

- Physical distance — need to represent time
  - Network latency
- Partial failures — randomness and probability
  - Server may be down
  - Routers may be down
- Scale
  - Workload characterisation
- Resource sharing
  - Network contention
  - CPU load

Modelling computer systems: the challenges

- Physical distance — need to represent time
  - Network latency
- Partial failures — randomness and probability
  - Server may be down
  - Routers may be down
- Scale — need to quantify population sizes
  - Workload characterisation
- Resource sharing
  - Network contention
  - CPU load

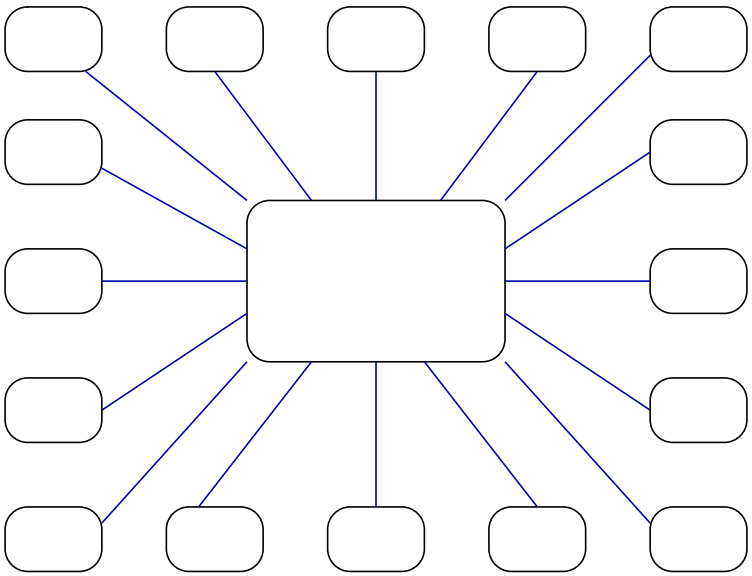
Modelling computer systems: the challenges

- Physical distance — need to represent time
  - Network latency
- Partial failures — randomness and probability
  - Server may be down
  - Routers may be down
- Scale — need to quantify population sizes
  - Workload characterisation
- Resource sharing — need to express percentages
  - Network contention
  - CPU load

Modelling computer systems: the challenges

Time	What representation of time will we use?
Randomness	What kind of random number distributions will we use?
Probability	How can we have probabilities in the model without uncertainty in the results?
Scale	How can we escape the state-space explosion problem?
Percentages	What can it mean to have a fraction of a process?

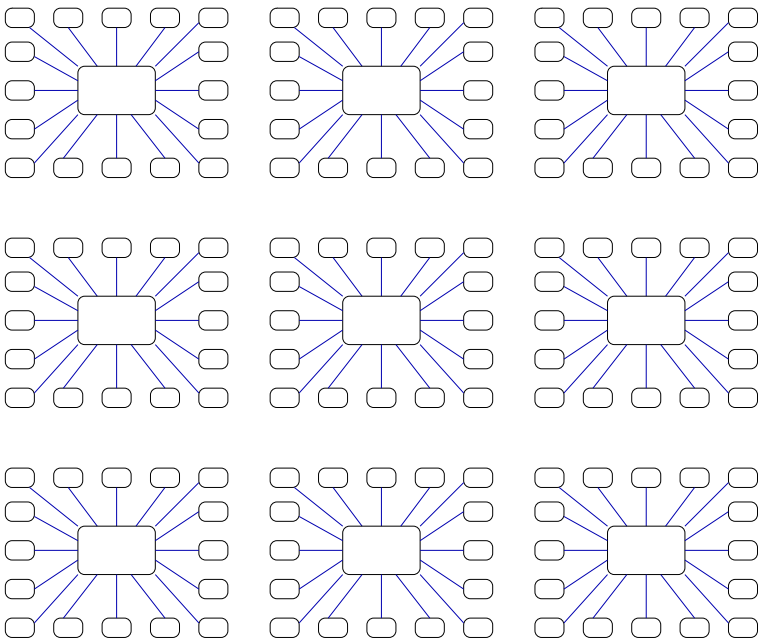
Quantitative Modelling: Motivation



Quality of Service issues

- Can the server maintain reasonable response times?

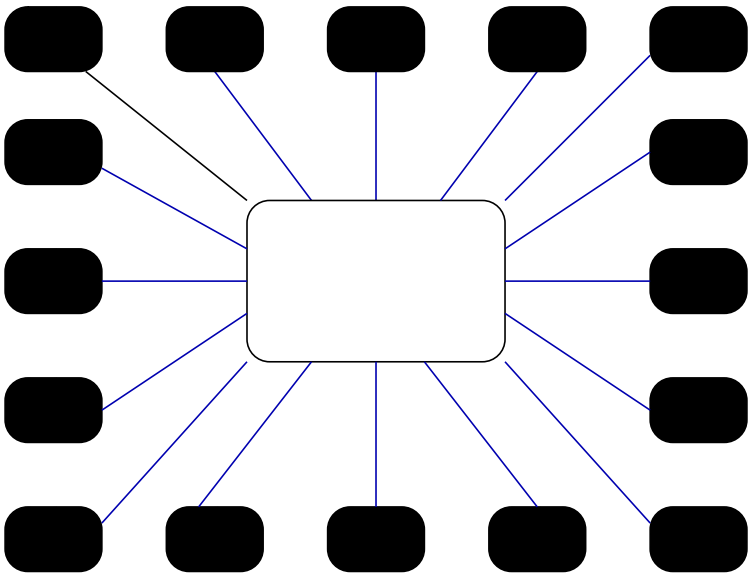
Quantitative Modelling: Motivation



Scalability issues

- How many times do we have to replicate this service to support all of the subscribers?

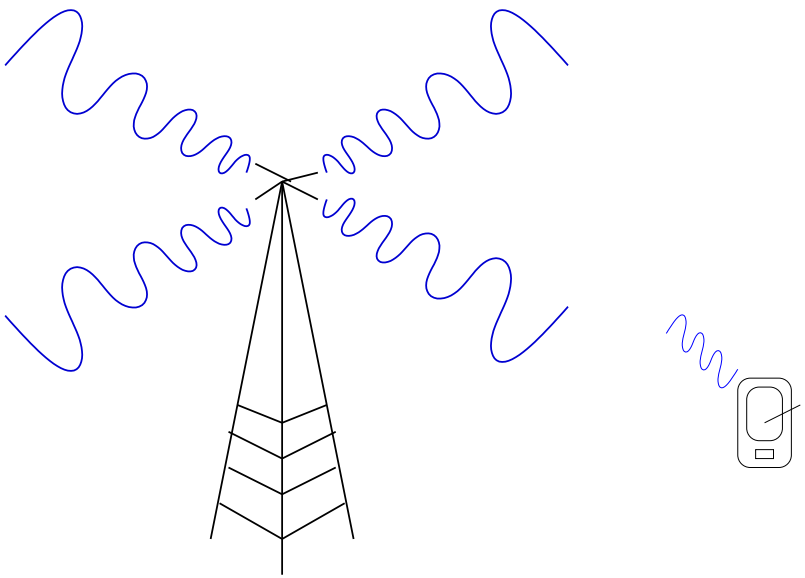
Quantitative Modelling: Motivation



Scalability issues

- Will the server withstand a distributed denial of service attack?

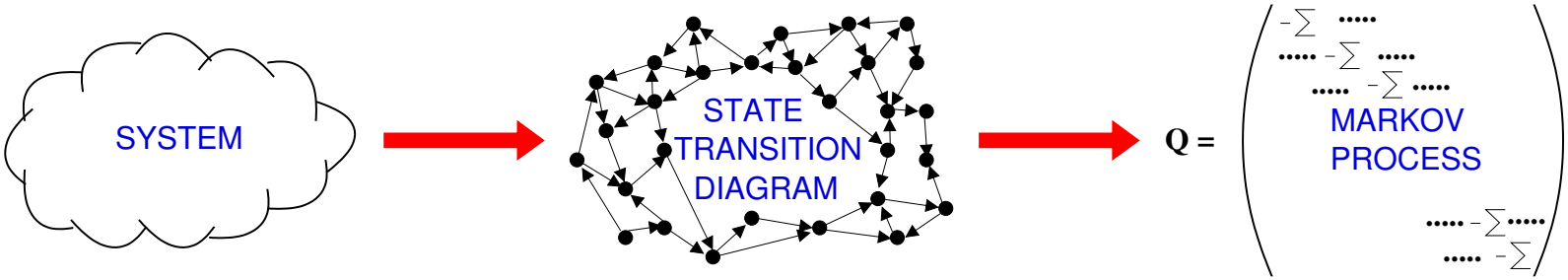
Quantitative Modelling: Motivation



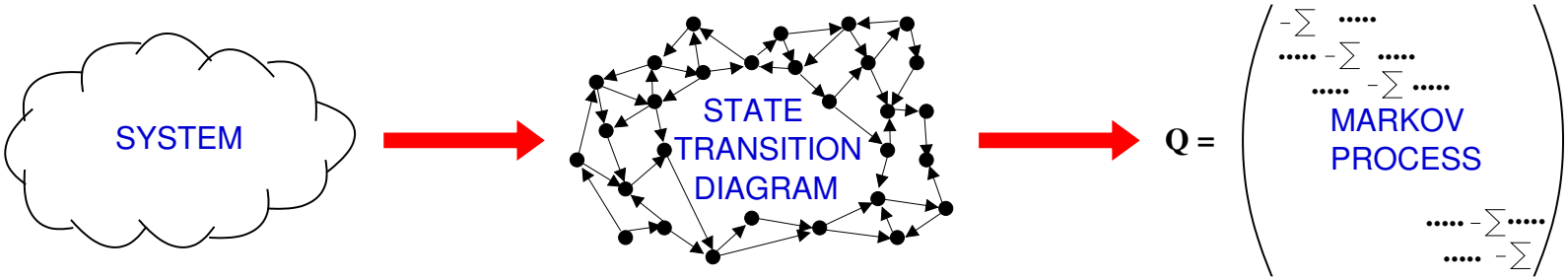
Service-level agreements

- What percentage of downloads do complete within the time we advertised?

Performance Modelling using CTMC



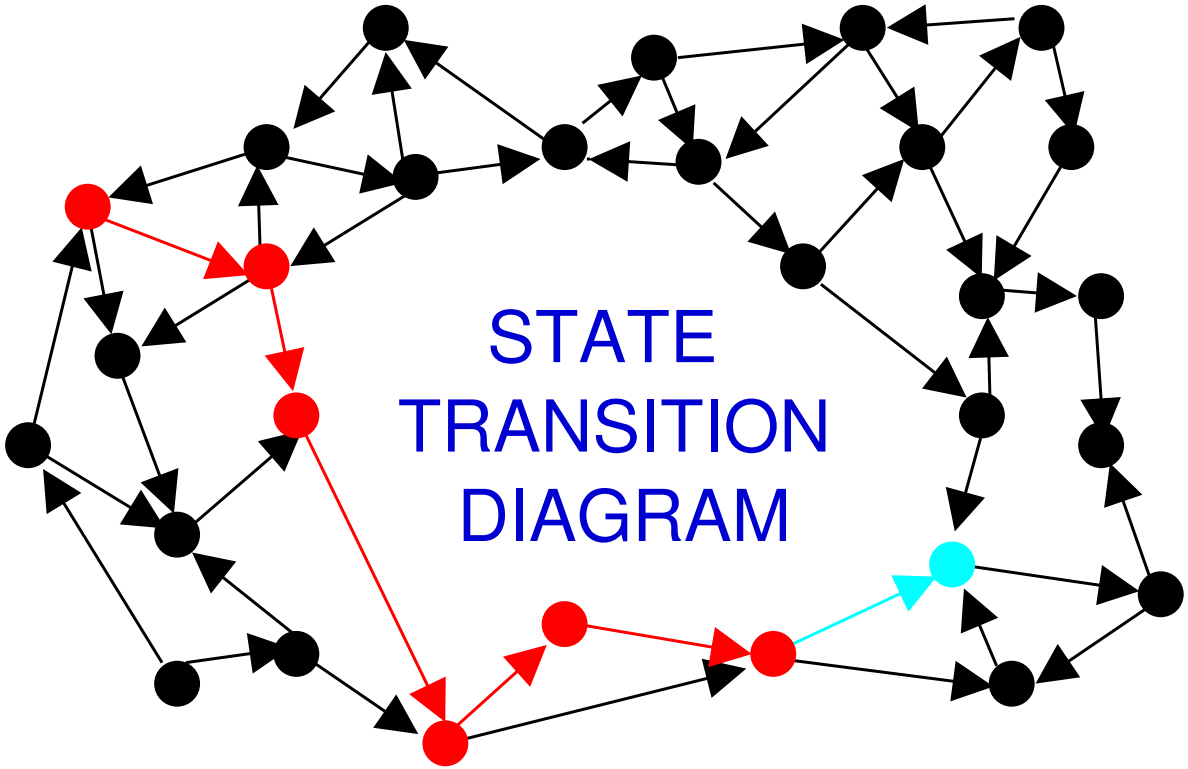
Performance Modelling using CTMC



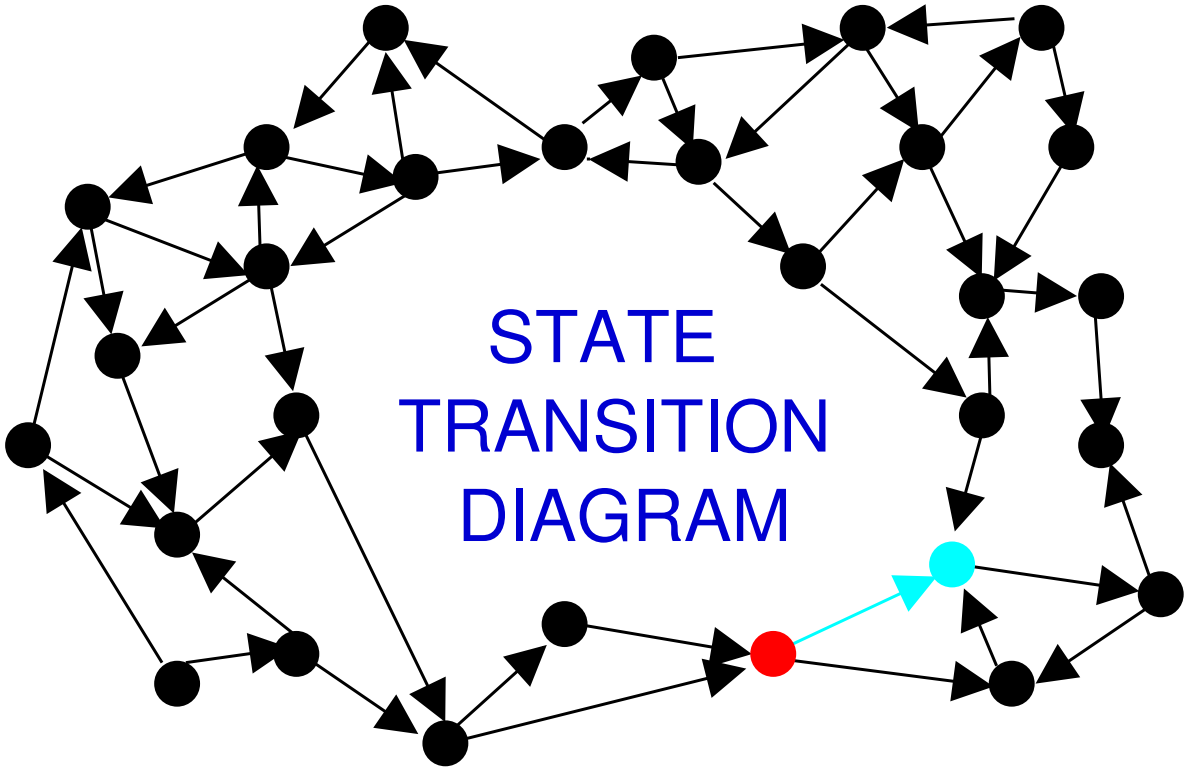
A stochastic process X(t) is a Markov process iff for all $t_0 < t_1 < .... < t_n < t_{n+1}$, the joint probability distribution of $(X(t_0), X(t_1), ..., X(t_n), X(t_{n+1}))$ is such that

$$Pr(X(t_{n+1}) = s_{i_{n+1}} \mid X(t_0) = s_{i_0}, ..., X(t_n) = s_{i_n}) = Pr(X(t_{n+1}) = s_{i_{n+1}} \mid X(t_n) = s_{i_n})$$

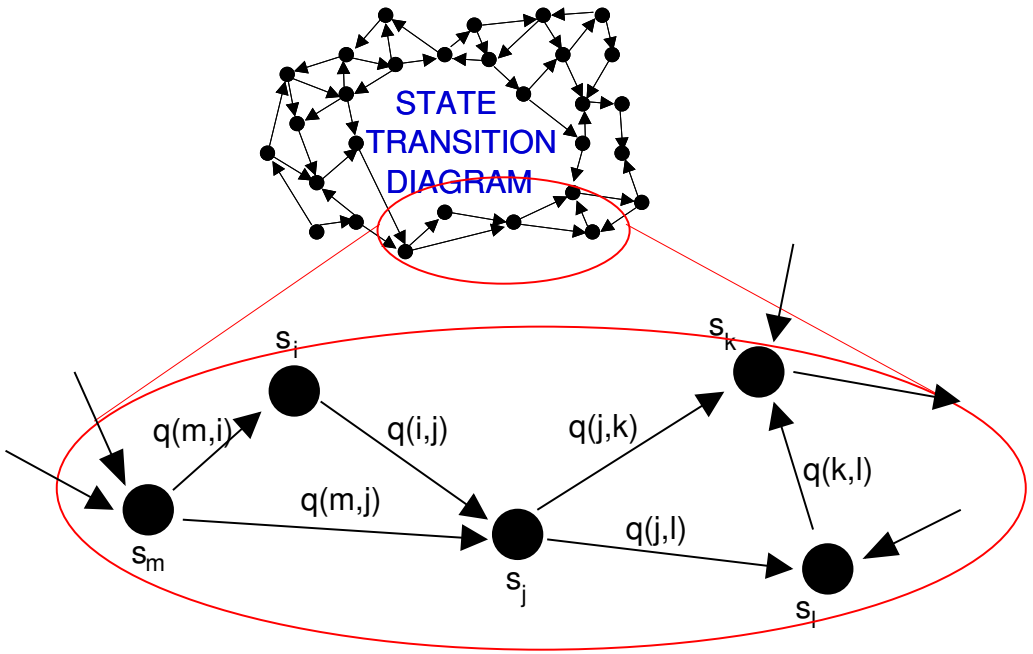
Performance Modelling using CTMC



Performance Modelling using CTMC

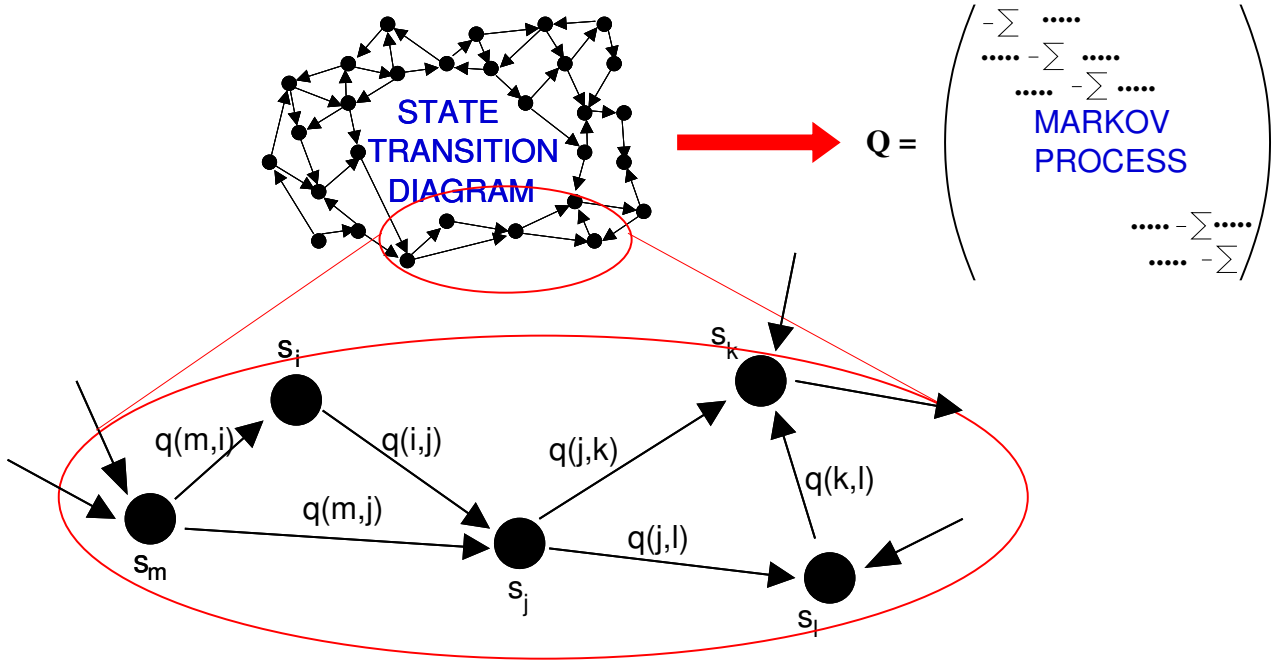


Performance Modelling using CTMC



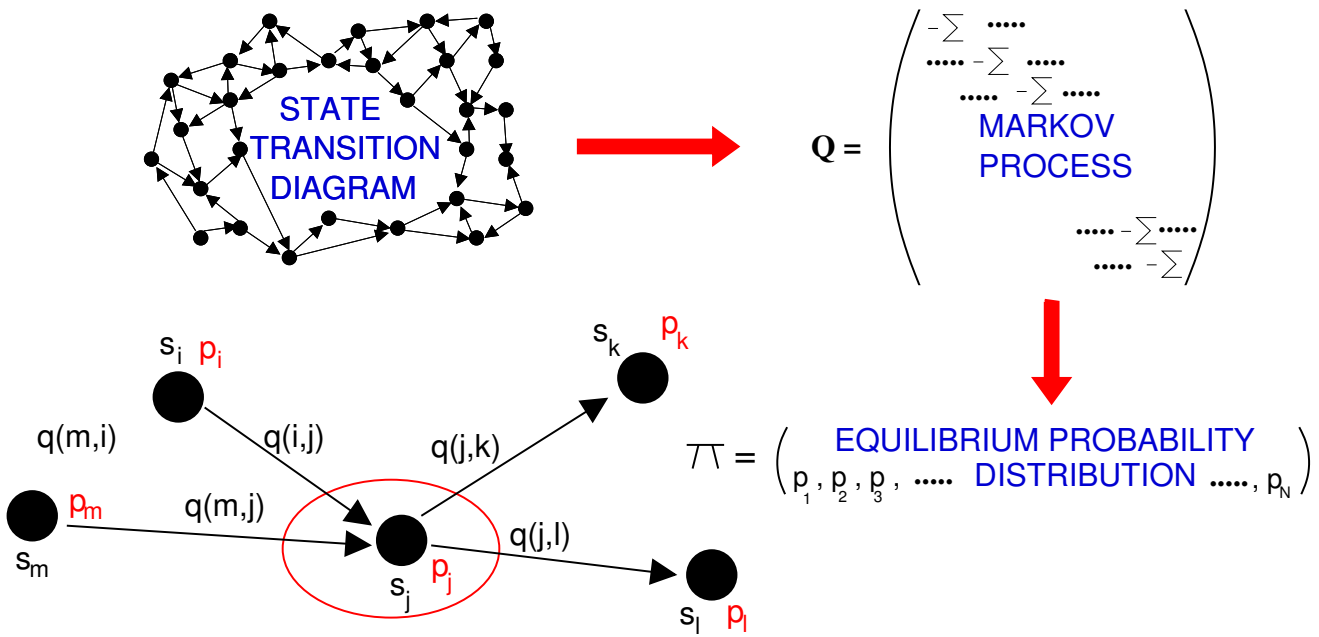
A negative exponentially distributed duration is associated with each transition.

Performance Modelling using CTMC



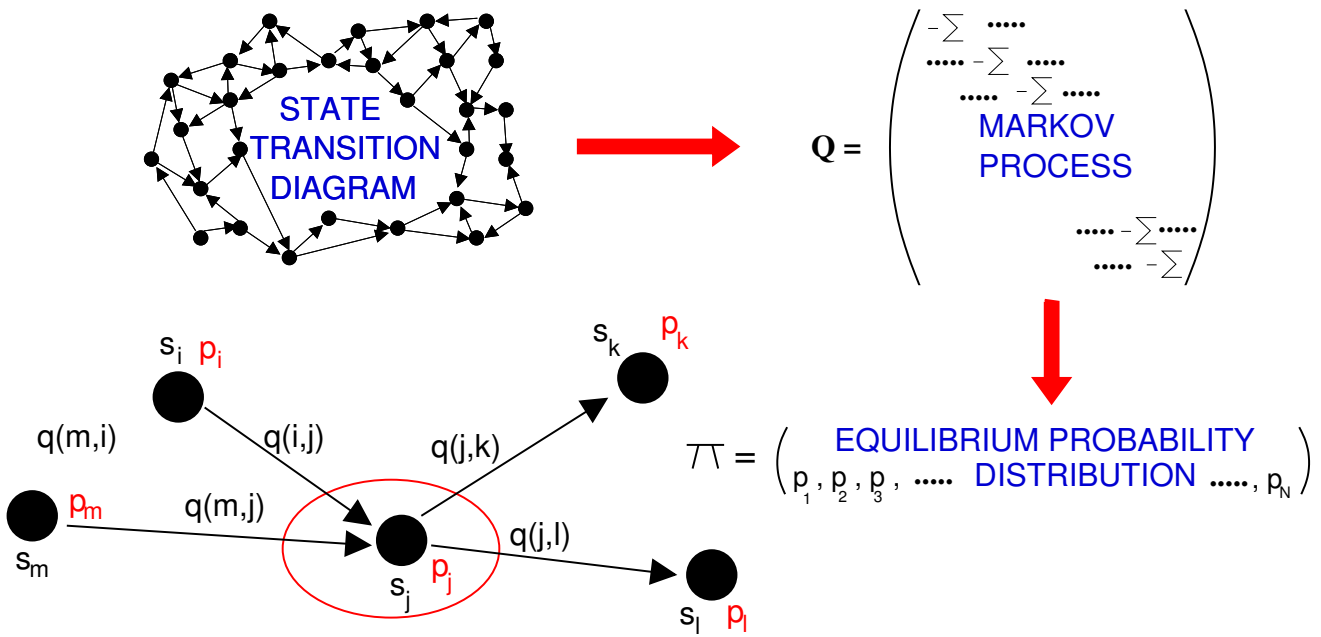
these parameters form the entries of the infinitesimal generator matrix Q

Performance Modelling using CTMC



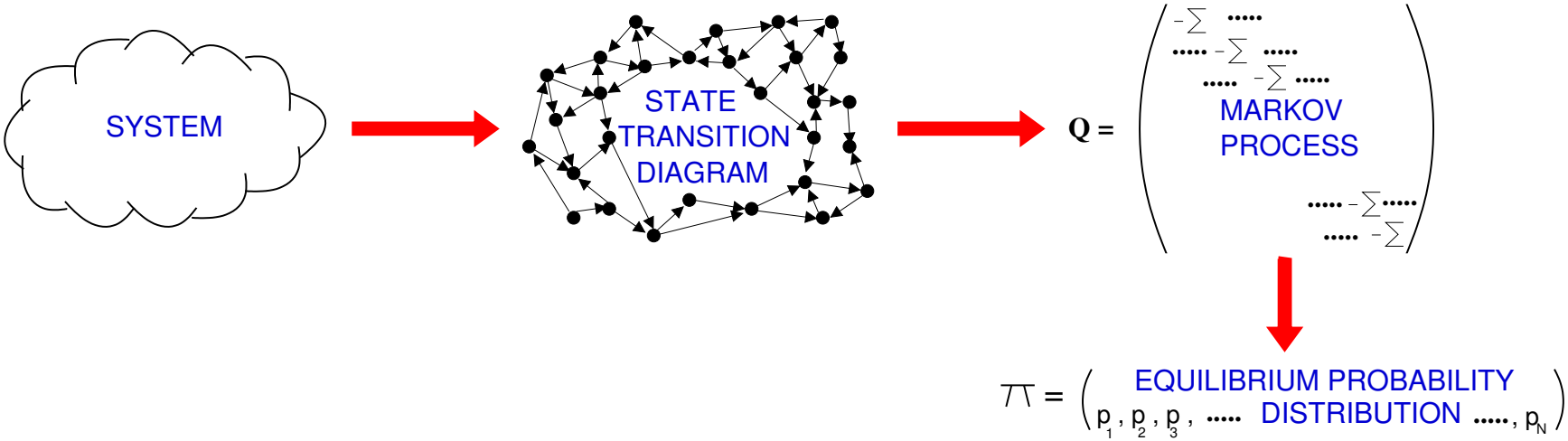
In steady state the probability flux out of a state is balanced by the flux in.

Performance Modelling using CTMC

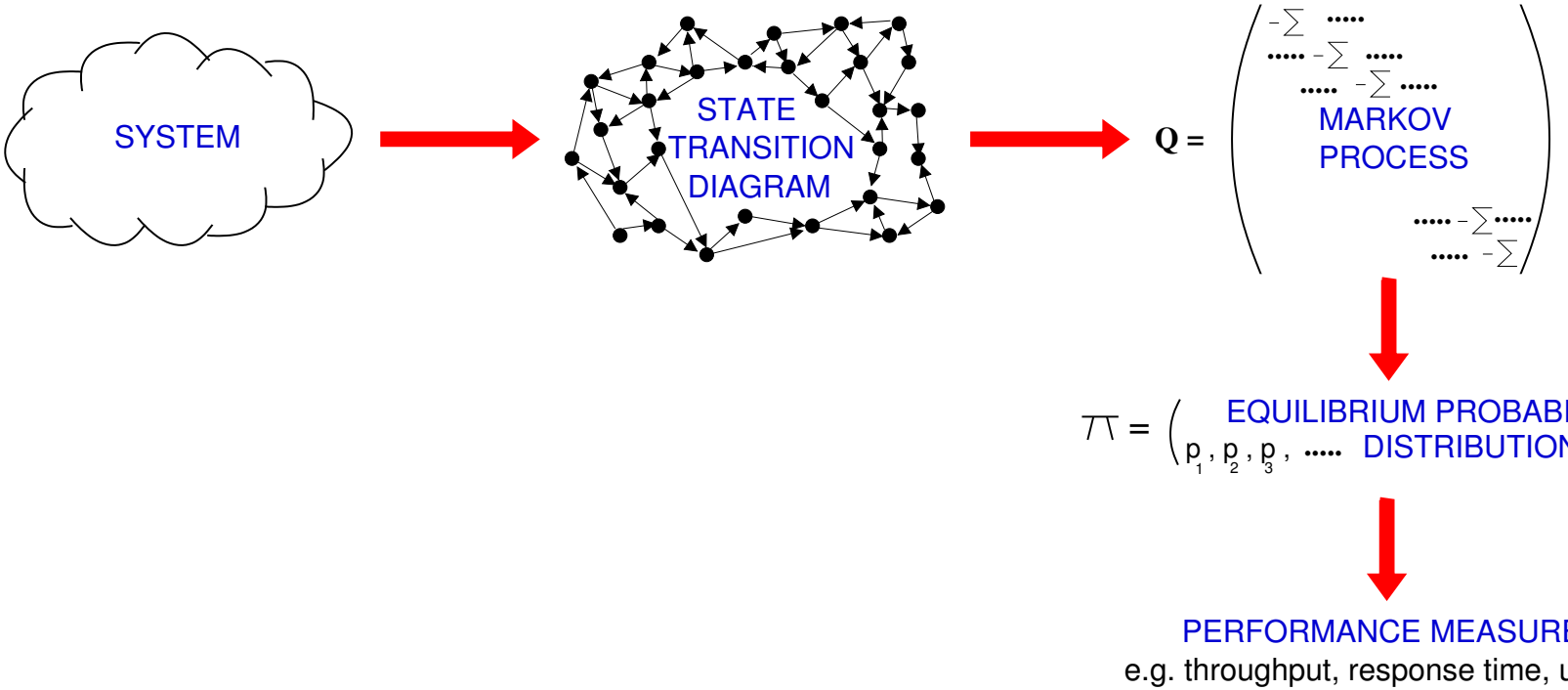


"Global balance equations" captured by $\pi Q = 0$ solved by linear algebra

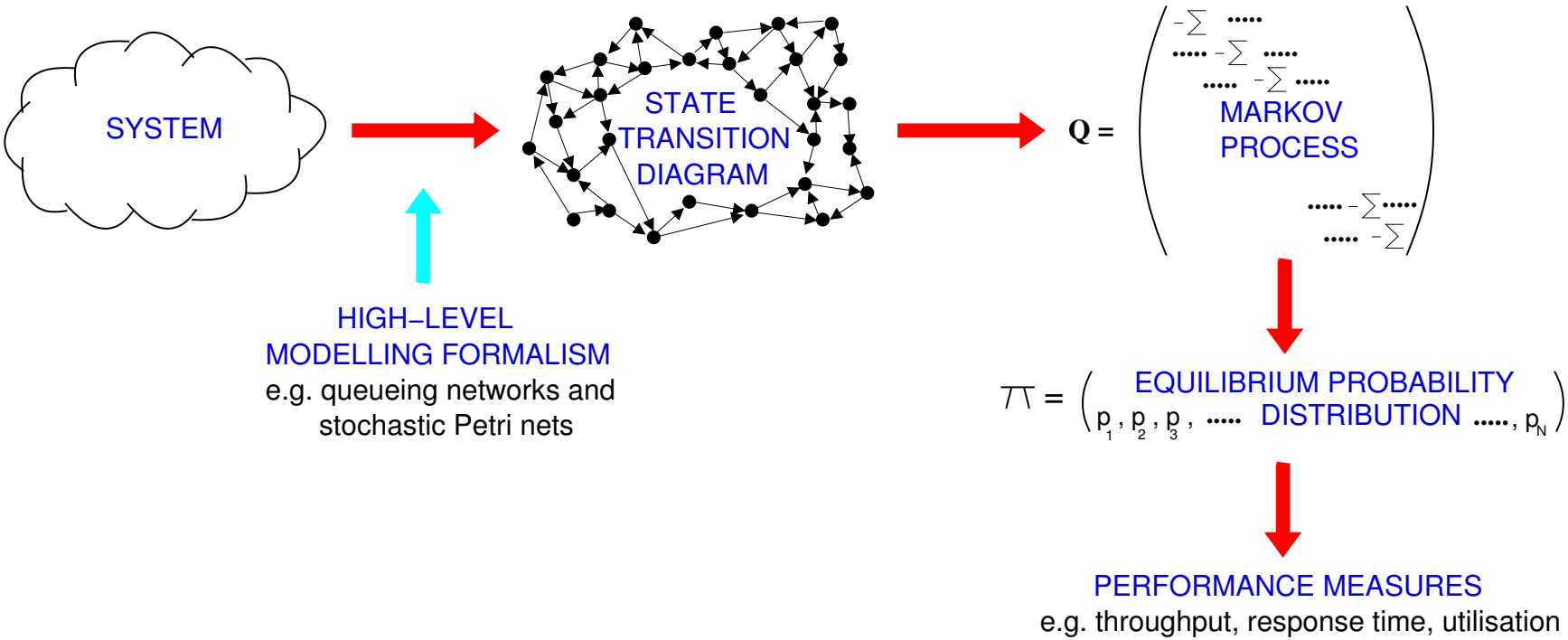
Performance Modelling using CTMC



Performance Modelling using CTMC



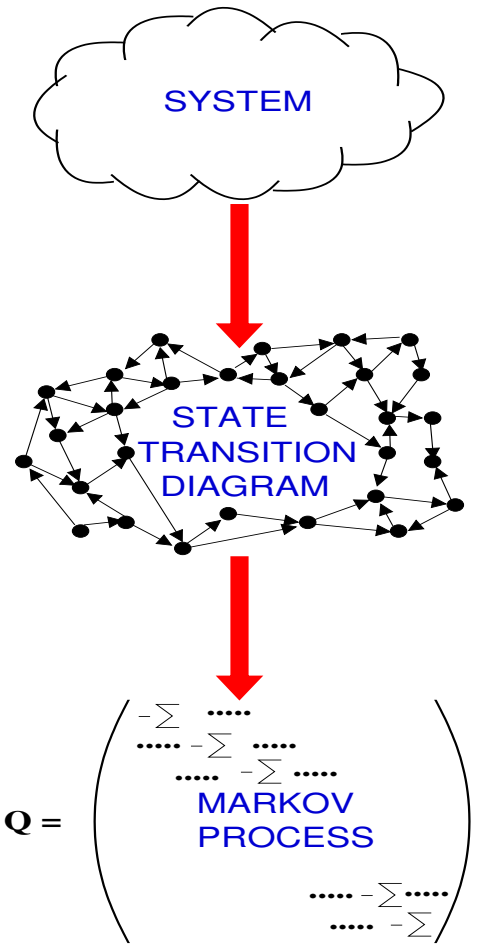
Performance Modelling using CTMC



Performance Modelling using CTMC

Model Construction

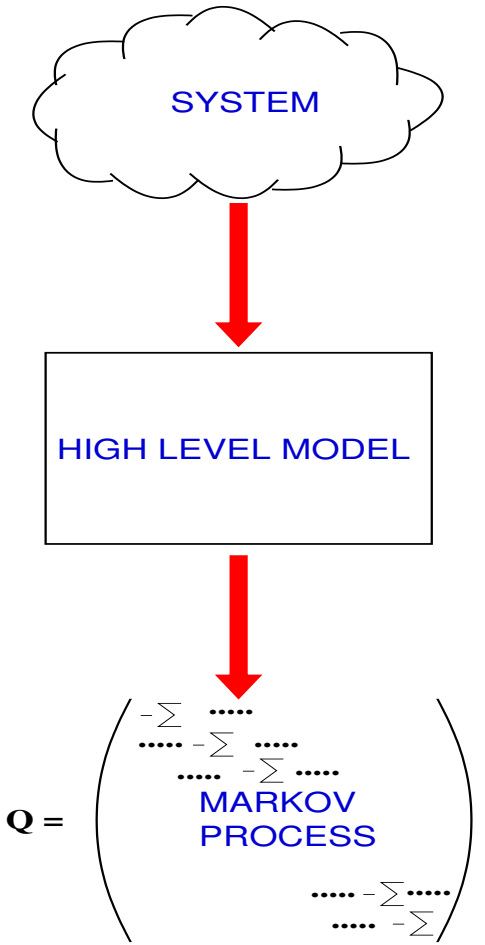
- describing the system using a high level modelling formalism
- generating the underlying CTMC



Performance Modelling using CTMC

Model Construction

- describing the system using a high level modelling formalism
- generating the underlying CTMC



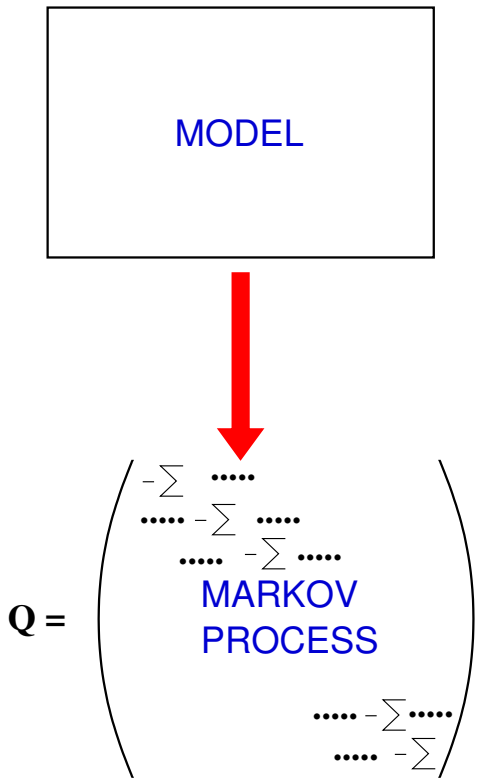
Performance Modelling using CTMC

Model Construction

- describing the system using a high level modelling formalism
- generating the underlying CTMC

Model Manipulation

- model simplification
- model aggregation



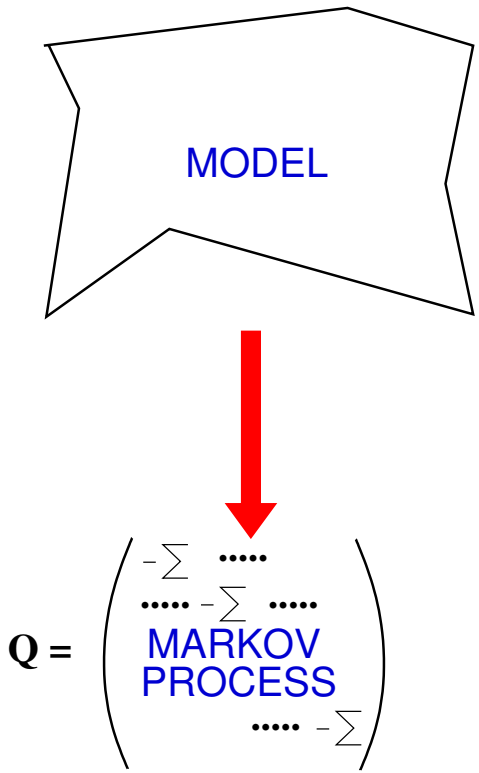
Performance Modelling using CTMC

Model Construction

- describing the system using a high level modelling formalism
- generating the underlying CTMC

Model Manipulation

- model simplification
- model aggregation



Performance Modelling using CTMC

Model Construction

- describing the system using a high level modelling formalism
- generating the underlying CTMC

Model Manipulation

- model simplification
- model aggregation

Model Solution

- solving the CTMC to find steady state probability distribution
- deriving performance measures

Outline

The PEPA project

- The PEPA project started in Edinburgh in 1991.

The PEPA project

- The PEPA project started in Edinburgh in 1991.
- It was motivated by problems encountered when carrying out performance analysis of large computer and communication systems, based on numerical analysis of Markov processes.

The PEPA project

- The PEPA project started in Edinburgh in 1991.
- It was motivated by problems encountered when carrying out performance analysis of large computer and communication systems, based on numerical analysis of Markov processes.
- Process algebras offered a compositional description technique supported by apparatus for formal reasoning.

The PEPA project

- The PEPA project started in Edinburgh in 1991.
- It was motivated by problems encountered when carrying out performance analysis of large computer and communication systems, based on numerical analysis of Markov processes.
- Process algebras offered a compositional description technique supported by apparatus for formal reasoning.
- Performance Evaluation Process Algebra (PEPA) sought to address these problems by the introduction of a suitable process algebra.

The PEPA project

- The PEPA project started in Edinburgh in 1991.
- It was motivated by problems encountered when carrying out performance analysis of large computer and communication systems, based on numerical analysis of Markov processes.
- Process algebras offered a compositional description technique supported by apparatus for formal reasoning.
- Performance Evaluation Process Algebra (PEPA) sought to address these problems by the introduction of a suitable process algebra.
- We have sought to investigate and exploit the interplay between the process algebra and the continuous time Markov chain (CTMC).

Performance Evaluation Process Algebra

- PEPA (Performance Evaluation Process Algebra) is a high-level modelling language for distributed systems. It can be used to develop models of existing systems (abstraction) or designs for proposed ones (specification).
- PEPA can capture performance information in a process algebra setting. It is a stochastic process algebra.
- For technical details the definitive reference is *A Compositional Approach to Performance Modelling*, Jane Hillston, Cambridge University Press, 1996.

Strengths of stochastic process algebras

SPAs have strengths in the areas of semantic definition, inherent compositionality and the existence of important equivalence relations (including bisimulation). This relation provides the basis for aggregation of PEPA models.

Terminology

The components in a PEPA model engage, cooperatively or individually, in activities.

Terminology

The components in a PEPA model engage, cooperatively or individually, in activities.

Each activity has an action type which corresponds to the actions of the system being modelled.

Terminology

The components in a PEPA model engage, cooperatively or individually, in activities.

Each activity has an action type which corresponds to the actions of the system being modelled.

To represent unimportant or unknown actions there is a distinguished action type, τ.

Quantitative aspects

Every activity in PEPA has an associated activity rate which may be any positive real number, or the distinguished symbol "$\top$", meaning unspecified, read as 'top'.

Quantitative aspects

Every activity in PEPA has an associated activity rate which may be any positive real number, or the distinguished symbol "$\top$", meaning unspecified, read as 'top'.

Components and activities are primitives. PEPA also provides a small set of combinators.

PEPA syntax

$$
\begin{array}{lcll}
S & ::= & (\alpha,\, r).S & \text{(prefix)} \\
 & | & S_1 + S_2 & \text{(choice)} \\
 & | & X & \text{(variable)} \\
C & ::= & C_1 \underset{L}{\bowtie} C_2 & \text{(cooperation)} \\
 & | & C \,/\, L & \text{(hiding)} \\
 & | & S & \text{(sequential)}
\end{array}
$$

PEPA: informal semantics (sequential sublanguage)

$(\alpha, r).S$

The activity (α, r) takes time Δt (drawn from the exponential distribution with parameter r).

$S_1 + S_2$

In this choice either S_1 or S_2 will complete an activity first. The other is discarded.

PEPA: informal semantics (combinators)

$C_1 \underset{L}{\bowtie} C_2$

All activities of C_1 and C_2 with types in L are shared: others remain individual.
NOTATION: write $C_1 \parallel C_2$ if L is empty.

C / L

Activities of C with types in L are hidden (τ type activities) to be thought of as internal delays.

Expansion Law

$$P \underset{L}{\bowtie} Q =$$

Expansion Law

$$P \underset{L}{\bowtie} Q =$$

$$\sum\{(\alpha, r).(P' \underset{L}{\bowtie} Q) : \; P \xrightarrow{(\alpha,r)} P';\; \alpha \notin L\} \; +$$

Expansion Law

$$P \underset{L}{\bowtie} Q =$$

$$\sum\{(\alpha, r).(P' \underset{L}{\bowtie} Q) : \; P \xrightarrow{(\alpha,r)} P';\; \alpha \notin L\} \; +$$

$$\sum\{(\alpha, r).(P \underset{L}{\bowtie} Q') : \; Q \xrightarrow{(\alpha,r)} Q';\; \alpha \notin L\} \; +$$

Expansion Law

$$P \underset{L}{\bowtie} Q =$$

$$\sum\{(\alpha, r).(P' \underset{L}{\bowtie} Q) : \; P \xrightarrow{(\alpha,r)} P';\; \alpha \notin L\} \; +$$

$$\sum\{(\alpha, r).(P \underset{L}{\bowtie} Q') : \; Q \xrightarrow{(\alpha,r)} Q';\; \alpha \notin L\} \; +$$

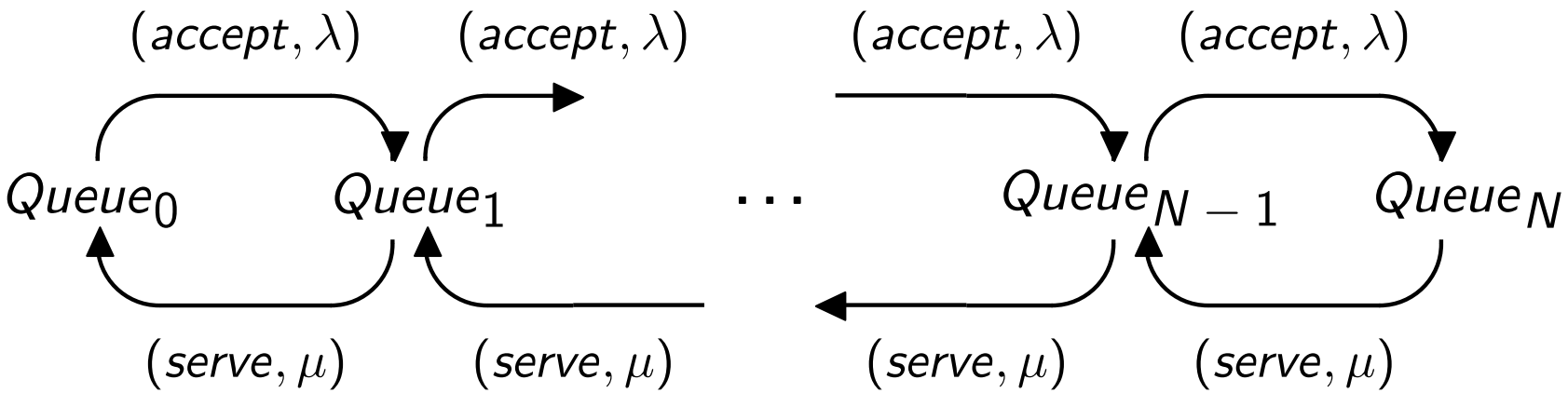
$$\sum\{(\alpha, r).(P' \underset{L}{\bowtie} Q') :$$

$$P \xrightarrow{(\alpha,r_1)} P';\; Q \xrightarrow{(\alpha,r_2)} Q';\; \alpha \in L\}$$

Example: M/M/1/N/N queue

$$
\begin{aligned}
Arrival_0 &\stackrel{def}{=} (accept, \lambda).Arrival_1 \\
Arrival_i &\stackrel{def}{=} (accept, \lambda).Arrival_{i+1} + (serve, \top).Arrival_{i-1} \quad (0 < i < N) \\
Arrival_N &\stackrel{def}{=} (serve, \top).Arrival_{N-1} \\
Server &\stackrel{def}{=} (serve, \mu).Server
\end{aligned}
$$

Example: M/M/1/N/N queue



$$Queue_i \equiv Arrival_i \underset{\{serve\}}{\bowtie} Server$$

Synchronisation

What should be the impact of synchronisation on rate? There are many possibilities.

- Restrict synchronisations to have one active partner and one passive partner.

Synchronisation

What should be the impact of synchronisation on rate? There are many possibilities.

- Restrict synchronisations to have one active partner and one passive partner.
- Choose a function which satisfies a small number of algebraic properties.

Synchronisation

What should be the impact of synchronisation on rate? There are many possibilities.

- Restrict synchronisations to have one active partner and one passive partner.
- Choose a function which satisfies a small number of algebraic properties.
- Have the rate limited by the slowest participant in terms of apparent rate. This is the approach adopted by PEPA.

Bounded capacity

Within the cooperation framework, PEPA assumes bounded capacity: that is, a component cannot be made to perform an activity faster by cooperation, so the rate of a shared activity is the minimum of the apparent rates of the activity in the cooperating components.

Apparent rate

The total capacity of a component P to carry out activities of type α is termed the apparent rate of α in P, denoted $r_\alpha(P)$.

Apparent rate

The total capacity of a component P to carry out activities of type α is termed the apparent rate of α in P, denoted $r_\alpha(P)$.

It is used heavily when calculating the pairwise cooperation rate: when cooperating with another component, the bounded capacity principle ensures that the overall rate of cooperation does not exceed either of the consistuent apparent rates.

Apparent rate: definition

To summarise the original ruleset from [Hillston 96], the apparent rate function can be defined as:

$$r_\alpha(P) = \sum_{P \xrightarrow{(\alpha, \lambda_i)}} \lambda_i \tag{1}$$

where $\lambda_i \in \mathbb{R}^+ \cup \{n\top \mid n \in \mathbb{Q}, n > 0\}$, $n\top$ is shorthand for $n \times \top$ and $\top$ represents the passive action rate that inherits the rate of the coaction from the cooperating component.

Properties of $\top$ (the "unspecified" symbol)

$\top$ requires the following arithmetic rules:

$$
\begin{aligned}
m\top < n\top &\quad : \quad \text{for } m < n \text{ and } m, n \in \mathbb{Q} \\
r < n\top &\quad : \quad \text{for all } r \in \mathbb{R}, n \in \mathbb{Q} \\
m\top + n\top = (m+n)\top &\quad : \quad m, n \in \mathbb{Q} \\
\frac{m\top}{n\top} = \frac{m}{n} &\quad : \quad m, n \in \mathbb{Q}
\end{aligned}
$$

Properties of $\top$ (the "unspecified" symbol)

$\top$ requires the following arithmetic rules:

$$
\begin{aligned}
m\top < n\top &\quad : \quad \text{for } m < n \text{ and } m, n \in \mathbb{Q} \\
r < n\top &\quad : \quad \text{for all } r \in \mathbb{R}, n \in \mathbb{Q} \\
m\top + n\top = (m+n)\top &\quad : \quad m, n \in \mathbb{Q} \\
\frac{m\top}{n\top} = \frac{m}{n} &\quad : \quad m, n \in \mathbb{Q}
\end{aligned}
$$

Note that $(r + n\top)$ is undefined for all $r \in \mathbb{R}$ in PEPA therefore disallowing sequential components which enable both active and passive actions in the same action type at the same time, e.g. $(\alpha, \lambda).P + (\alpha, \top).P'$.

Outline

Random experiments and events

- To apply probability theory to the process under study, we view it as a random experiment.

Random experiments and events

- To apply probability theory to the process under study, we view it as a random experiment.
- The sample space of a random experiment is the set of all individual outcomes of the experiment.

Random experiments and events

- To apply probability theory to the process under study, we view it as a random experiment.
- The sample space of a random experiment is the set of all individual outcomes of the experiment.
- These individual outcomes are also called sample points or elementary events.

Random experiments and events

- To apply probability theory to the process under study, we view it as a random experiment.
- The sample space of a random experiment is the set of all individual outcomes of the experiment.
- These individual outcomes are also called sample points or elementary events.
- An event is a subset of a sample space.

Random variables

We are interested in the dynamics of a system as events happen over time.

A function which associates a (real-valued) number with the outcome of an experiment is known as a random variable.

Formally, a random variable X is a real-valued function defined on a sample space Ω.

Measurable functions

If X is a random variable, and x is a real number, we write $X \leq x$ for the event

$$\{\, \omega : \omega \in \Omega \text{ and } X(\omega) \leq x \,\}$$

Measurable functions

If X is a random variable, and x is a real number, we write $X \leq x$ for the event

$$\{\omega : \omega \in \Omega \text{ and } X(\omega) \leq x\}$$

and we write $X = x$ for the event

$$\{\omega : \omega \in \Omega \text{ and } X(\omega) = x\}$$

Measurable functions

If X is a random variable, and x is a real number, we write $X \leq x$ for the event

$$\{\omega : \omega \in \Omega \text{ and } X(\omega) \leq x\}$$

and we write $X = x$ for the event

$$\{\omega : \omega \in \Omega \text{ and } X(\omega) = x\}$$

Another property required of a random variable is that the set $X \leq x$ is an event for each real x. This is necessary so that probability calculations can be made. A function having this property is said to be a measurable function or measurable in the Borel sense.

Distribution function

For each random variable X we define its distribution function F for each real x by

$$F(x) = \Pr[X \leq x]$$

Distribution function

For each random variable X we define its distribution function F for each real x by

$$F(x) = \Pr[X \leq x]$$

We associate another function $p(\cdot)$, called the probability mass function, with X (pmf), for each x:

$$p(x) = \Pr[X = x]$$

Distribution function

For each random variable X we define its distribution function F for each real x by

$$F(x) = \Pr[X \leq x]$$

We associate another function $p(\cdot)$, called the probability mass function, with X (pmf), for each x:

$$p(x) = \Pr[X = x]$$

A random variable X is continuous if $p(x) = 0$ for all real x.

Distribution function

For each random variable X we define its distribution function F for each real x by

$$F(x) = \Pr[X \leq x]$$

We associate another function $p(\cdot)$, called the probability mass function, with X (pmf), for each x:

$$p(x) = \Pr[X = x]$$

A random variable X is continuous if $p(x) = 0$ for all real x.

(If X is a *continuous* random variable, then X can assume infinitely many values, and so it is reasonable that the probability of its assuming any *specific* value we choose beforehand is zero.)

Exponential random variables, distribution function

The random variable X is said to be an *exponential random variable with parameter* λ $(\lambda > 0)$ or to have an exponential distribution with parameter λ if it has the distribution function

$$F(x) = \begin{cases} 1 - e^{-\lambda x} & \text{for } x > 0 \\ 0 & \text{for } x \leq 0 \end{cases}$$

Exponential random variables, distribution function

The random variable X is said to be an *exponential random variable with parameter* λ ($\lambda > 0$) or to have an exponential distribution with parameter λ if it has the distribution function

$$F(x) = \begin{cases} 1 - e^{-\lambda x} & \text{for } x > 0 \\ 0 & \text{for } x \leq 0 \end{cases}$$

Some authors call this distribution the negative exponential distribution.

Exponential random variables, density function

The density function $f = dF/dx$ is given by

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$$

Mean, or expected value

If X is a continuous random variable with density function $f(\cdot)$, we define the mean or expected value of X, $\mu = E[X]$ by

$$\mu = E[X] = \int_{-\infty}^{\infty} xf(x)dx$$

Mean, or expected value

If X is a continuous random variable with density function $f(\cdot)$, we define the mean or expected value of X, $\mu = E[X]$ by

$$\mu = E[X] = \int_{-\infty}^{\infty} xf(x)dx$$

If X is a discrete random variable with probability mass function $p(\cdot)$, we define the mean or expected value of $X \in S$, $\mu = E[X]$ by

$$E(X) = \sum_{x \in S} xp(x)$$

Mean, or expected value, of the exponential distribution

Suppose X has an exponential distribution with parameter $\lambda > 0$. Then

$$\mu = E[X] = \int_{-\infty}^{\infty} x\lambda e^{-\lambda x} dx$$

Mean, or expected value, of the exponential distribution

Suppose X has an exponential distribution with parameter $\lambda > 0$. Then

$$\mu = E[X] = \int_{-\infty}^{\infty} x\lambda e^{-\lambda x} dx = \frac{1}{\lambda}$$

Exponential inter-event time distribution

The time interval between successive events can also be deduced.

Let $F(t)$ be the distribution function of T, the time between events. Consider $\Pr(T > t) = 1 - F(t)$:

$$\Pr(T > t) \quad = \quad \Pr(\text{No events in an interval of length } t)$$

Exponential inter-event time distribution

The time interval between successive events can also be deduced.

Let $F(t)$ be the distribution function of T, the time between events. Consider $\Pr(T > t) = 1 - F(t)$:

$$\begin{aligned} \Pr(T > t) &= \Pr(\text{No events in an interval of length } t) \\ &= 1 - F(t) \end{aligned}$$

Exponential inter-event time distribution

The time interval between successive events can also be deduced.

Let $F(t)$ be the distribution function of T, the time between events. Consider $\Pr(T > t) = 1 - F(t)$:

$$\begin{aligned} \Pr(T > t) &= \Pr(\text{No events in an interval of length } t) \\ &= 1 - F(t) \\ &= 1 - (1 - e^{-\lambda t}) \end{aligned}$$

Exponential inter-event time distribution

The time interval between successive events can also be deduced.

Let $F(t)$ be the distribution function of T, the time between events. Consider $\Pr(T > t) = 1 - F(t)$:

$$\begin{aligned} \Pr(T > t) &= \Pr(\text{No events in an interval of length } t) \\ &= 1 - F(t) \\ &= 1 - (1 - e^{-\lambda t}) \\ &= e^{-\lambda t} \end{aligned}$$

Memoryless property of the exponential distribution

The memoryless property of the exponential distribution is so called because the time to the next event is independent of when the last event occurred.

Memoryless property of the exponential distribution

Suppose that the last event was at time 0. What is the probability that the next event will be after $t + s$, given that time t has elapsed since the last event, and no events have occurred?

Memoryless property of the exponential distribution

Suppose that the last event was at time 0. What is the probability that the next event will be after $t + s$, given that time t has elapsed since the last event, and no events have occurred?

$$\Pr(T > t + s \mid T > t) = \frac{\Pr(T > t + s \text{ and } T > t)}{\Pr(T > t)}$$

Memoryless property of the exponential distribution

Suppose that the last event was at time 0. What is the probability that the next event will be after $t + s$, given that time t has elapsed since the last event, and no events have occurred?

$$\begin{aligned} \Pr(T > t + s \mid T > t) &= \frac{\Pr(T > t + s \text{ and } T > t)}{\Pr(T > t)} \\ &= \frac{e^{-\lambda(t+s)}}{e^{-\lambda t}} \end{aligned}$$

Memoryless property of the exponential distribution

Suppose that the last event was at time 0. What is the probability that the next event will be after $t + s$, given that time t has elapsed since the last event, and no events have occurred?

$$\begin{aligned} \Pr(T > t + s \mid T > t) &= \frac{\Pr(T > t + s \text{ and } T > t)}{\Pr(T > t)} \\ &= \frac{e^{-\lambda(t+s)}}{e^{-\lambda t}} \\ &= e^{-\lambda s} \end{aligned}$$

Memoryless property of the exponential distribution

Suppose that the last event was at time 0. What is the probability that the next event will be after $t + s$, given that time t has elapsed since the last event, and no events have occurred?

$$\begin{aligned} \Pr(T > t + s \mid T > t) &= \frac{\Pr(T > t + s \text{ and } T > t)}{\Pr(T > t)} \\ &= \frac{e^{-\lambda(t+s)}}{e^{-\lambda t}} \\ &= e^{-\lambda s} \end{aligned}$$

This value is independent of t (and so the time already spent has not been remembered).

PEPA activities and rates

When enabled an activity, $a = (\alpha, \lambda)$, will delay for a period determined by its associated distribution function, i.e. the probability that the activity a happens within a period of time of length t is $F_a(t) = 1 - e^{-\lambda t}$.

PEPA activities and rates

- We can think of this as the activity setting a timer whenever it becomes enabled.

PEPA activities and rates

- We can think of this as the activity setting a timer whenever it becomes enabled.
- The time allocated to the timer is determined by the exponential distribution via the rate of the activity.

PEPA activities and rates

- We can think of this as the activity setting a timer whenever it becomes enabled.
- The time allocated to the timer is determined by the exponential distribution via the rate of the activity.
- If several activities are enabled at the same time each will have its own associated timer.

PEPA activities and rates

- We can think of this as the activity setting a timer whenever it becomes enabled.
- The time allocated to the timer is determined by the exponential distribution via the rate of the activity.
- If several activities are enabled at the same time each will have its own associated timer.
- When the first timer finishes that activity takes place—the activity is said to complete or succeed.

PEPA activities and rates

- We can think of this as the activity setting a timer whenever it becomes enabled.
- The time allocated to the timer is determined by the exponential distribution via the rate of the activity.
- If several activities are enabled at the same time each will have its own associated timer.
- When the first timer finishes that activity takes place—the activity is said to complete or succeed.
- This means that the activity is considered to "happen": an external observer will witness the event of activity of type α.

PEPA activities and rates

- We can think of this as the activity setting a timer whenever it becomes enabled.
- The time allocated to the timer is determined by the exponential distribution via the rate of the activity.
- If several activities are enabled at the same time each will have its own associated timer.
- When the first timer finishes that activity takes place—the activity is said to complete or succeed.
- This means that the activity is considered to "happen": an external observer will witness the event of activity of type α.
- An activity may be preempted, or aborted, if another one completes first.

PEPA and Markov processes

In a PEPA model if we define the stochastic process $X(t)$, such that $X(t) = C_i$ indicates that the system behaves as component C_i at time t, then $X(t)$ is a Markov process which can be characterised by a matrix, $\boldsymbol{Q}$.

PEPA and Markov processes

In a PEPA model if we define the stochastic process $X(t)$, such that $X(t) = C_i$ indicates that the system behaves as component C_i at time t, then $X(t)$ is a Markov process which can be characterised by a matrix, $\boldsymbol{Q}$.

A stationary or equilibrium probability distribution, $\pi(\cdot)$, exists for every time-homogeneous irreducible Markov process whose states are all positive-recurrent.

PEPA and Markov processes

In a PEPA model if we define the stochastic process $X(t)$, such that $X(t) = C_i$ indicates that the system behaves as component C_i at time t, then $X(t)$ is a Markov process which can be characterised by a matrix, $\boldsymbol{Q}$.

A stationary or equilibrium probability distribution, $\pi(\cdot)$, exists for every time-homogeneous irreducible Markov process whose states are all positive-recurrent.

This distribution is found by solving the global balance equation

$$\pi \boldsymbol{Q} = \boldsymbol{0}$$

subject to the normalisation condition

$$\sum \pi(C_i) = 1.$$

PEPA and time

All PEPA models are time-homogeneous since all activities are time-homogeneous: the rate and type of activities enabled by a component are independent of time.

PEPA and irreducibility and positive-recurrence

The other conditions, irreducibility and positive-recurrent states, are easily expressed in terms of the derivation graph of the PEPA model.

PEPA and irreducibility and positive-recurrence

The other conditions, irreducibility and positive-recurrent states, are easily expressed in terms of the derivation graph of the PEPA model.

We only consider PEPA models with a finite number of states so if the model is irreducible then all states must be positive-recurrent i.e. the derivation graph is strongly connected.

PEPA and irreducibility and positive-recurrence

The other conditions, irreducibility and positive-recurrent states, are easily expressed in terms of the derivation graph of the PEPA model.

We only consider PEPA models with a finite number of states so if the model is irreducible then all states must be positive-recurrent i.e. the derivation graph is strongly connected.

In terms of the PEPA model this means that all behaviours of the system must be recurrent; in particular, for every choice, whichever path is chosen it must eventually return to the point where the choice can be made again, possibly with a different outcome.

Outline

Case study: active badges

We have used the PEPA modelling language to analyse the configuration of a location tracking system based on active badges.

Active badges transmit unique infra-red signals which are detected by networked sensors. These report locations back to a central database.

Case study: active badges

The badges are battery-powered and the tradeoff in the system is between the conservation of battery power and the accuracy of the information harvested from the sensors.

Case study: active badges

The badges are battery-powered and the tradeoff in the system is between the conservation of battery power and the accuracy of the information harvested from the sensors.

When transmissions from badges collide, the badges sleep for a randomly determined time before retrying.

Active badges: the PEPA model

The PEPA model of this system tracks the progress of one badge-wearer around three connected corridors (numbered 14, 15 and 16).

Active badges: the PEPA model

The PEPA model of this system tracks the progress of one badge-wearer around three connected corridors (numbered 14, 15 and 16).

The activities which are performed in the system include the badge registering with a sensor (at rate r), the person moving to another corridor (at rate m) and a sensor reporting back to the central database (at rate s).

Active badges: the PEPA model

Person

$$
\begin{aligned}
P_{14} &\stackrel{def}{=} (reg_{14}, r).P_{14} + (move_{15}, m).P_{15} \\
P_{15} &\stackrel{def}{=} (reg_{15}, r).P_{15} + (move_{14}, m).P_{14} + (move_{16}, m).P_{16} \\
P_{16} &\stackrel{def}{=} (reg_{16}, r).P_{16} + (move_{15}, m).P_{15}
\end{aligned}
$$

Active badges: the PEPA model

Person

$$
\begin{aligned}
P_{14} &\stackrel{def}{=} (reg_{14}, r).P_{14} + (move_{15}, m).P_{15} \\
P_{15} &\stackrel{def}{=} (reg_{15}, r).P_{15} + (move_{14}, m).P_{14} + (move_{16}, m).P_{16} \\
P_{16} &\stackrel{def}{=} (reg_{16}, r).P_{16} + (move_{15}, m).P_{15}
\end{aligned}
$$

Sensor

$$
\begin{aligned}
S_{14} &\stackrel{def}{=} (reg_{14}, \top).(rep_{14}, s).S_{14} \\
S_{15} &\stackrel{def}{=} (reg_{15}, \top).(rep_{15}, s).S_{15} \\
S_{16} &\stackrel{def}{=} (reg_{16}, \top).(rep_{16}, s).S_{16}
\end{aligned}
$$

Active badges: the PEPA model

Database

$$
\begin{aligned}
DB_{14} &\stackrel{def}{=} (rep_{14}, \top).DB_{14} + (rep_{15}, \top).DB_{15} + (rep_{16}, \top).DB_{16} \\
DB_{15} &\stackrel{def}{=} (rep_{14}, \top).DB_{14} + (rep_{15}, \top).DB_{15} + (rep_{16}, \top).DB_{16} \\
DB_{16} &\stackrel{def}{=} (rep_{14}, \top).DB_{14} + (rep_{15}, \top).DB_{15} + (rep_{16}, \top).DB_{16}
\end{aligned}
$$

Active badges: the PEPA model

Database

$$
\begin{aligned}
DB_{14} &\stackrel{def}{=} (rep_{14}, \top).DB_{14} + (rep_{15}, \top).DB_{15} + (rep_{16}, \top).DB_{16} \\
DB_{15} &\stackrel{def}{=} (rep_{14}, \top).DB_{14} + (rep_{15}, \top).DB_{15} + (rep_{16}, \top).DB_{16} \\
DB_{16} &\stackrel{def}{=} (rep_{14}, \top).DB_{14} + (rep_{15}, \top).DB_{15} + (rep_{16}, \top).DB_{16}
\end{aligned}
$$

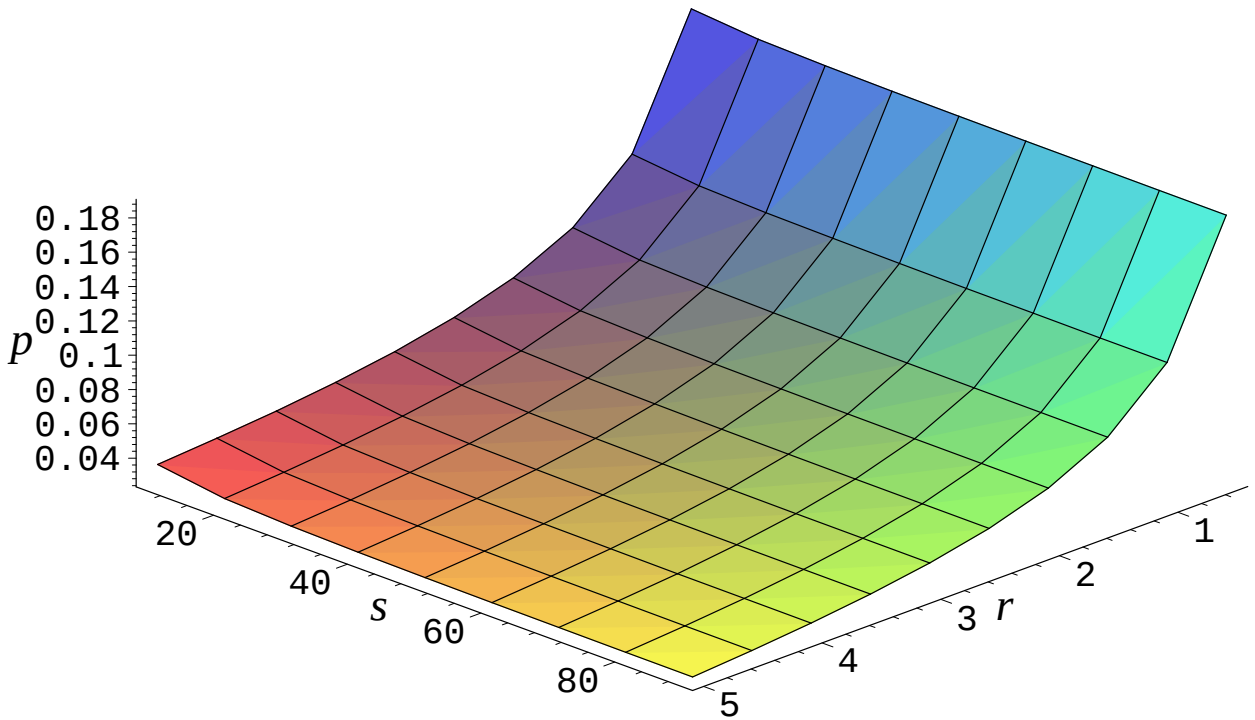
System

$$P_{14} \underset{L}{\bowtie} (S_{14} \parallel S_{15} \parallel S_{16}) \underset{M}{\bowtie} DB_{14}$$

where $L = \{ reg_{14}, reg_{15}, reg_{16} \}$

$M = \{ rep_{14}, rep_{15}, rep_{16} \}$

Probability that the database holds inaccurate information



Outline

A high-level language for performance modelling

PEPA is a high-level language for performance modelling. PEPA models describe stochastic (in fact, Markovian) processes.

A high-level language for performance modelling

PEPA is a high-level language for performance modelling. PEPA models describe stochastic (in fact, Markovian) processes.

Without a high-level modelling language, the modeller would be forced to work with unstructured matrix representations of stochastic processes.

A high-level language for performance modelling

PEPA is a high-level language for performance modelling. PEPA models describe stochastic (in fact, Markovian) processes.

Without a high-level modelling language, the modeller would be forced to work with unstructured matrix representations of stochastic processes.

Process algebras are useful because they allow the definition of equivalence relations between model components and these relations may be used in model simplification.

The PEPA Eclipse Plug-in

Calculating by hand the transitions of a PEPA model and subsequently expressing these in a form which was suitable for solution was a tedious task prone to errors. The PEPA Eclipse Plug-in relieves the modeller of this work.

The PEPA Eclipse Plug-in: functionality

The plug-in will report errors in the model function:

- deadlock,
- absorbing states,
- static synchronisation mismatch (cooperations which do not involve active participants).

The PEPA Eclipse Plug-in: functionality

The plug-in will report errors in the model function:

- deadlock,
- absorbing states,
- static synchronisation mismatch (cooperations which do not involve active participants).

The plug-in also generates the transition graph of the model, computes the number of states, formulates the Markov process matrix $\boldsymbol{Q}$ and communicates the matrix to a solver.

The PEPA Eclipse Plug-in: functionality

The plug-in will report errors in the model function:

- deadlock,
- absorbing states,
- static synchronisation mismatch (cooperations which do not involve active participants).

The plug-in also generates the transition graph of the model, computes the number of states, formulates the Markov process matrix $\boldsymbol{Q}$ and communicates the matrix to a solver.

The plug-in provides a simple pattern language for selecting states from the stationary distribution.

PEPA Eclipse Plug-In input

$$P_1 \stackrel{def}{=} (start, r_1).P_2 \qquad P_2 \stackrel{def}{=} (run, r_2).P_3 \qquad P_3 \stackrel{def}{=} (stop, r_3).P_1$$

$$P_1 \parallel P_1$$

PEPA Eclipse Plug-In input

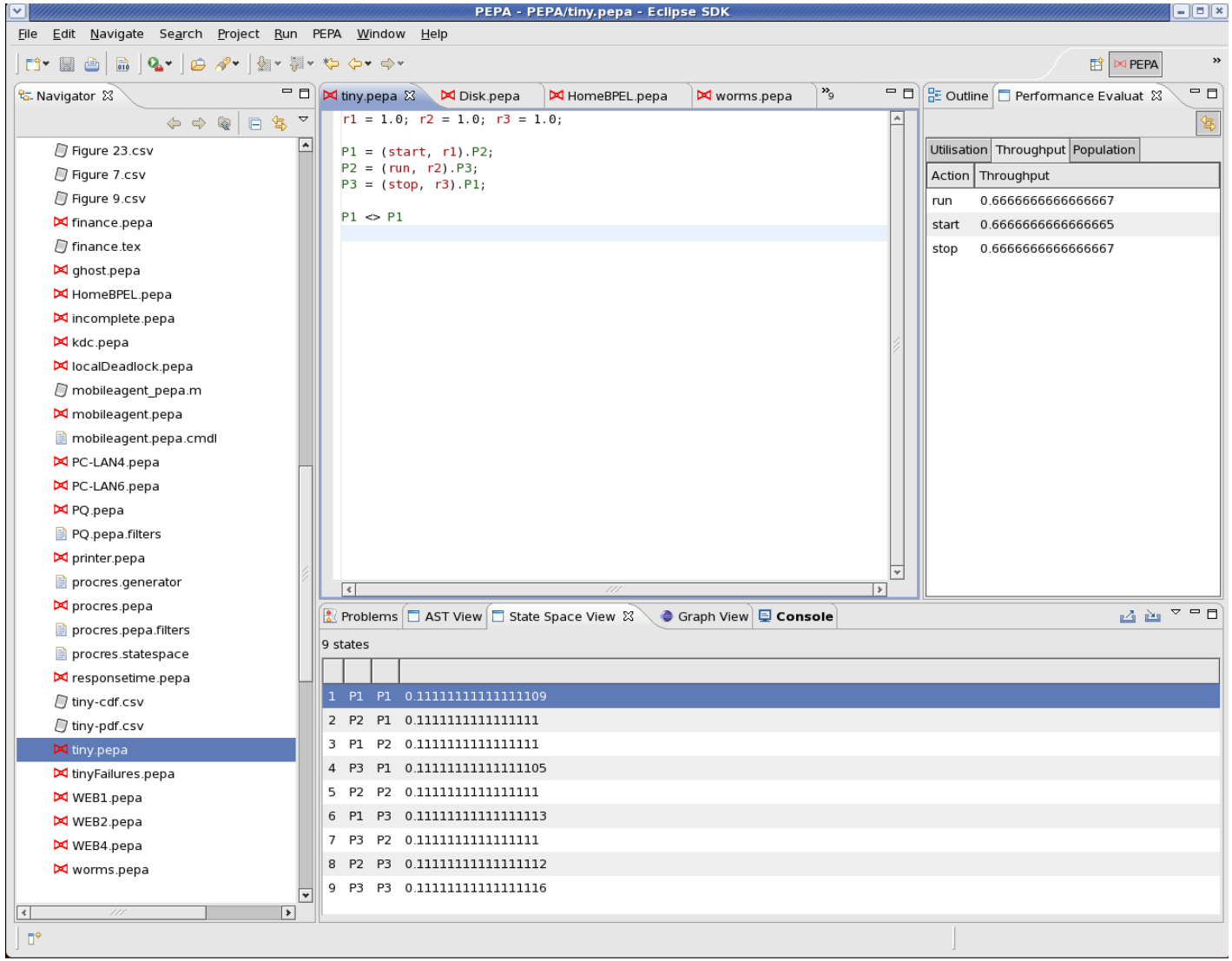
$$P_1 \stackrel{def}{=} (start, r_1).P_2 \qquad P_2 \stackrel{def}{=} (run, r_2).P_3 \qquad P_3 \stackrel{def}{=} (stop, r_3).P_1$$

$$P_1 \parallel P_1$$

CTMC representation computed by the plug-in

$$\begin{pmatrix}
-2r_1 & r_1 & r_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -r_1 - r_2 & 0 & r_2 & r_1 & 0 & 0 & 0 & 0 \\
0 & 0 & -r_1 - r_2 & 0 & r_1 & r_2 & 0 & 0 & 0 \\
r_3 & 0 & 0 & -r_1 - r_3 & 0 & 0 & 0 & r_1 & 0 \\
0 & 0 & 0 & 0 & -2r_2 & 0 & r_2 & r_2 & 0 \\
r_3 & 0 & 0 & 0 & 0 & -r_1 - r_3 & r_1 & 0 & 0 \\
0 & r_3 & 0 & 0 & 0 & 0 & -r_2 - r_3 & 0 & r_2 \\
0 & 0 & r_3 & 0 & 0 & 0 & 0 & -r_2 - r_3 & r_2 \\
0 & 0 & 0 & r_3 & 0 & r_3 & 0 & 0 & -2r_3
\end{pmatrix}$$

The PEPA Eclipse Plug-in processing the model



The PEPA website

`http://www.dcs.ed.ac.uk/pepa`

From the website the PEPA Eclipse Plug-in and some other tools are available for download.

There is also information about people involved in the PEPA project, projects undertaken and a collection of published papers.