Process Algebras for Quantitative Analysis

Jane Hillston. LFCS, University of Edinburgh

28th June 2005

Outline

Introduction

Compositionality: Interaction and Independence

Applications and Acceptance

Conclusions

Outline

Introduction

Compositionality: Interaction and Independence

Applications and Acceptance

Conclusions

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.

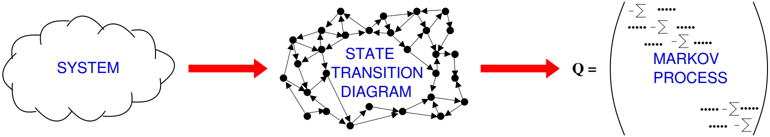
The PEPA project

- 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.

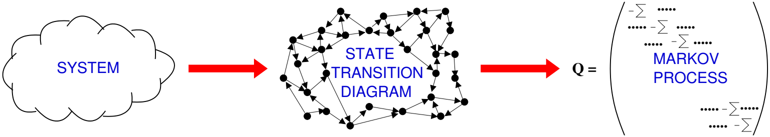
The PEPA project

- 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 Modelling using CTMC



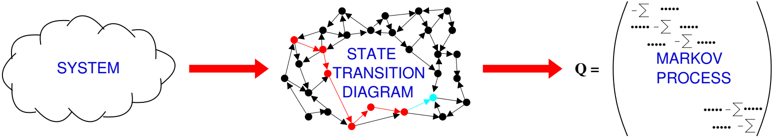
Performance Modelling using CTMC



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

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

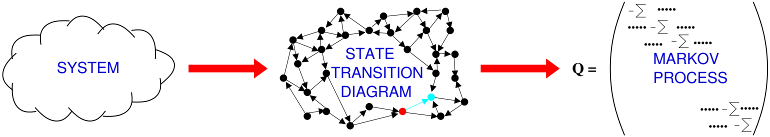
Performance Modelling using CTMC



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

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

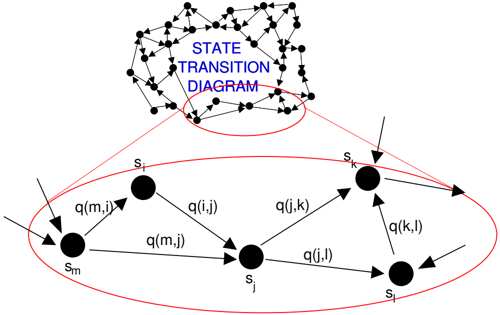
Performance Modelling using CTMC



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

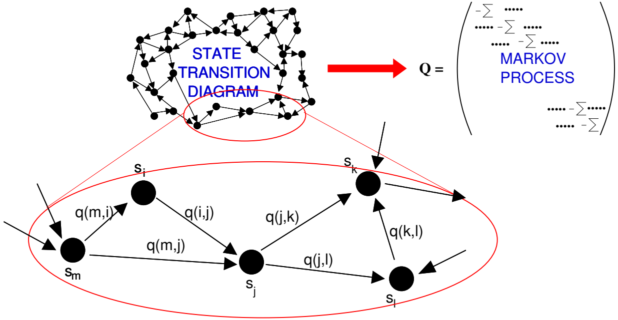
$$\Pr(X(t_{n+1}) = s_{i_{n+1}} \mid X(t_0) = s_{i_0}, \ldots, X(t_n) = s_{i_n}) = \Pr(X(t_{n+1}) = s_{i_{n+1}} \mid X(t_n) = s_{i_n})$$

Performance Modelling using CTMC



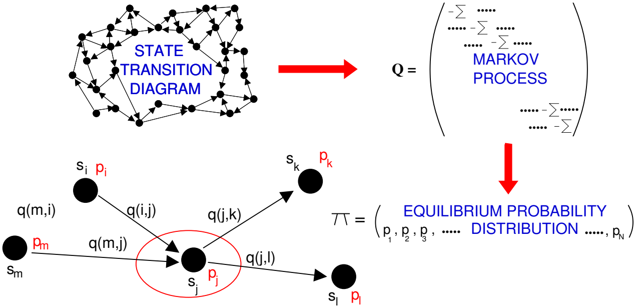
A negative exponentially distributed duration is associated with each transition.

Performance Modelling using CTMC



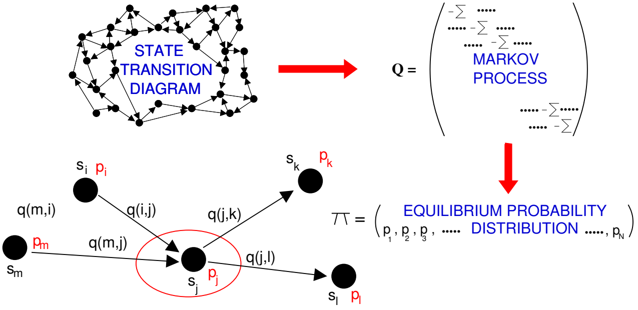
these parameters form the entries of the infinitesimal generator matrix Q

Performance Modelling using CTMC



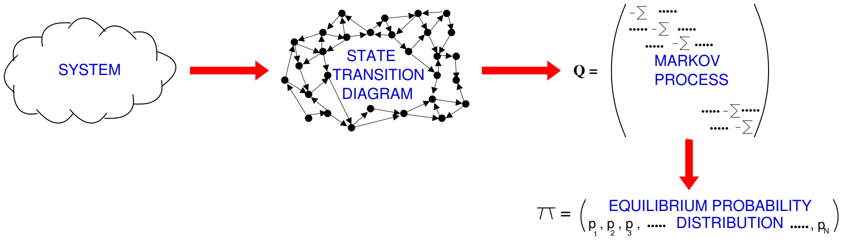
In steady state the probability flux out of a state is balanced by the flux in.

Performance Modelling using CTMC

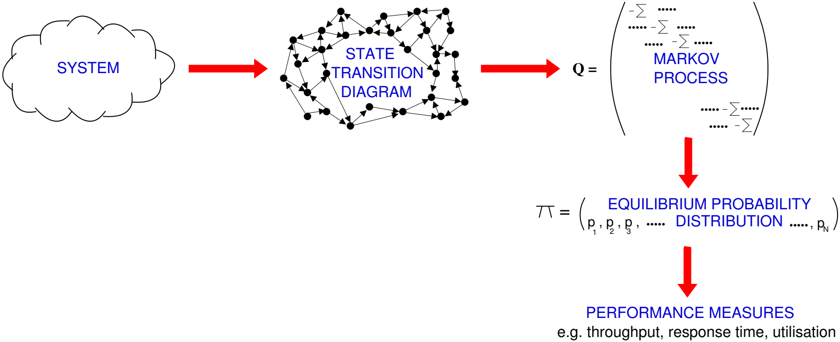


"Global balance equations" captured by $\pi Q = 0$ solved by linear algebra

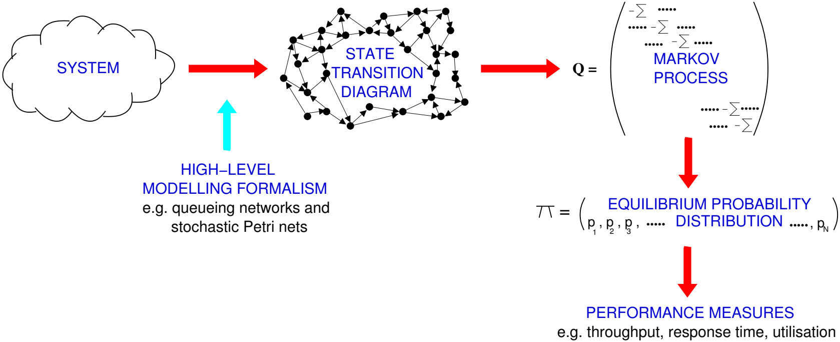
Performance Modelling using CTMC



Performance Modelling using CTMC



Performance Modelling using CTMC



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.

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

Process Algebra

- Models consist of agents which engage in actions.

$$\alpha.P$$

action type or name

agent/ component

- 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 → α; agent/component → P

- 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 → α; agent/ component → P

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

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

Stochastic Process Algebra

From the performance analyst's perspective process algebras had a number of benefits:

Stochastic Process Algebra

From the performance analyst's perspective process algebras had a number of benefits:

Qualitative analysis: Proving the functionality of systems correct.

Stochastic Process Algebra

From the performance analyst's perspective process algebras had a number of benefits:

Qualitative analysis: Proving the functionality of systems correct.

Compositionality: Established benefits for decomposed qualitative analysis as well as model construction for complex systems.

Stochastic Process Algebra

From the performance analyst's perspective process algebras had a number of benefits:

Qualitative analysis: Proving the functionality of systems correct.

Compositionality: Established benefits for decomposed qualitative analysis as well as model construction for complex systems.

Wide acceptance: Process algebras had some acceptance outside academia (e.g. LOTOS).

Stochastic Process Algebra

From the performance analyst's perspective process algebras had a number of benefits:

Qualitative analysis: Proving the functionality of systems correct.
Most systems need to be both correct and timely in their response.

Compositionality: Established benefits for decomposed qualitative analysis as well as model construction for complex systems.

Wide acceptance: Process algebras had some acceptance outside academia (e.g. LOTOS).

Stochastic Process Algebra

From the performance analyst's perspective process algebras had a number of benefits:

Qualitative analysis: Proving the functionality of systems correct. Most systems need to be both correct and timely in their response.

Compositionality: Established benefits for decomposed qualitative analysis as well as model construction for complex systems. Compositionality for quantitative analysis would address the state space explosion problem.

Wide acceptance: Process algebras had some acceptance outside academia (e.g. LOTOS).

Stochastic Process Algebra

From the performance analyst's perspective process algebras had a number of benefits:

Qualitative analysis: Proving the functionality of systems correct. Most systems need to be both correct and timely in their response.

Compositionality: Established benefits for decomposed qualitative analysis as well as model construction for complex systems. Compositionality for quantitative analysis would address the state space explosion problem.

Wide acceptance: Process algebras had some acceptance outside academia (e.g. LOTOS). This offered hope that performance analysis could become better integrated into the software development lifecycle.

SPA Languages

SPA

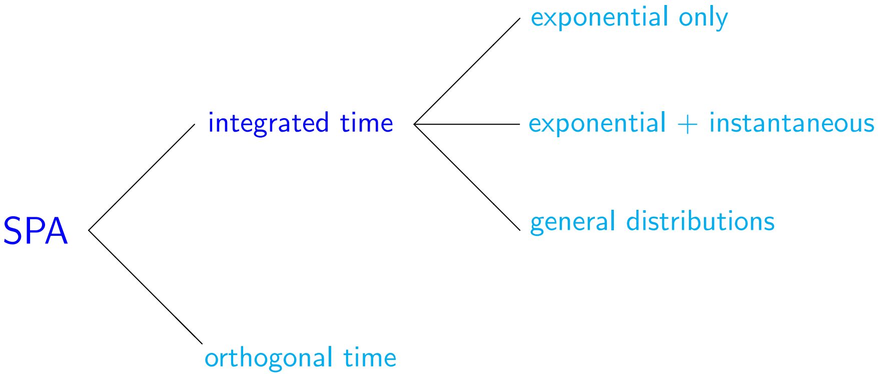
SPA Languages

SPA

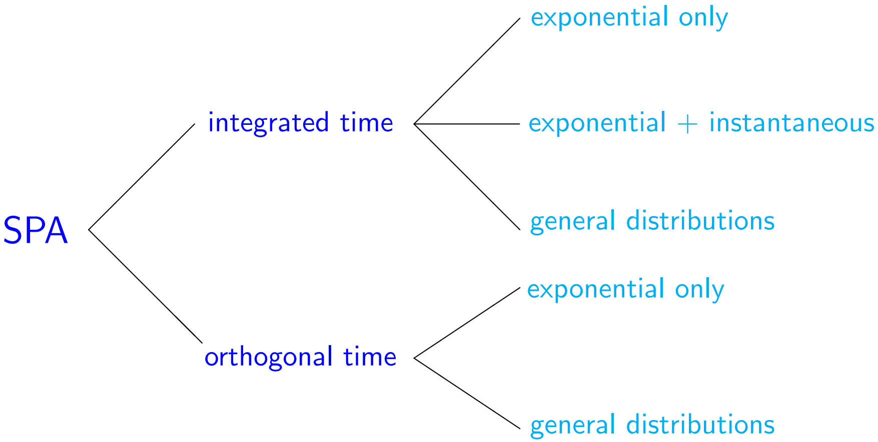
integrated time

orthogonal time

SPA Languages



SPA Languages



SPA Languages

- SPA
  - integrated time
    - exponential only
      PEPA, Stochastic π-calculus
    - exponential + instantaneous
    - general distributions
  - orthogonal time
    - exponential only
    - general distributions

SPA Languages

- SPA
  - integrated time
    - exponential only
      - PEPA, Stochastic π-calculus
    - exponential + instantaneous
      - EMPA, Markovian TIPP
    - general distributions
  - orthogonal time
    - exponential only
    - general distributions

SPA Languages

- SPA
  - integrated time
    - exponential only
      - PEPA, Stochastic π-calculus
    - exponential + instantaneous
      - EMPA, Markovian TIPP
    - general distributions
      - TIPP, SPADES, GSMPA
  - orthogonal time
    - exponential only
    - general distributions

SPA Languages

- SPA
  - integrated time
    - exponential only
      - PEPA, Stochastic π-calculus
    - exponential + instantaneous
      - EMPA, Markovian TIPP
    - general distributions
      - TIPP, SPADES, GSMPA
  - orthogonal time
    - exponential only
      - IMC
    - general distributions

SPA Languages

- SPA
  - integrated time
    - exponential only
      - PEPA, Stochastic π-calculus
    - exponential + instantaneous
      - EMPA, Markovian TIPP
    - general distributions
      - TIPP, SPADES, GSMPA
  - orthogonal time
    - exponential only
      - IMC
    - general distributions
      - IGSMP, Modest

SPA Languages

- SPA
  - integrated time
    - exponential only
      - PEPA, Stochastic π-calculus
    - exponential + instantaneous
      - EMPA, Markovian TIPP
    - general distributions
      - TIPP, SPADES, GSMPA
  - orthogonal time
    - exponential only
      - IMC
    - general distributions
      - IGSMP, Modest

Performance Evaluation 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

Performance Evaluation 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

Performance Evaluation 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

Performance Evaluation 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

Performance Evaluation Process Algebra

- Models are constructed from components which engage in activities.

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

action type or name

activity rate (parameter of an exponential distribution)

component/ derivative

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

Performance Evaluation Process Algebra

- Models are constructed from components which engage in activities.

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

action type or name

activity rate (parameter of an exponential distribution)

component/ derivative

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

PEPA
MODEL

Performance Evaluation Process Algebra

- Models are constructed from components which engage in activities.

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

action type or name

activity rate (parameter of an exponential distribution)

component/ derivative

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

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

Performance Evaluation Process Algebra

- Models are constructed from components which engage in activities.

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

action type or name

activity rate (parameter of an exponential distribution)

component/ derivative

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

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

Performance Evaluation Process Algebra

- Models are constructed from components which engage in activities.

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

action type or name

activity rate (parameter of an exponential distribution)

component/ derivative

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

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

Performance Evaluation Process Algebra

- Models are constructed from components which engage in activities.

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

action type or name

activity rate (parameter of an exponential distribution)

component/ derivative

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

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

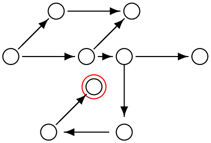
Qualitative Analysis

- The labelled transition system underlying a process algebra model can be used for functional verification e.g.: reachability analysis, specification matching and model checking.

Qualitative Analysis

- The labelled transition system underlying a process algebra model can be used for functional verification e.g.: reachability analysis, specification matching and model checking.

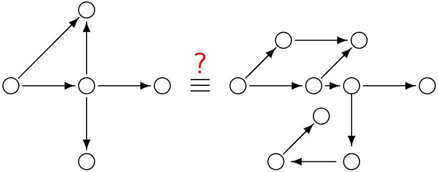
Will the system arrive in a particular state?



Qualitative Analysis

- The labelled transition system underlying a process algebra model can be used for functional verification e.g.: reachability analysis, specification matching and model checking.

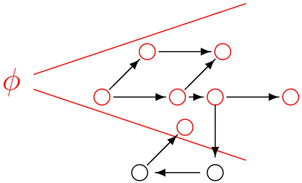
Does system behaviour match its specification?



Qualitative Analysis

- The labelled transition system underlying a process algebra model can be used for functional verification e.g.: reachability analysis, specification matching and model checking.

Does a given property ϕ hold within the system?



Integrated analysis

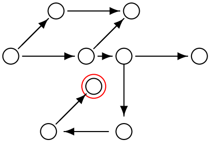
- Qualitative verification can now be complemented by quantitative verification:

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?

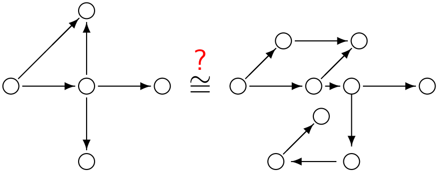


Integrated analysis

- Qualitative verification can now be complemented by quantitative verification:

Specification matching

With what probability does system behaviour match its specification?

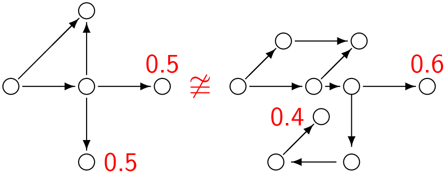


Integrated analysis

- Qualitative verification can now be complemented by quantitative verification:

Specification matching

Does the "*frequency profile*" of the system match that of the specification?

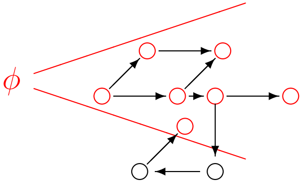


Integrated analysis

- Qualitative verification can now be complemented by quantitative verification:

Model checking

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

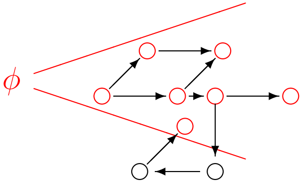


Integrated analysis

- 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?



Outline

Introduction

Compositionality: Interaction and Independence

Applications and Acceptance

Conclusions

Interplay between process algebra and Markov chain

- The theoretical development underpinning PEPA has focused on the interplay between the process algebra and the underlying mathematical structure, the Markov chain.

Interplay between process algebra and Markov chain

- The theoretical development underpinning PEPA has focused on the interplay between the process algebra and the underlying mathematical structure, the Markov chain.
- From the process algebra side the Markov chain had a profound influence on the design of the language and in particular on the interactions between components.

Interplay between process algebra and Markov chain

- The theoretical development underpinning PEPA has focused on the interplay between the process algebra and the underlying mathematical structure, the Markov chain.
- From the process algebra side the Markov chain had a profound influence on the design of the language and in particular on the interactions between components.
- From the Markov chain perspective the process algebra structure has been exploited to find aspects of independence even between interacting components.

PEPA

$$
\begin{array}{rcl}
S & ::= & (\alpha, r).S \mid S + S \mid A \\
P & ::= & S \mid P \underset{L}{\bowtie} P \mid P/L
\end{array}
$$

PEPA

$$
\begin{array}{rcl}
S & ::= & (\alpha, r).S \mid S + S \mid A \\
P & ::= & S \mid P \underset{L}{\bowtie} P \mid P/L
\end{array}
$$

PREFIX: $(\alpha, r).S$ designated first action

PEPA

$$
\begin{array}{rcl}
S & ::= & (\alpha, r).S \mid S + S \mid A \\
P & ::= & S \mid P \underset{L}{\bowtie} P \mid P/L
\end{array}
$$

PREFIX:	$(\alpha, r).S$	designated first action
CHOICE:	$S + S$	competing components (race policy)

PEPA

$$
\begin{array}{rcl}
S & ::= & (\alpha, r).S \mid S + S \mid A \\
P & ::= & S \mid P \underset{L}{\bowtie} P \mid P/L
\end{array}
$$

PREFIX:	$(\alpha, r).S$	designated first action
CHOICE:	$S + S$	competing components (race policy)
CONSTANT:	$A \stackrel{\mathit{def}}{=} S$	assigning names

PEPA

$$
\begin{array}{rcl}
S & ::= & (\alpha, r).S \mid S + S \mid A \\
P & ::= & S \mid P \underset{L}{\bowtie} P \mid P/L
\end{array}
$$

PREFIX:	$(\alpha, r).S$	designated first action
CHOICE:	$S + S$	competing components (race policy)
CONSTANT:	$A \stackrel{\mathit{def}}{=} S$	assigning names
COOPERATION:	$P \underset{L}{\bowtie} P$	$\alpha \notin L$ concurrent activity (*individual actions*) $\alpha \in L$ cooperative activity (*shared actions*)

PEPA

$$\begin{array}{rcl} S & ::= & (\alpha, r).S \mid S + S \mid A \\ P & ::= & S \mid P \underset{L}{\bowtie} P \mid P/L \end{array}$$

PREFIX:	$(\alpha, r).S$	designated first action
CHOICE:	$S + S$	competing components (race policy)
CONSTANT:	$A \stackrel{\mathit{def}}{=} S$	assigning names
COOPERATION:	$P \underset{L}{\bowtie} P$	$\alpha \notin L$ concurrent activity (*individual actions*) $\alpha \in L$ cooperative activity (*shared actions*)
HIDING:	P/L	abstraction $\alpha \in L \Rightarrow \alpha \rightarrow \tau$

The Importance of Being Exponential

$$
\begin{tikzcd}
 & (\alpha, r).Stop \parallel (\beta, s).Stop \arrow[dl, "{(\alpha, r)}"'] \arrow[dr, "{(\beta, s)}"] & \\
Stop \parallel (\beta, s).Stop \arrow[dr, "{(\beta, s)}"'] & & (\alpha, r).Stop \parallel Stop \arrow[dl, "{(\alpha, r)}"] \\
 & Stop \parallel Stop &
\end{tikzcd}
$$

The Importance of Being Exponential

$$
\begin{tikzcd}
 & (\alpha, r).Stop \parallel (\beta, s).Stop \arrow[dl, "{(\alpha, r)}"'] \arrow[dr, "{(\beta, s)}"] & \\
Stop \parallel (\beta, s).Stop \arrow[dr, "{(\beta, s)}"'] & & (\alpha, r).Stop \parallel Stop \arrow[dl, "{(\alpha, r)}"] \\
 & Stop \parallel Stop &
\end{tikzcd}
$$

The Importance of Being Exponential

$$
\begin{tikzcd}
 & (\alpha, r).Stop \parallel (\beta, s).Stop \arrow[dl, "{(\alpha, r)}"'] \arrow[dr, "{(\beta, s)}"] & \\
Stop \parallel (\beta, s).Stop \arrow[dr, "{(\beta, s)}"'] & & (\alpha, r).Stop \parallel Stop \arrow[dl, "{(\alpha, r)}"] \\
 & Stop \parallel Stop &
\end{tikzcd}
$$

The Importance of Being Exponential

$$\begin{tikzcd}
 & (\alpha, r).Stop \parallel (\beta, s).Stop \arrow[dl, "{(\alpha, r)}"'] \arrow[dr, "{(\beta, s)}"] \arrow[dd, "{(\alpha, r) \mid (\beta, s)}"] & \\
Stop \parallel (\beta, s).Stop \arrow[dr, "{(\beta, s)}"'] & & (\alpha, r).Stop \parallel Stop \arrow[dl, "{(\alpha, r)}"] \\
 & Stop \parallel Stop &
\end{tikzcd}$$

The Importance of Being Exponential

$$
\begin{tikzcd}
& (\alpha, r).Stop \parallel (\beta, s).Stop \arrow[dl, "{(\alpha, r)}"'] \arrow[dr, "{(\beta, s)}"] & \\
Stop \parallel (\beta, s).Stop \arrow[dr, "{(\beta, s)}"'] & & (\alpha, r).Stop \parallel Stop \arrow[dl, "{(\alpha, r)}"] \\
& Stop \parallel Stop &
\end{tikzcd}
$$

The Importance of Being Exponential

$$\begin{tikzcd}
 & (\alpha, r).Stop \parallel (\beta, s).Stop \arrow[dl, "{(\alpha, r)}"'] \arrow[dr, "{(\beta, s)}"] & \\
Stop \parallel (\beta, s).Stop \arrow[dr, "{(\beta, s)}"'] & & (\alpha, r).Stop \parallel Stop \arrow[dl, "{(\alpha, r)}"] \\
 & Stop \parallel Stop &
\end{tikzcd}$$

The Importance of Being Exponential

$$
\begin{tikzcd}
& (\alpha, r).Stop \parallel (\beta, s).Stop \arrow[dl, "(\alpha, r)"'] \arrow[dr, "(\beta, s)"] & \\
Stop \parallel (\beta, s).Stop \arrow[dr, "(\beta, s)"'] & & (\alpha, r).Stop \parallel Stop \arrow[dl, "(\alpha, r)"] \\
& Stop \parallel Stop &
\end{tikzcd}
$$

The Importance of Being Exponential

$$
\begin{tikzcd}
 & (\alpha, r).Stop \parallel (\beta, s).Stop \arrow[dl, "{(\alpha, r)}"'] \arrow[dr, "{(\beta, s)}"] & \\
Stop \parallel (\beta, s).Stop \arrow[dr, "{(\beta, s)}"'] & & (\alpha, r).Stop \parallel Stop \arrow[dl, "{(\alpha, r)}"] \\
 & Stop \parallel Stop &
\end{tikzcd}
$$

The memoryless property of the negative exponential distribution means that residual times do not need to be recorded.

The Importance of Being Exponential

$$
\begin{tikzcd}
& (\alpha, r).Stop \parallel (\beta, s).Stop \arrow[dl, "(\alpha, r)"'] \arrow[dr, "(\beta, s)"] & \\
Stop \parallel (\beta, s).Stop \arrow[dr, "(\beta, s)"'] & & (\alpha, r).Stop \parallel Stop \arrow[dl, "(\alpha, r)"] \\
& Stop \parallel Stop &
\end{tikzcd}
$$

We retain the expansion law of classical process algebra:

$$
\begin{aligned}
&(\alpha, r).Stop \parallel (\beta, s).Stop = \\
&\qquad (\alpha, r).(\beta, s).(Stop \parallel Stop) + (\beta, s).(\alpha, r).(Stop \parallel Stop)
\end{aligned}
$$

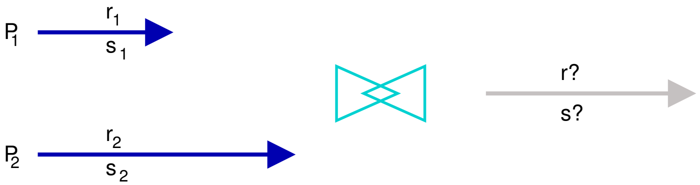
only if the negative exponential distribution is assumed.

Timed synchronisation

- The issue of what it means for two timed activities to synchronise is a vexed one....

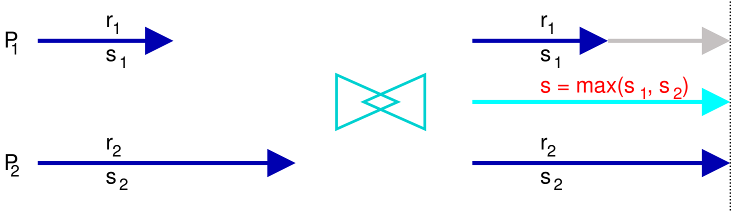
Timed synchronisation

- The issue of what it means for two timed activities to synchronise is a vexed one....



Timed synchronisation

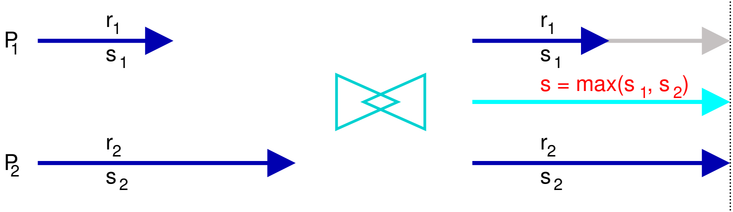
- The issue of what it means for two timed activities to synchronise is a vexed one....



Barrier Synchronisation

Timed synchronisation

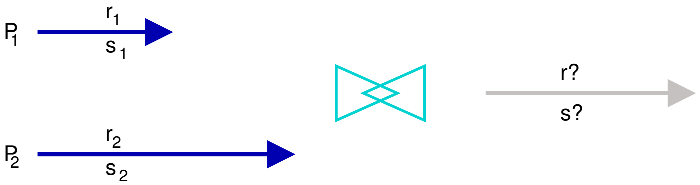
- The issue of what it means for two timed activities to synchronise is a vexed one....



s is no longer exponentially distributed

Timed synchronisation

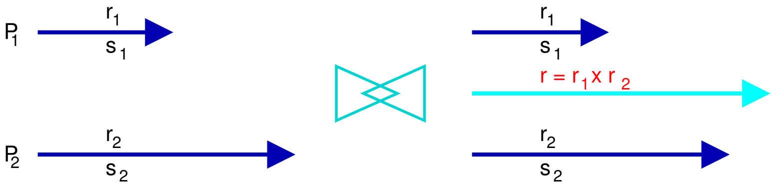
- The issue of what it means for two timed activities to synchronise is a vexed one....



algebraic considerations limit choices

Timed synchronisation

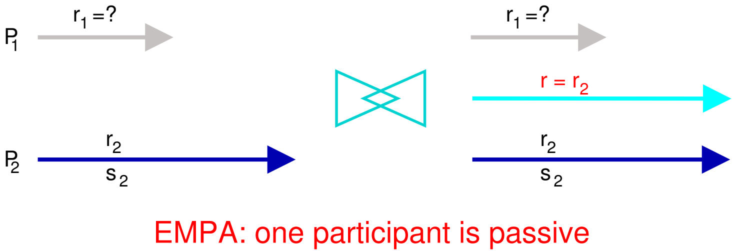
- The issue of what it means for two timed activities to synchronise is a vexed one....



TIPP: new rate is product of individual rates

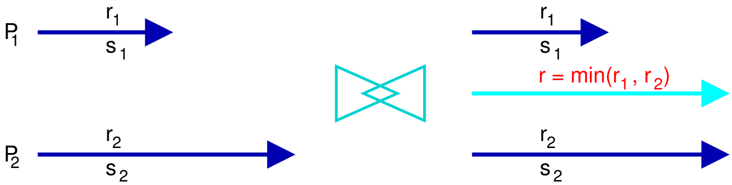
Timed synchronisation

- The issue of what it means for two timed activities to synchronise is a vexed one....



Timed synchronisation

- The issue of what it means for two timed activities to synchronise is a vexed one....



bounded capacity: new rate is the minimum of the rates

Cooperation in PEPA

- In PEPA each component has a bounded capacity to carry out activities of any particular type, determined by the apparent rate for that type.

Cooperation in PEPA

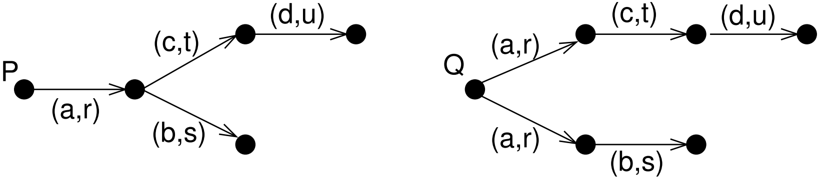
- In PEPA each component has a bounded capacity to carry out activities of any particular type, determined by the apparent rate for that type.
- Synchronisation, or cooperation cannot make a component exceed its bounded capacity.

Cooperation in PEPA

- In PEPA each component has a bounded capacity to carry out activities of any particular type, determined by the apparent rate for that type.
- Synchronisation, or cooperation cannot make a component exceed its bounded capacity.
- Thus the apparent rate of a cooperation is the minimum of the apparent rates of the co-operands.

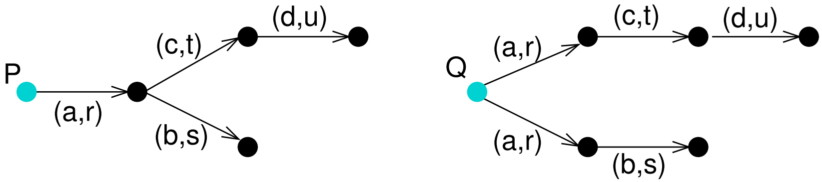
Equivalence Relations

In process algebra equivalence relations are defined based on the notion of observability:



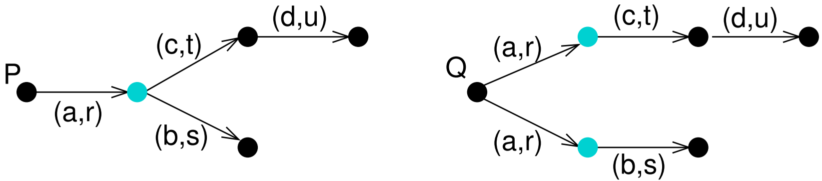
Equivalence Relations

In process algebra equivalence relations are defined based on the notion of observability:



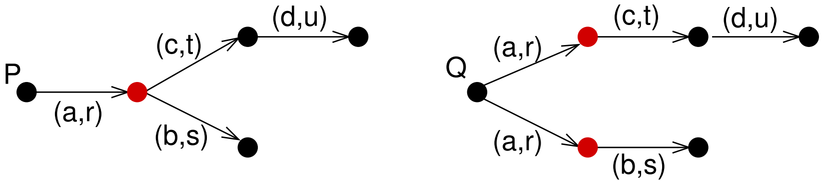
Equivalence Relations

In process algebra equivalence relations are defined based on the notion of observability:



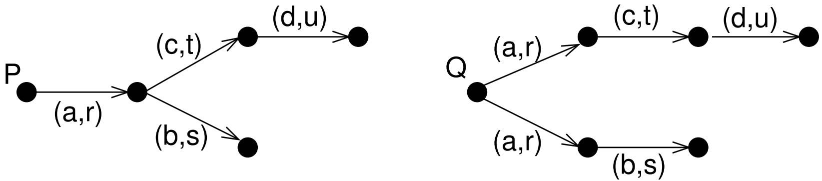
Equivalence Relations

In process algebra equivalence relations are defined based on the notion of observability:



Equivalence Relations

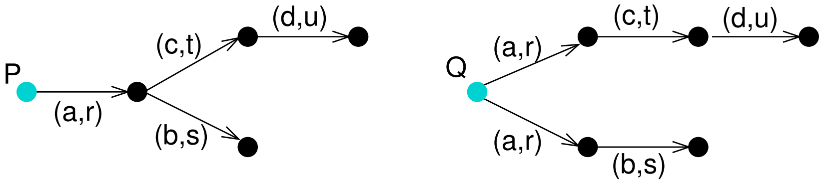
In process algebra equivalence relations are defined based on the notion of observability:



In PEPA observation is assumed to include the ability to record timing information over a number of runs.

Equivalence Relations

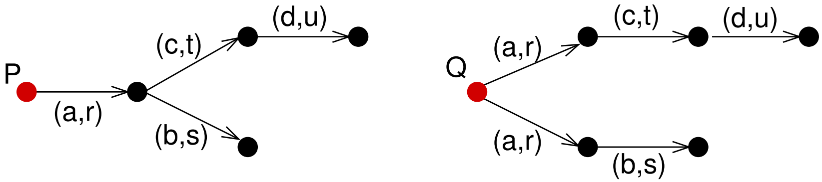
In process algebra equivalence relations are defined based on the notion of observability:



In PEPA observation is assumed to include the ability to record timing information over a number of runs.

Equivalence Relations

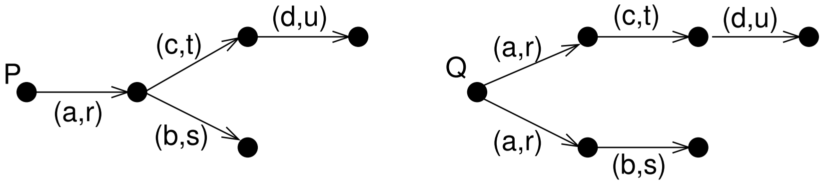
In process algebra equivalence relations are defined based on the notion of observability:



In PEPA observation is assumed to include the ability to record timing information over a number of runs.

Equivalence Relations

In process algebra equivalence relations are defined based on the notion of observability:



In PEPA observation is assumed to include the ability to record timing information over a number of runs.

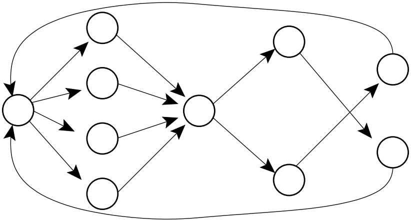
The resulting equivalence relation is a bisimulation in the style of Larsen and Skou, and coincides with the Markov process notion of lumpability.

Aggregation and lumpability

Model aggregation: use a state-state equivalence to establish a partition of the state space of a model, and replace each set of states by one macro-state, i.e. take a different stochastic representation of the same model.

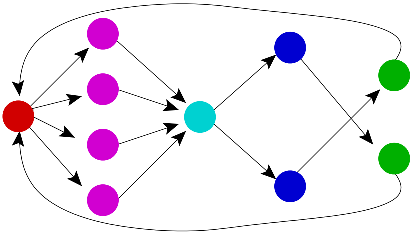
Aggregation and lumpability

Model aggregation: use a state-state equivalence to establish a partition of the state space of a model, and replace each set of states by one macro-state, i.e. take a different stochastic representation of the same model.



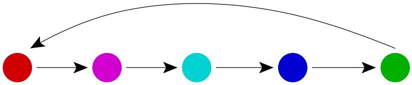
Aggregation and lumpability

Model aggregation: use a state-state equivalence to establish a partition of the state space of a model, and replace each set of states by one macro-state, i.e. take a different stochastic representation of the same model.



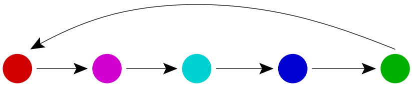
Aggregation and lumpability

Model aggregation: use a state-state equivalence to establish a partition of the state space of a model, and replace each set of states by one macro-state, i.e. take a different stochastic representation of the same model.



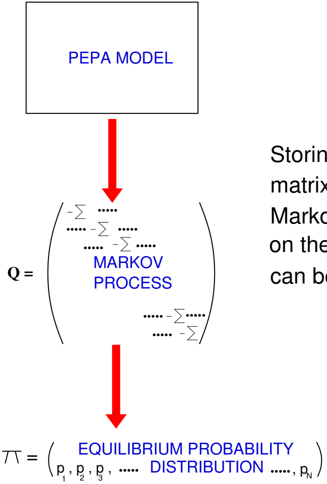
Aggregation and lumpability

Model aggregation: use a state-state equivalence to establish a partition of the state space of a model, and replace each set of states by one macro-state, i.e. take a different stochastic representation of the same model.



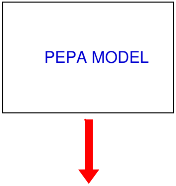
A lumpable partition is the only partition of a Markov process which preserves the Markov property.

Characterising efficient solution

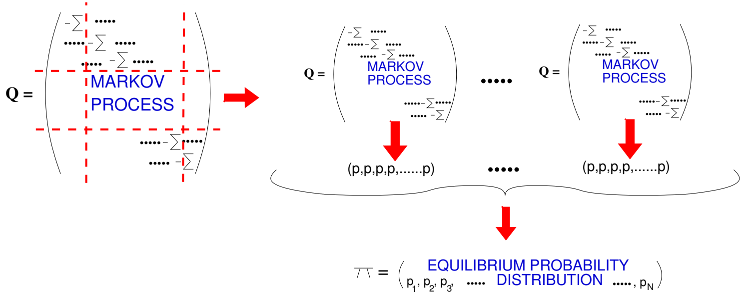


Storing and manipulating the matrix which represents the Markov process places limitations on the size of model which can be analysed.

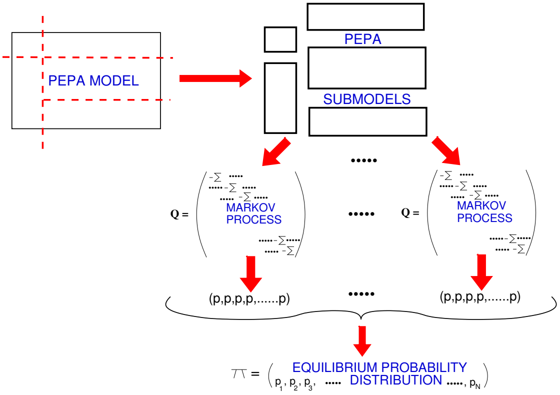
Characterising efficient solution



Certain structures in the matrix are known to be amenable to efficient, decomposed solution.

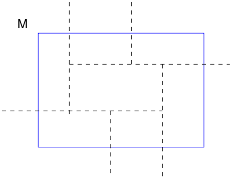


Characterising efficient solution

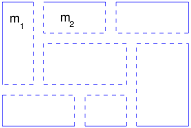


Finding the corresponding structures in the process algebra means that these techniques can be applied automatically, before the monolithic matrix is formed.

Decomposed solution: product form models



$M = (m_1, m_2, \ldots, m_n)$



p(M)

$p(M) = G \times p(m_1) \times p(m_2) \times \ldots \times p(m_n)$

Partition the model M into n statistically independent submodels $m_1, m_2, \ldots, m_n$

In isolation, find the steady state distribution p for each of the submodels m_i

Form the steady state distribution of M as the product of the solutions for each submodel m_i and a normalising constant

When do PEPA components behave as if they were statistically independent...?

Product Form PEPA Models

$$P \equiv S_1 \parallel S_2$$

Product Form PEPA Models

$$P \equiv S_1 \parallel S_2$$

Add restricted direct interaction between components with a particular structure

$$P \equiv S_1 \underset{L}{\bowtie} S_2$$

S_1, S_2 and L all restricted

Product Form PEPA Models

$$P \equiv S_1 \parallel S_2$$

Add restricted direct interaction between components with a particular structure

$$P \equiv S_1 \underset{L}{\bowtie} S_2$$

S_1, S_2 and L all restricted

- Quasi-reversibility

Product Form PEPA Models

$$P \equiv S_1 \parallel S_2$$

Add restricted direct interaction between components with a particular structure

$$P \equiv S_1 \underset{L}{\bowtie} S_2$$

S_1, S_2 and L all restricted

- Quasi-reversibility
- Reversibility

Product Form PEPA Models

$$P \equiv S_1 \parallel S_2$$

Add restricted direct interaction between components with a particular structure

$$P \equiv S_1 \underset{L}{\bowtie} S_2$$

S_1, S_2 and L all restricted

- Quasi-reversibility
- Reversibility
- Routing process approach

Product Form PEPA Models

$$P \equiv S_1 \parallel S_2$$

Add restricted direct interaction between components with a particular structure

$$P \equiv S_1 \underset{L}{\bowtie} S_2$$

S_1, S_2 and L all restricted

- Quasi-reversibility
- Reversibility
- Routing process approach

Add indirect interaction via a third component with a particular structure and type of interaction

$$P \equiv (S_1 \parallel S_2) \underset{L}{\bowtie} R$$

L and R restricted (wrt S_1 and S_2)

Product Form PEPA Models

$$P \equiv S_1 \parallel S_2$$

Add restricted direct interaction between components with a particular structure

$$P \equiv S_1 \underset{L}{\bowtie} S_2$$

S_1, S_2 and L all restricted

- Quasi-reversibility
- Reversibility
- Routing process approach

Add indirect interaction via a third component with a particular structure and type of interaction

$$P \equiv (S_1 \parallel S_2) \underset{L}{\bowtie} R$$

L and R restricted (wrt S_1 and S_2)

- Boucherie resource contention

Product Form PEPA Models

$$P \equiv S_1 \parallel S_2$$

Add restricted direct interaction between components with a particular structure

$$P \equiv S_1 \underset{L}{\bowtie} S_2$$

S_1, S_2 and L all restricted

- Quasi-reversibility
- Reversibility
- Routing process approach

Add indirect interaction via a third component with a particular structure and type of interaction

$$P \equiv (S_1 \parallel S_2) \underset{L}{\bowtie} R$$

L and R restricted (wrt S_1 and S_2)

- Boucherie resource contention
- Queueing discipline models

Product Form PEPA Models

$$P \equiv S_1 \parallel S_2$$

Add restricted direct interaction between components with a particular structure

$$P \equiv S_1 \underset{L}{\bowtie} S_2$$

S_1, S_2 and L all restricted

- Quasi-reversibility
- Reversibility
- Routing process approach

Add indirect interaction via a third component with a particular structure and type of interaction

$$P \equiv (S_1 \parallel S_2) \underset{L}{\bowtie} R$$

L and R restricted (wrt S_1 and S_2)

- Boucherie resource contention
- Queueing discipline models
- Quasi-separability

Product Form PEPA Models

$$P \equiv S_1 \parallel S_2$$

Add restricted direct interaction between components with a particular structure

$$P \equiv S_1 \underset{L}{\bowtie} S_2$$

S_1, S_2 and L all restricted

- Quasi-reversibility
- Reversibility
- Routing process approach

Add indirect interaction via a third component with a particular structure and type of interaction

$$P \equiv (S_1 \parallel S_2) \underset{L}{\bowtie} R$$

L and R restricted (wrt S_1 and S_2)

- Boucherie resource contention
- Queueing discipline models
- Quasi-separability

Outline

Introduction

Compositionality: Interaction and Independence

Applications and Acceptance

Conclusions

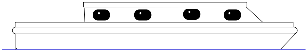
Applications and Acceptance

- Developing models of real applications has always been an integral part of the PEPA project.
- This allows us to demonstrate to ourselves and others that the theory we have developed is useful.
- It serves to promote the acceptance of both stochastic process algebra itself and performance modelling more generally.
- It is also a valuable source of inspiration for new theory and future directions.

PEPA Case Studies (1)

- Multiprocessor access-contention protocols (Gilmore, Hillston and Ribaudo, Edinburgh and Turin)
- Protocols for fault-tolerant systems (Clark, Gilmore, Hillston and Ribaudo, Edinburgh and Turin)
- Multimedia traffic characteristics (Bowman et al, Kent)
- Database systems (The STEADY group, Heriot-Watt University)
- Software Architectures (Pooley, Bradley and Thomas, Heriot-Watt and Durham)
- Switch behaviour in active networks (Hillston, Kloul and Mokhtari, Edinburgh and Versailles)

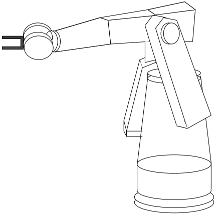
PEPA Case Studies (2)



- Locks and movable bridges in inland shipping in Belgium (Knapen, Hasselt)

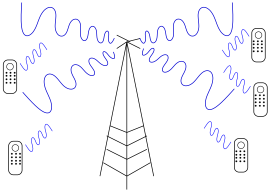
PEPA Case Studies (2)

- Locks and movable bridges in inland shipping in Belgium (Knapen, Hasselt)
- Robotic workcells (Holton, Gilmore and Hillston, Bradford and Edinburgh)



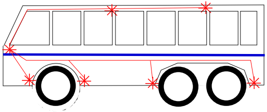
PEPA Case Studies (2)

- Locks and movable bridges in inland shipping in Belgium (Knapen, Hasselt)
- Robotic workcells (Holton, Gilmore and Hillston, Bradford and Edinburgh)
- Cellular telephone networks (Kloul, Fourneau and Valois, Versailles)



PEPA Case Studies (2)

- Locks and movable bridges in inland shipping in Belgium (Knapen, Hasselt)
- Robotic workcells (Holton, Gilmore and Hillston, Bradford and Edinburgh)
- Cellular telephone networks (Kloul, Fourneau and Valois, Versailles)
- Automotive diagnostic expert systems (Console, Picardi and Ribaudo, Turin)



Tool Support

PEPA
Workbench
(Edinburgh University)

Imperial PEPA
Compiler/Dnamaca
and Hydra
(Imperial College)

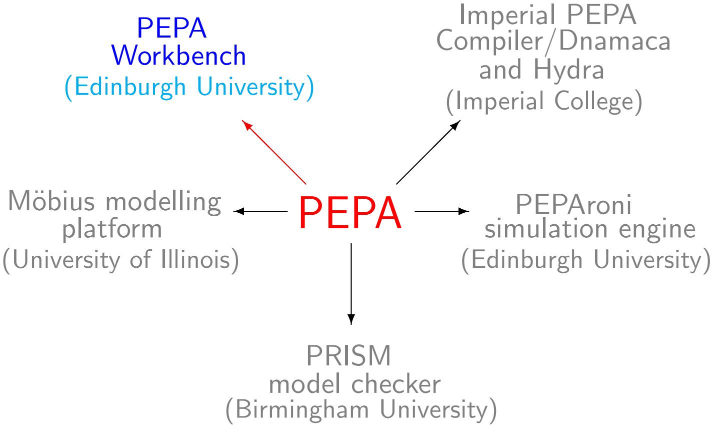
Möbius modelling
platform
(University of Illinois)

PEPA

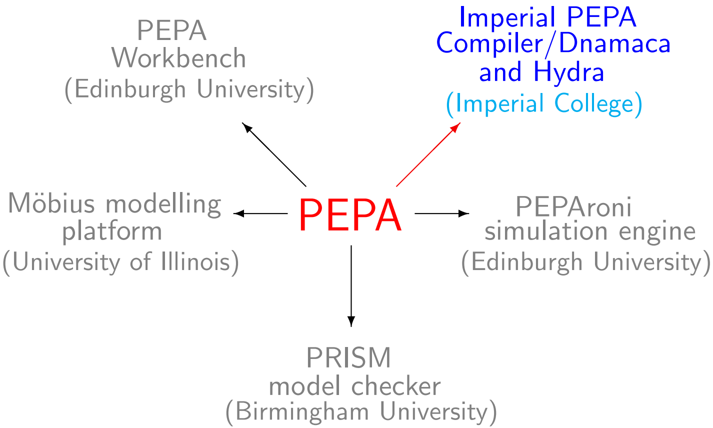
PEPAroni
simulation engine
(Edinburgh University)

PRISM
model checker
(Birmingham University)

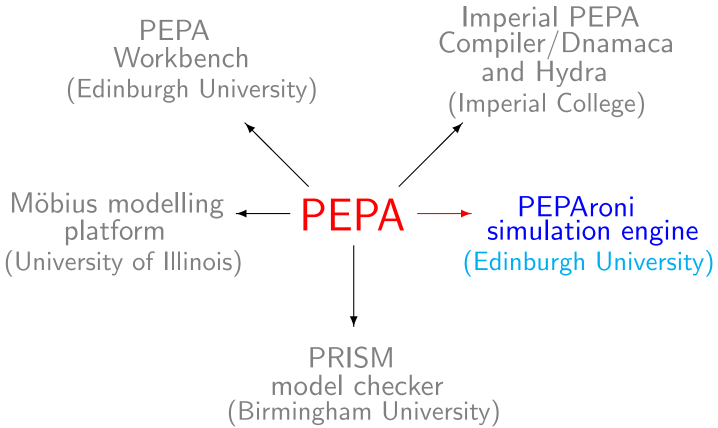
Tool Support



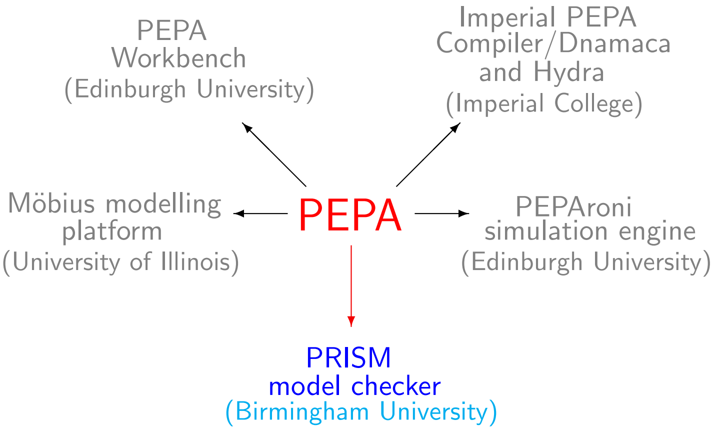
Tool Support



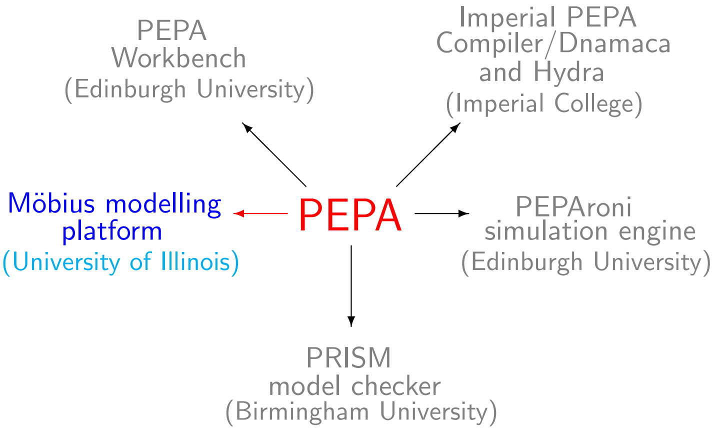
Tool Support



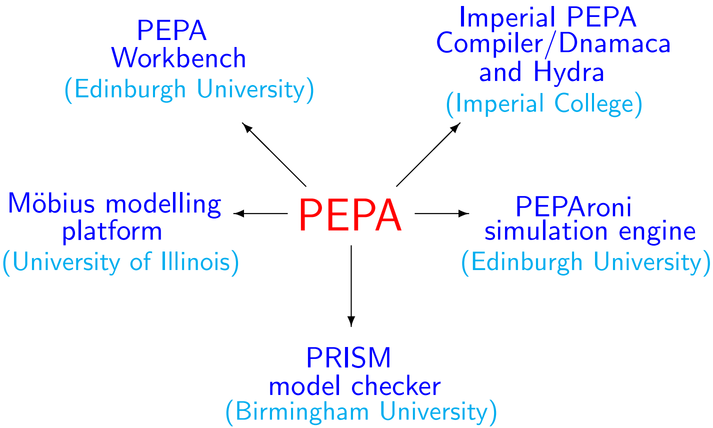
Tool Support



Tool Support



Tool Support



Wider Acceptance

- Stochastic process algebras have not been widely adopted within industry.
- Real integration into industrial processes will only be achieved via notations and methods already used within industry

Wider Acceptance

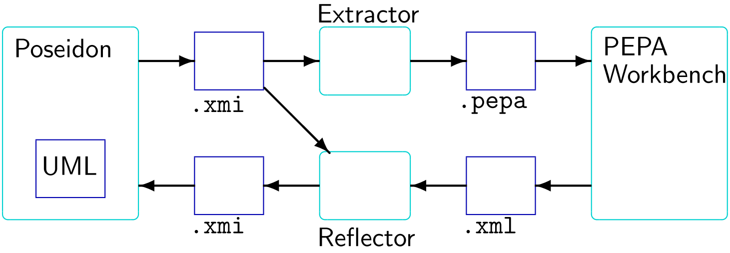
- Stochastic process algebras have not been widely adopted within industry.
- Real integration into industrial processes will only be achieved via notations and methods already used within industry
- ...UML...

Wider Acceptance

- Stochastic process algebras have not been widely adopted within industry.
- Real integration into industrial processes will only be achieved via notations and methods already used within industry
- ...UML...
- Thus recent work has explored this route.

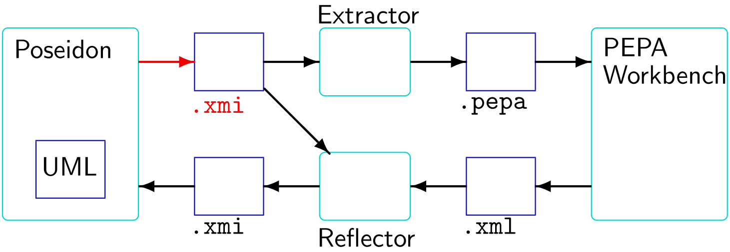
PEPA via UML

In the European-funded DEGAS research project we have been investigating ways to make performance modelling using PEPA more accessible to software designers.



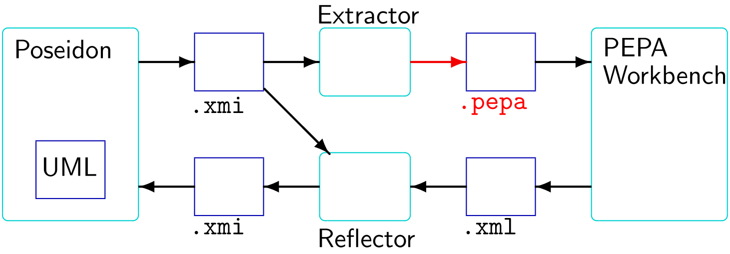
PEPA via UML

In the European-funded DEGAS research project we have been investigating ways to make performance modelling using PEPA more accessible to software designers.



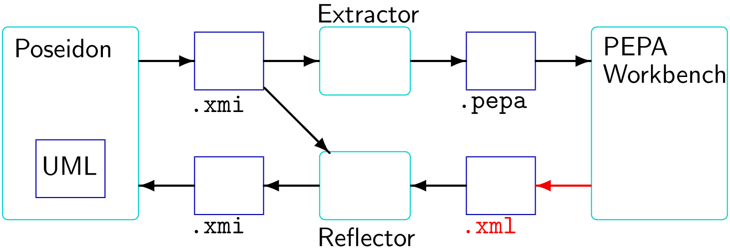
PEPA via UML

In the European-funded DEGAS research project we have been investigating ways to make performance modelling using PEPA more accessible to software designers.



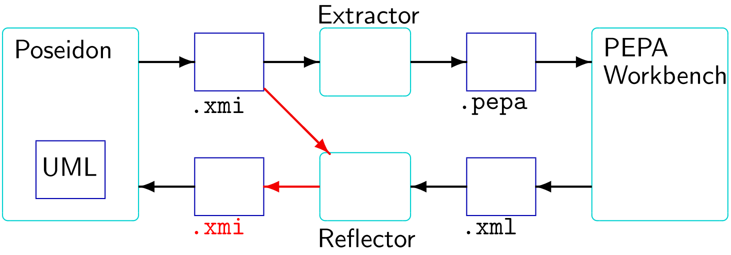
PEPA via UML

In the European-funded DEGAS research project we have been investigating ways to make performance modelling using PEPA more accessible to software designers.



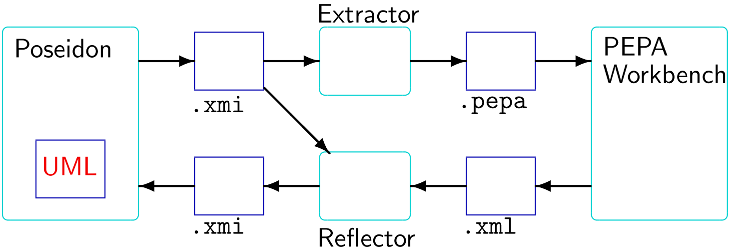
PEPA via UML

In the European-funded DEGAS research project we have been investigating ways to make performance modelling using PEPA more accessible to software designers.



PEPA via UML

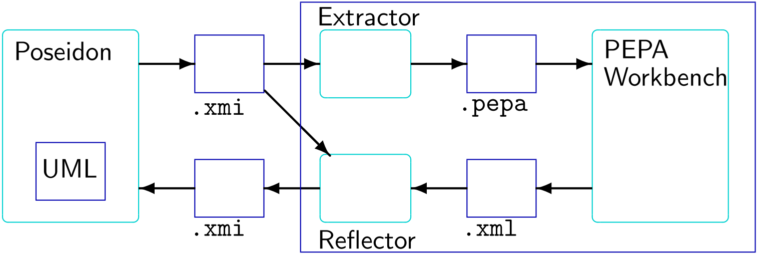
In the European-funded DEGAS research project we have been investigating ways to make performance modelling using PEPA more accessible to software designers.



It is essential that results are reported in terms which make sense to the software designer, i.e. in terms of the original UML model.

PEPA via UML

In the European-funded DEGAS research project we have been investigating ways to make performance modelling using PEPA more accessible to software designers.



Outline

Introduction

Compositionality: Interaction and Independence

Applications and Acceptance

Conclusions

Initial Objectives

Initial Objectives

Qualitative analysis: Only fully realised with the definition of appropriate complementary logics, this objective has been perhaps the most successful.

Initial Objectives

Qualitative analysis: Only fully realised with the definition of appropriate complementary logics, this objective has been perhaps the most successful.

Compositionality: In addition to the clear benefits for model construction, it has been established that compositionality can be exploited during Markovian analysis. There is more to do here e.g. with respect to model checking.

Initial Objectives

Qualitative analysis: Only fully realised with the definition of appropriate complementary logics, this objective has been perhaps the most successful.

Compositionality: In addition to the clear benefits for model construction, it has been established that compositionality can be exploited during Markovian analysis. There is more to do here e.g. with respect to model checking.

Wide acceptance: Initial hopes were perhaps naïve, but SPA is now playing a part in encouraging the wider adoption of performance analysis in software analysis.

Future Work

Many possibilities, for example:

- The state space explosion problem still remains a major challenge.
- Extending the range of applicability of the modelling language for new application areas.
- Improving the analysis capabilities of the modelling tools.

New mathematical structures

For a generation, performance modellers have seen their choices as being:

- Closed form analytical models;

New mathematical structures

For a generation, performance modellers have seen their choices as being:

- Closed form analytical models;
- Simulations; or

New mathematical structures

For a generation, performance modellers have seen their choices as being:

- Closed form analytical models;
- Simulations; or
- Numerical solution of continuous time Markov chains (CTMC)

New mathematical structures

For a generation, performance modellers have seen their choices as being:

- Closed form analytical models;
- Simulations; or
- Numerical solution of continuous time Markov chains (CTMC)

The major limitations of the CTMC approach are the state space explosion problem and the reliance on exponential distributions.

New mathematical structures: differential equations

In a PEPA model the state at any current time is the local derivative or state of each component of the model. When we have large numbers of repeated components it can make sense to represent each component type as a continuous variable, and the state of the model as a whole as the set of such variables. The evolution of each such variable can then be described by an ODE.

New mathematical structures: differential equations

The PEPA definitions of the component specify the activities which can increase or decrease the number of components exhibited in the current state. The cooperations show when the number of instances of another component will have an influence on the evolution of this component.

New mathematical structures: differential equations

- Use a more abstract state representation rather than the CTMC complete state space.

New mathematical structures: differential equations

- Use a more abstract state representation rather than the CTMC complete state space.
- No longer aim to calculate the probability distribution over the entire state space of the model.

New mathematical structures: differential equations

- Use a more abstract state representation rather than the CTMC complete state space.
- No longer aim to calculate the probability distribution over the entire state space of the model.
- Assume that these state variables are subject to continuous rather than discrete change.

New mathematical structures: differential equations

- Use a more abstract state representation rather than the CTMC complete state space.
- No longer aim to calculate the probability distribution over the entire state space of the model.
- Assume that these state variables are subject to continuous rather than discrete change.

Only appropriate for some models, but results are promising in those cases.

New application domains: biochemical signalling pathways

- Biological advances mean that much more is now known about the components of cells and the interactions between them.

New application domains: biochemical signalling pathways

- Biological advances mean that much more is now known about the components of cells and the interactions between them.
- Systems biology aims to develop a better understanding of the processes involved.

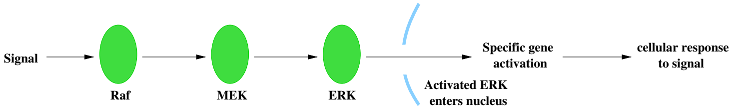
New application domains: biochemical signalling pathways

- Biological advances mean that much more is now known about the components of cells and the interactions between them.
- Systems biology aims to develop a better understanding of the processes involved.
- Stochastic process algebras have found a new role in developing models for systems biology, allowing biologists to test hypotheses and prioritise experiments.

Extracellular signalling

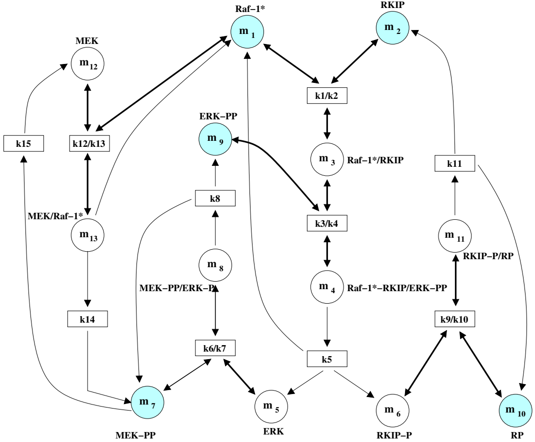
Extracellular signalling — communication between cells.

- signalling molecules released by one cell migrate to another;
- these molecules enter the cell and instigate a pathway, or series of reactions, which carries the information from the membrane to the nucleus;
- the Ras/Raf-1/MEK/ERK pathway conveys differentiation signals to the nucleus of a cell.



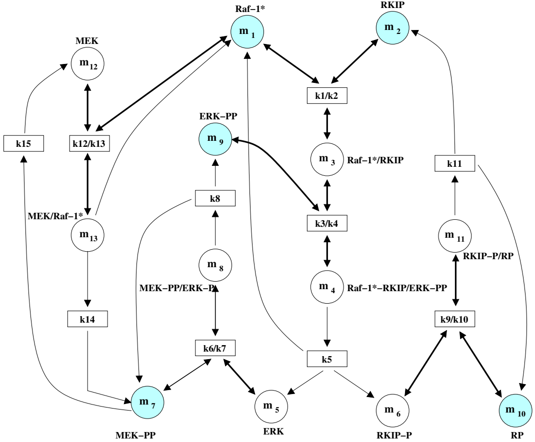
Special relevance to cancer research because when pathways operate abnormally cells divide uncontrollably.

The Ras/Raf-1/MEK/ERK pathway



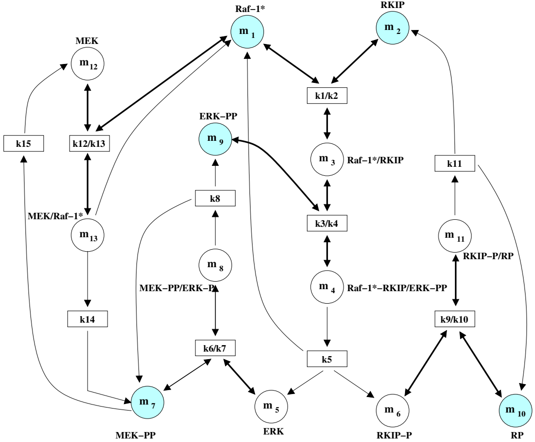
We have constructed two, complementary, PEPA models of the pathway.

The Ras/Raf-1/MEK/ERK pathway



We have constructed two, complementary, PEPA models of the pathway.

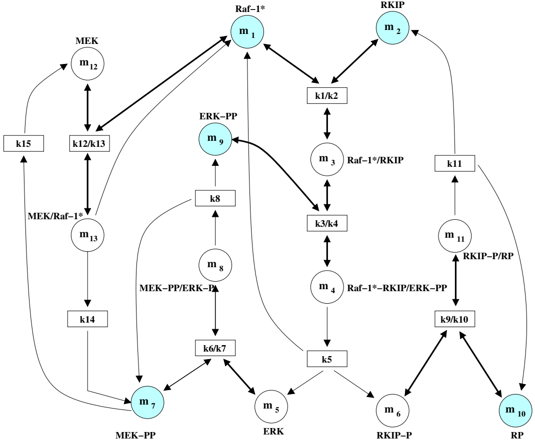
The Ras/Raf-1/MEK/ERK pathway



We have constructed two, complementary, PEPA models of the pathway.

- Reagents-centric

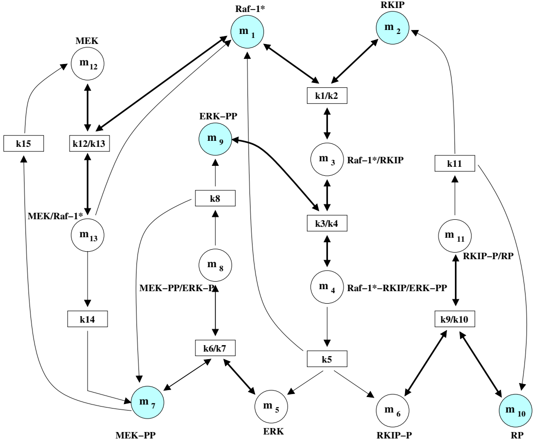
The Ras/Raf-1/MEK/ERK pathway



We have constructed two, complementary, PEPA models of the pathway.

- Reagents-centric
- Pathway-centric

The Ras/Raf-1/MEK/ERK pathway



We have constructed two, complementary, PEPA models of the pathway.

- Reagents-centric
- Pathway-centric

and shown them to be equivalent.

Acknowledgements

The PEPA project has been funded by SERC, EPSRC and the CEC. Over the years many talented people have also contributed to the project in a variety of ways.

Acknowledgements

The PEPA project has been funded by SERC, EPSRC and the CEC. Over the years many talented people have also contributed to the project in a variety of ways.

Coauthors:

- Anne Benoit, Linda Brodo, Muffy Calder, Catherine Canevet, Murray Cole, Graham Clark, Pierpaolo Degano, Susanna Donatelli, Stephen Gilmore, Zully Grant-Duff, Valentin Haenel, Peter Harrison, Robert Holton, Leïla Kloul, Vassilis Mertsiotakis, Amdjed Mokhtari, Corrado Priami, Matthew Prowse, Laura Recalde, Michael Rettelbach, Marina Ribaudo, Manuel Silva, Perdita Stevens, Jennifer Tenzer, Joanna Tomasik, Nigel Thomas.

Acknowledgements

The PEPA project has been funded by SERC, EPSRC and the CEC. Over the years many talented people have also contributed to the project in a variety of ways.

Tool developers:

- Ashok Argent-Katwala, Jeremy Bradley, Graham Clark, Nick Dingle, Ross Duncan, Stephen Gilmore, Valentin Haenel, Jon Hunter, Will Knottenbelt, Dave Parker, Matthew Prowse, Fotis Stathopoulos, Feng Wan.

Acknowledgements

The PEPA project has been funded by SERC, EPSRC and the CEC.
Over the years many talented people have also contributed to the project in a variety of ways.
Other work on PEPA:

- Ashok Argent-Katwala, Howard Bowman, Jeremy Bradley, Jeremy Bryans, Aleksiy Burdakov, Luca Console, Lia Deotto, Nick Dingle, Kerstin Eder, James Edwards, Amani El-Rayes, Peter Harrison, Leïla Kloul, Luk Knapen, Will Knottenbelt, Marta Kwiatkowska, Ting Ting Lee, Steve Minton, Pete Mitton, Friedger Müffke, Gethin Norman, Claudia Picardi, Rob Pooley, Kris Powell, Marina Ribaudo, Roman Samarov, Nigel Thomas, David Thornley, Fabrice Valois, Wim Vanderbauwhede.

Thank you