Adventures in Systems Biology

Jane Hillston.
LFCS, University of Edinburgh

18th July 2005

Joint work with Muffy Calder and Stephen Gilmore

Outline

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.
- The project has sought to investigate and exploit the interplay between the process algebra and the continuous time Markov chain (CTMC).

PEPA for Performance Modelling

Model Construction: Compositionality leads to

- ease of construction
- reusable submodels
- easy to understand models

PEPA for Performance Modelling

Model Construction: Compositionality leads to

- ease of construction
- reusable submodels
- easy to understand models

Model Manipulation: Equivalence relations lead to

- term rewriting/state space reduction techniques
- aggregation techniques based on lumpability

PEPA for Performance Modelling

Model Construction: Compositionality leads to

- ease of construction
- reusable submodels
- easy to understand models

Model Manipulation: Equivalence relations lead to

- term rewriting/state space reduction techniques
- aggregation techniques based on lumpability

Model Solution: Formal semantics: lead to

- automatic identification of classes of models susceptible to efficient solution
- use of logics to express performance measures

Example PEPA Case Studies

- 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)

Motivation: why use process algebras to model pathways?

- Process algebraic formulation makes interactions/constraints explicit – not the case with classical ordinary differential equation (ODE) models. Structure can also be apparent.

Motivation: why use process algebras to model pathways?

- Process algebraic formulation makes interactions/constraints explicit – not the case with classical ordinary differential equation (ODE) models. Structure can also be apparent.
- Equivalence relations allow formal comparison of high-level descriptions.

Motivation: why use process algebras to model pathways?

- Process algebraic formulation makes interactions/constraints explicit – not the case with classical ordinary differential equation (ODE) models. Structure can also be apparent.
- Equivalence relations allow formal comparison of high-level descriptions.
- Access to techniques for reasoning about livelocks, deadlocks

Motivation: why use process algebras to model pathways?

- Process algebraic formulation makes interactions/constraints explicit – not the case with classical ordinary differential equation (ODE) models. Structure can also be apparent.
- Equivalence relations allow formal comparison of high-level descriptions.
- Access to techniques for reasoning about livelocks, deadlocks and when using a stochastic process algebra, performance (transient or steady state) and sophisticated reasoning using stochastic logics such as CSL and probabilistic model checking.

Stochastic Process Algebra

Attractive features of process algebras

Stochastic Process Algebra

Attractive features of process algebras

- Compositionality

Stochastic Process Algebra

Attractive features of process algebras

- Compositionality
- Formal definition

Stochastic Process Algebra

Attractive features of process algebras

- Compositionality
- Formal definition
- Parsimony

Stochastic Process Algebra

Attractive features of process algebras + Quantification

- Compositionality
- Formal definition
- Parsimony

Stochastic Process Algebra

Attractive features of process algebras

- Compositionality
- Formal definition
- Parsimony

+ Quantification

- Actions have duration

Stochastic Process Algebra

Attractive features of process algebras

- Compositionality
- Formal definition
- Parsimony

+ Quantification

- Actions have duration
- Probabilistic branching

Stochastic Process Algebra

Attractive features of process algebras

- Compositionality
- Formal definition
- Parsimony

+ Quantification

- Actions have duration
- Probabilistic branching

Performance Evaluation Process Algebra (PEPA)

Stochastic Process Algebra

Attractive features of process algebras

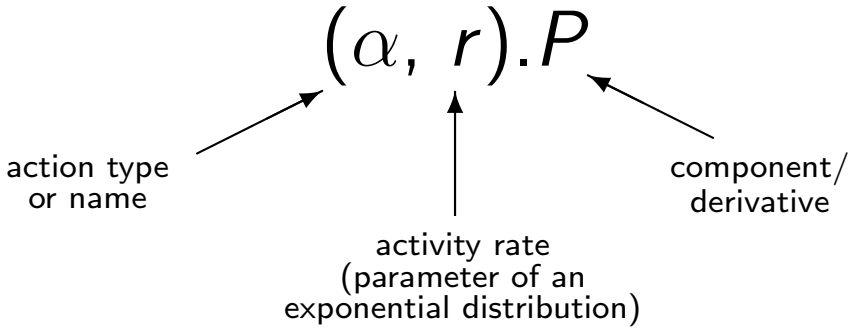
- Compositionality
- Formal definition
- Parsimony

$+$ Quantification

- Actions have duration
- Probabilistic branching

Performance Evaluation Process Algebra (PEPA)

Models are constructed from components which engage in activities.



Stochastic Process Algebra

Attractive features of process algebras

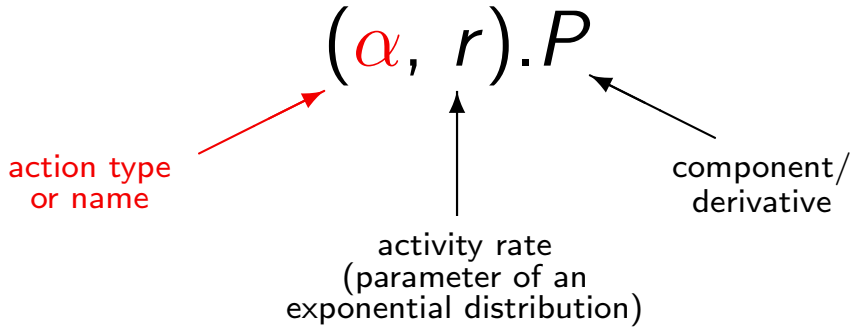
- Compositionality
- Formal definition
- Parsimony

+ Quantification

- Actions have duration
- Probabilistic branching

Performance Evaluation Process Algebra (PEPA)

Models are constructed from components which engage in activities.



Stochastic Process Algebra

Attractive features of process algebras

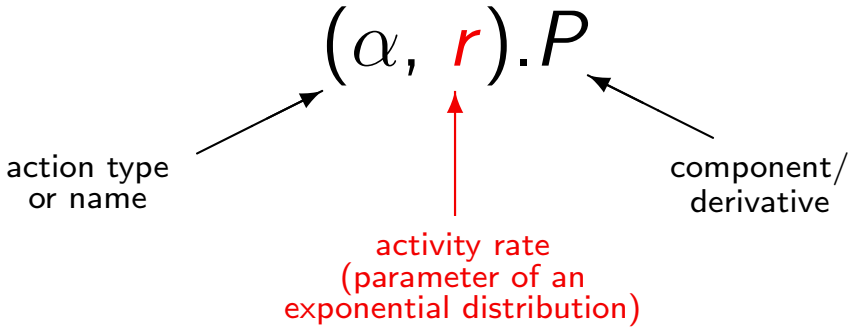
- Compositionality
- Formal definition
- Parsimony

+ Quantification

- Actions have duration
- Probabilistic branching

Performance Evaluation Process Algebra (PEPA)

Models are constructed from components which engage in activities.



Stochastic Process Algebra

Attractive features of process algebras

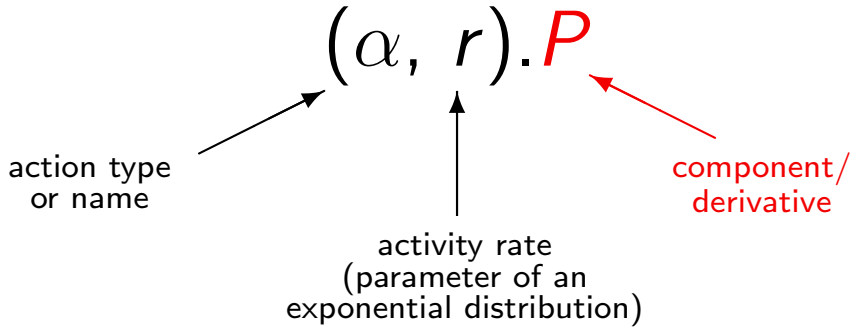
- Compositionality
- Formal definition
- Parsimony

+ Quantification

- Actions have duration
- Probabilistic branching

Performance Evaluation Process Algebra (PEPA)

Models are constructed from components which engage in activities.



Generating a CTMC

The corresponding Continuous Time Markov Chain (CTMC) is derived automatically from the structured operational semantics which define the language:

Generating a CTMC

The corresponding Continuous Time Markov Chain (CTMC) is derived automatically from the structured operational semantics which define the language:

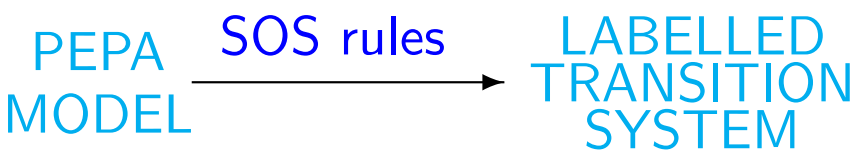
PEPA
MODEL

Generating a CTMC

The corresponding Continuous Time Markov Chain (CTMC) is derived automatically from the structured operational semantics which define the language:

Generating a CTMC

The corresponding Continuous Time Markov Chain (CTMC) is derived automatically from the structured operational semantics which define the language:



Generating a CTMC

The corresponding Continuous Time Markov Chain (CTMC) is derived automatically from the structured operational semantics which define the language:

Generating a CTMC

The corresponding Continuous Time Markov Chain (CTMC) is derived automatically from the structured operational semantics which define the language:

Note that contrary to classical process algebras, the multiplicity of transitions is important.

Generating a CTMC

The corresponding Continuous Time Markov Chain (CTMC) is derived automatically from the structured operational semantics which define the language:

Note that contrary to classical process algebras, the multiplicity of transitions is important.

The natural equivalence relation in the process algebra strong equivalence or Markovian bisimulation generates a lumpable partition in the underlying CTMC.

Generating a CTMC

The corresponding Continuous Time Markov Chain (CTMC) is derived automatically from the structured operational semantics which define the language:

Note that contrary to classical process algebras, the multiplicity of transitions is important.

The natural equivalence relation in the process algebra strong equivalence or Markovian bisimulation generates a lumpable partition in the underlying CTMC.

PEPA

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

PEPA

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

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

PEPA

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

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

PEPA

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

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

PEPA

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

PREFIX:	$(\alpha, r).S$	designated first action
CHOICE:	$S + S$	competing components (race policy)
CONSTANT:	$A \stackrel{def}{=} S$	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 $\alpha \in L \Rightarrow \alpha \rightarrow \tau$

Outline

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.

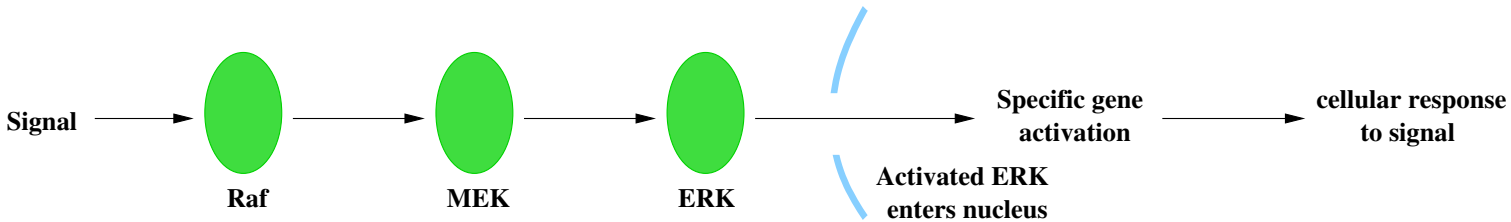
New application domains: biochemical signalling pathways

- 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

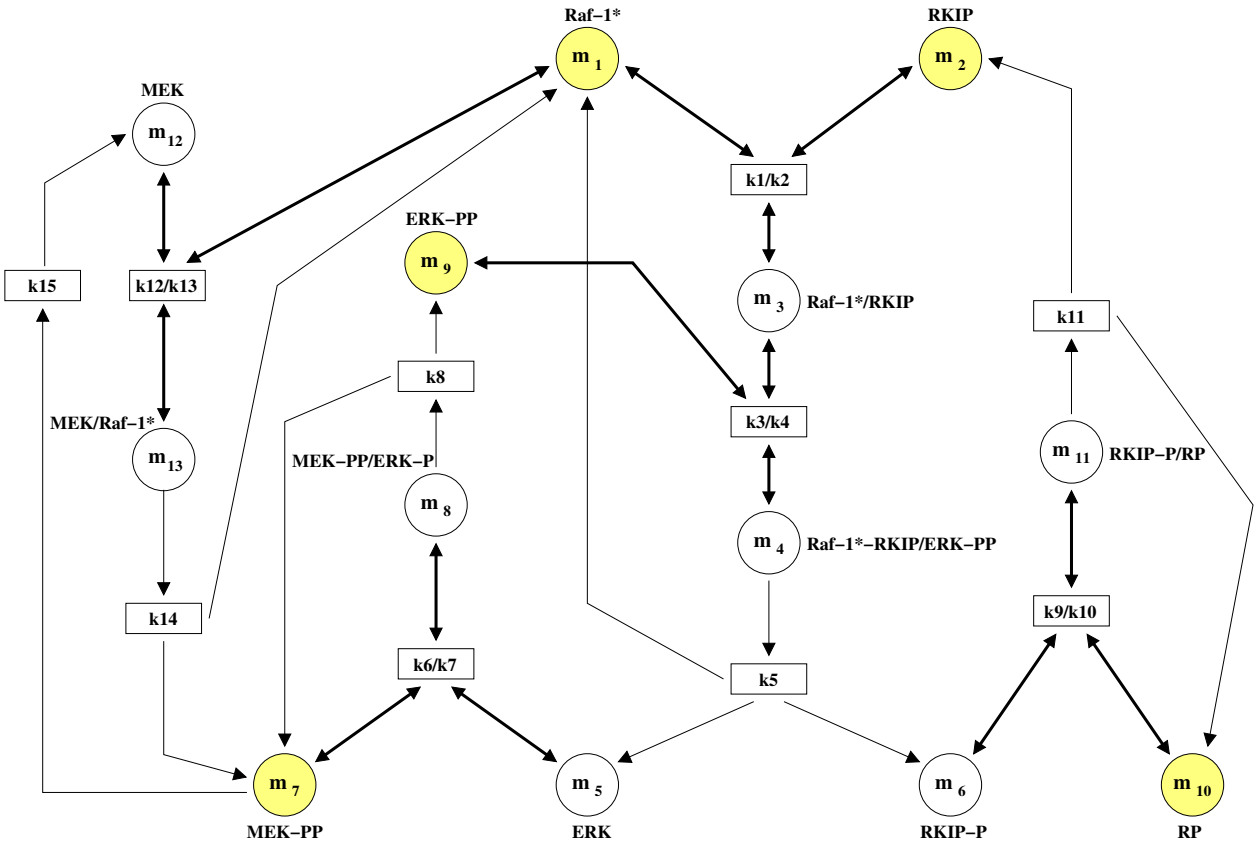
Extracellular signalling — communication between cells.

- 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.



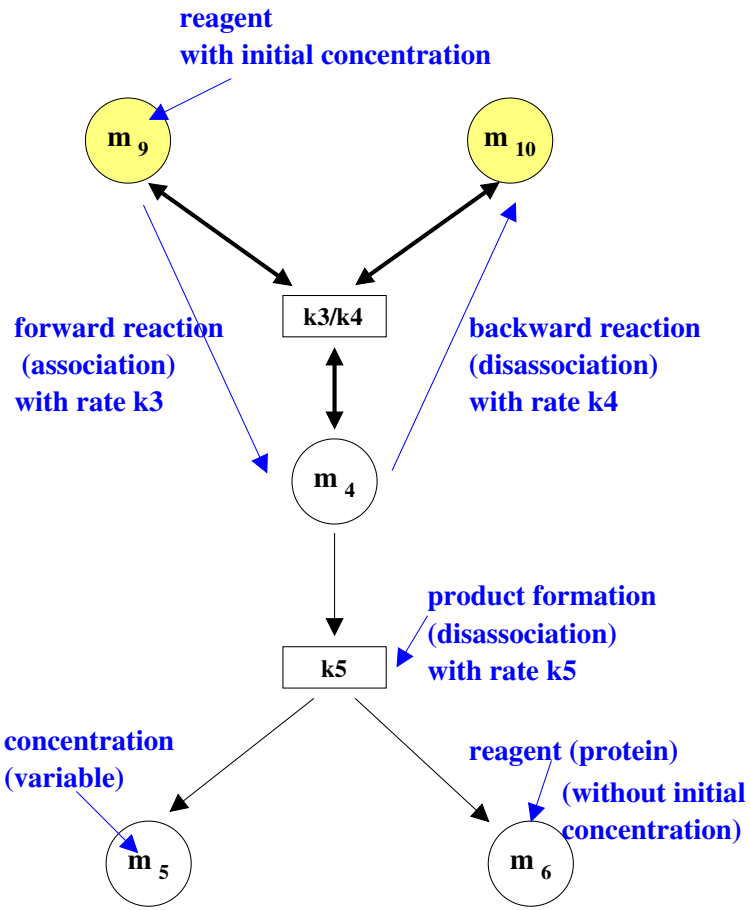
Special relevance to cancer research because when pathways operate abnormally cells divide uncontrollably.

The ERK signalling pathway

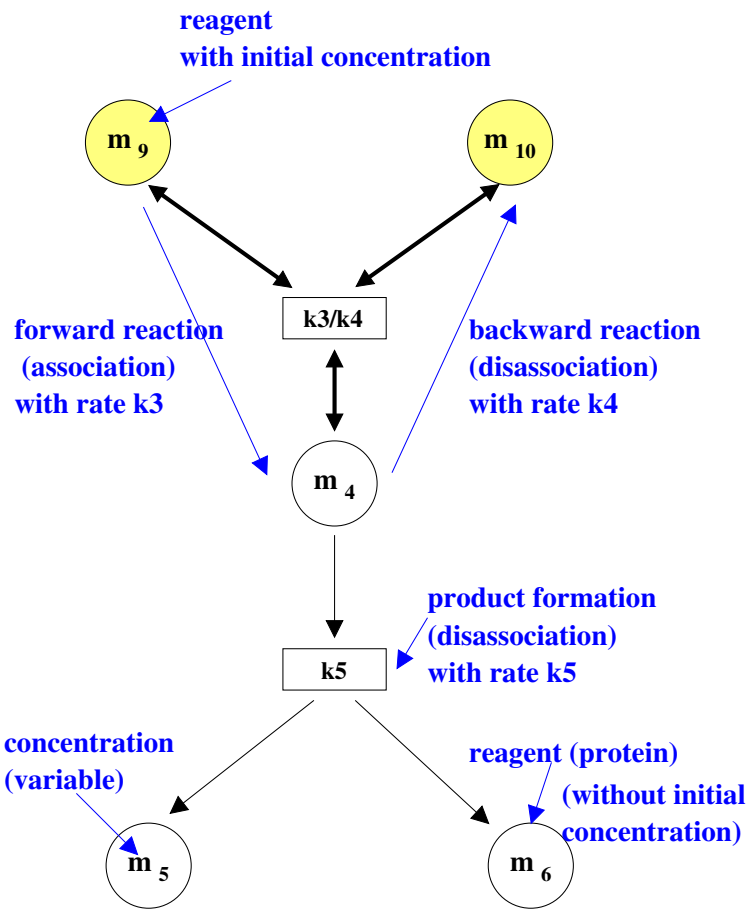


Researchers at the Beatson Institute in Glasgow are investigating the hypothesis that the protein RKIP has a regulatory influence on the ERK signalling pathway.

Dynamics of cell signalling

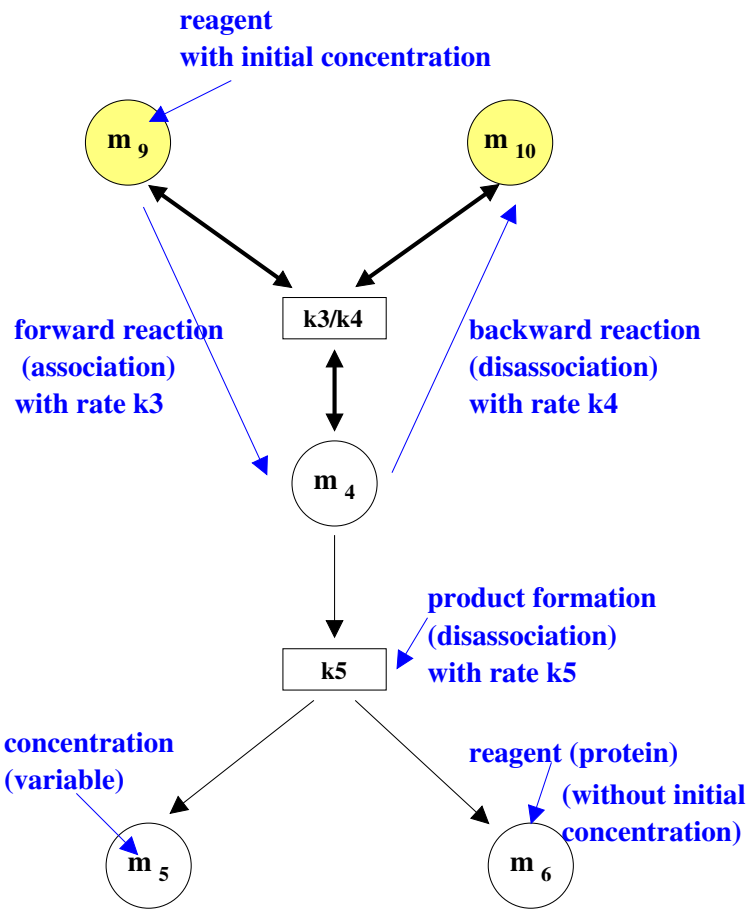


Dynamics of cell signalling



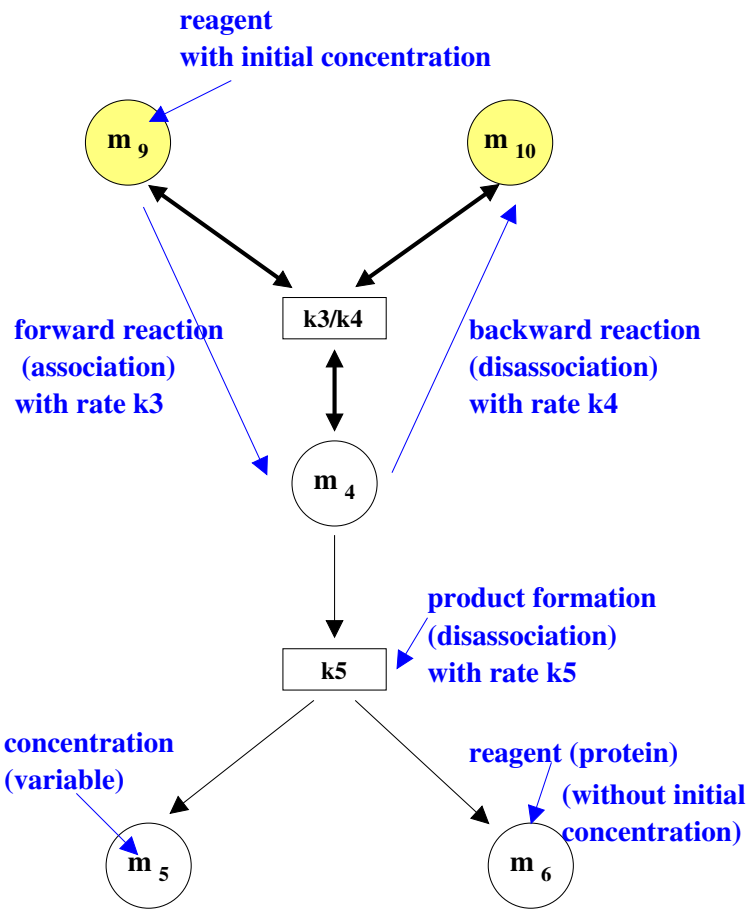
- Bi-directional arrows denote both forward and backward reactions;

Dynamics of cell signalling



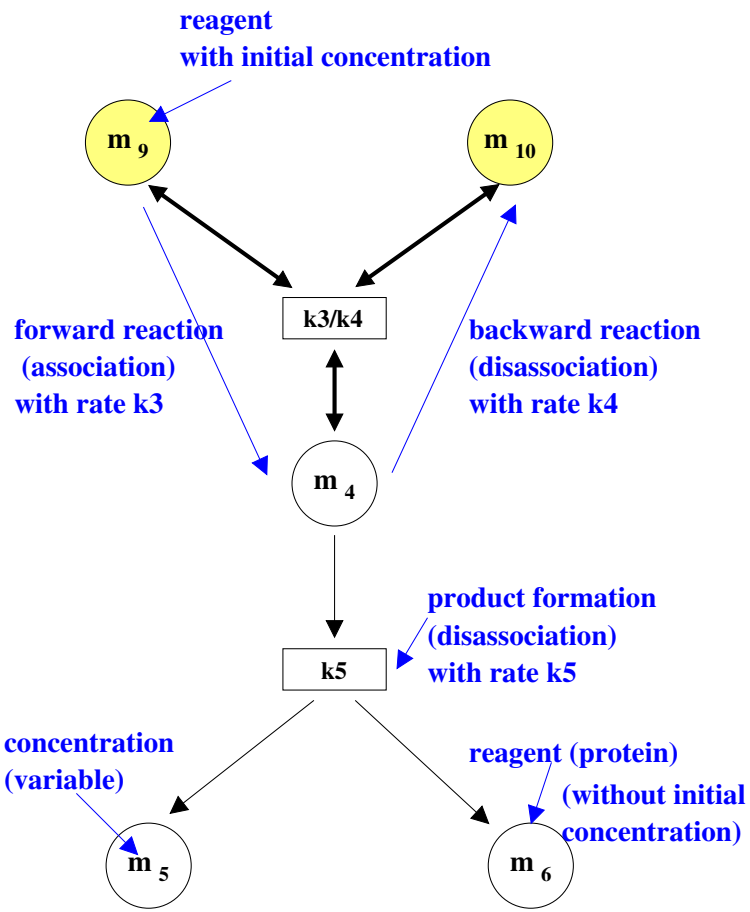
- Bi-directional arrows denote both forward and backward reactions;
- Uni-directional arrows denote reactions which are disassociations.

Dynamics of cell signalling



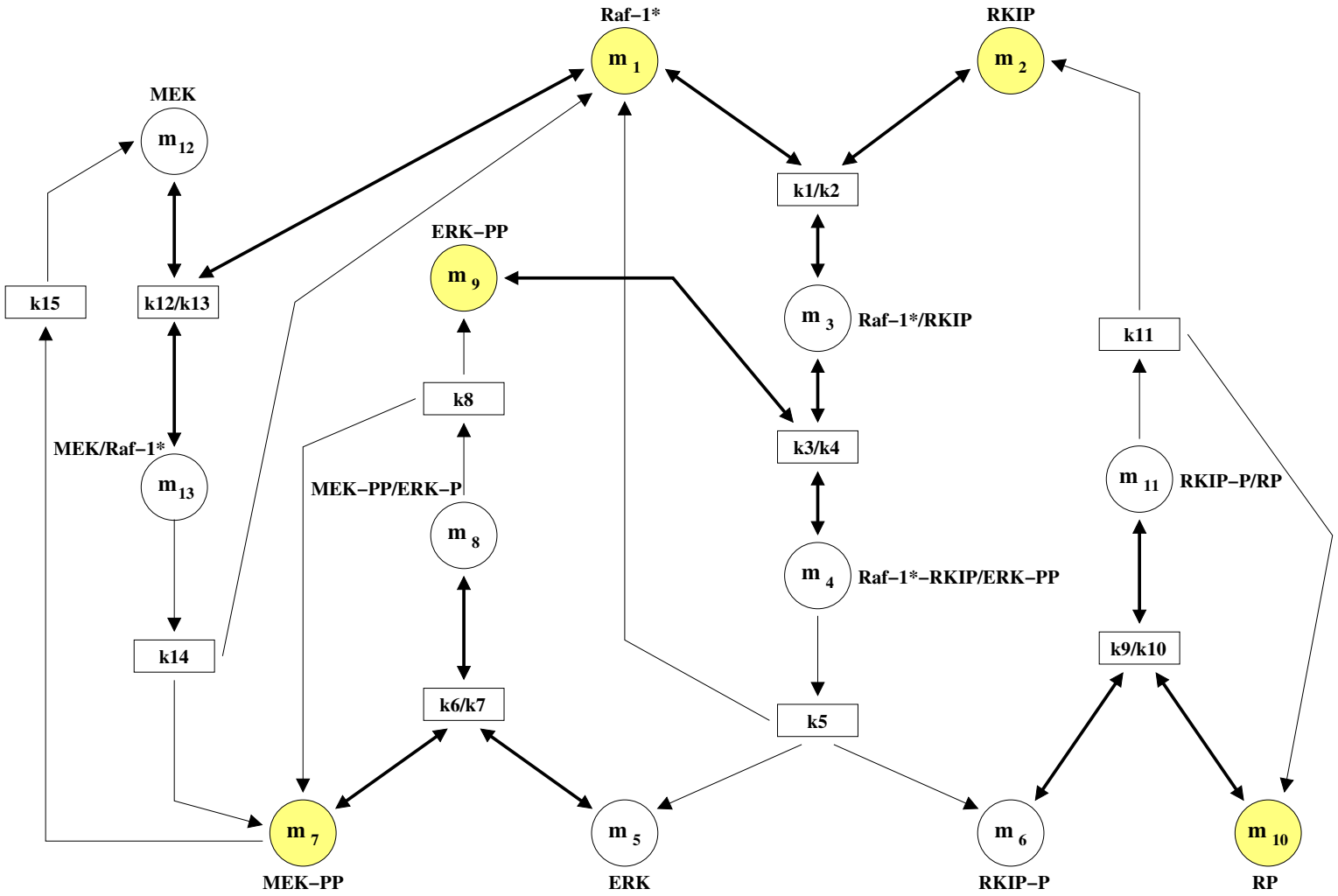
- Bi-directional arrows denote both forward and backward reactions;
- Uni-directional arrows denote reactions which are disassociations.
- Each reagent has a variable concentration, denoted m_i.

Dynamics of cell signalling



- Bi-directional arrows denote both forward and backward reactions;
- Uni-directional arrows denote reactions which are disassociations.
- Each reagent has a variable concentration, denoted m_i.
- Each reaction has a corresponding rate constant, e.g. k3, but the rate at which the reaction takes place is the product of this rate constant and the current concentrations of the used substrates.

The ERK signalling pathway



PEPA models of the pathway

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

PEPA models of the pathway

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

- The first focuses on the reagents and the variations in their concentrations caused by the reactions they undertake. Each distinct protein in the pathway is represented as a distinct PEPA component.

PEPA models of the pathway

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

- The first focuses on the reagents and the variations in their concentrations caused by the reactions they undertake. Each distinct protein in the pathway is represented as a distinct PEPA component.
- The second focuses on the sub-pathways within the pathway. Starting from a reagent which exhibits an initial concentration each sub-pathway tracks the valid reactions the substrate may pass through before returning the initial concentration of the initial substrate.

Dealing with concentrations

The dynamics or kinetics of the pathway are dependent on the concentrations of the reactants.

Dealing with concentrations

The dynamics or kinetics of the pathway are dependent on the concentrations of the reactants.

In both our models concentrations are treated in an abstract fashion.

Dealing with concentrations

The dynamics or kinetics of the pathway are dependent on the concentrations of the reactants.

In both our models concentrations are treated in an abstract fashion.

- In the first we distinguish only high and low concentrations.

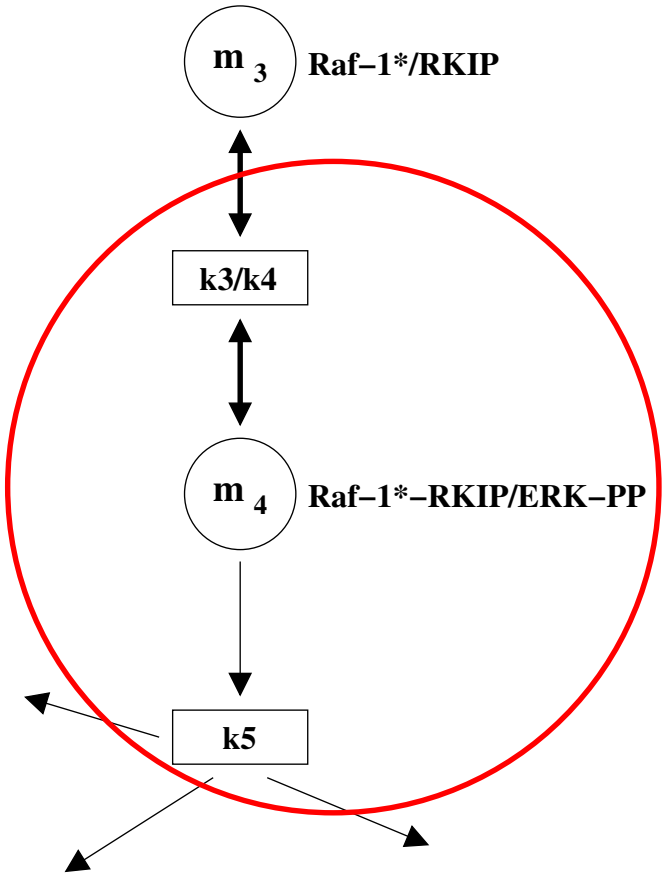
Dealing with concentrations

The dynamics or kinetics of the pathway are dependent on the concentrations of the reactants.

In both our models concentrations are treated in an abstract fashion.

- In the first we distinguish only high and low concentrations.
- In the second concentrations are not represented explicitly at all but may be inferred from the state of the appropriate pathways.

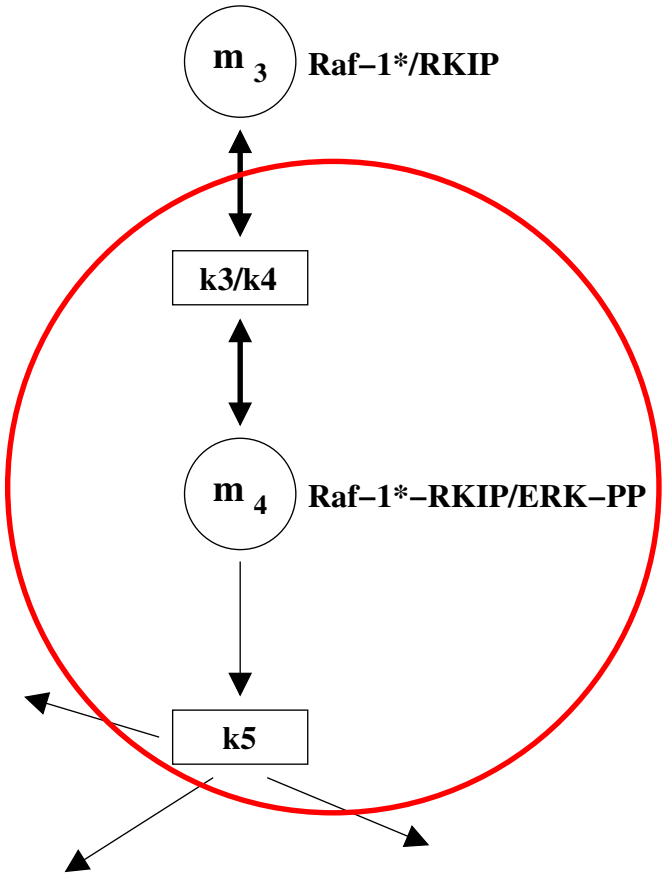
PEPA components of the reagent-centric model



$$\text{Raf-1}^*/\text{RKIP}/\text{ERK-PP}_\text{H} \stackrel{def}{=} (k5product, k_5).\text{Raf-1}^*/\text{RKIP}/\text{ERK-PP}_\text{L} + (k4react, k_4).\text{Raf-1}^*/\text{RKIP}/\text{ERK-PP}_\text{L}$$

$$\text{Raf-1}^*/\text{RKIP}/\text{ERK-PP}_\text{L} \stackrel{def}{=} (k3react, k_3).\text{Raf-1}^*/\text{RKIP}/\text{ERK-PP}_\text{H}$$

PEPA components of the reagent-centric model

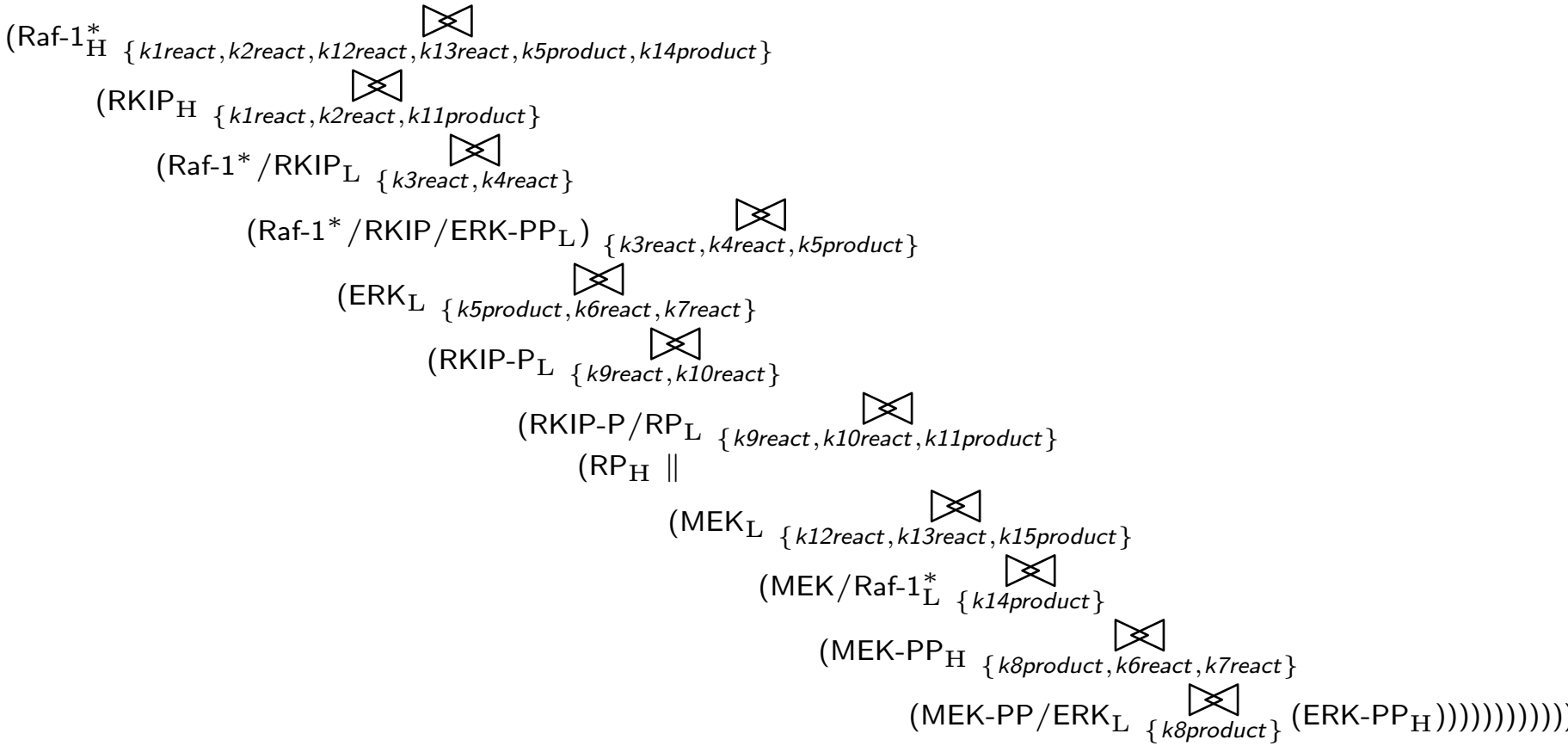


$$\text{Raf-1}^*/\text{RKIP}/\text{ERK-PP}_\text{H} \stackrel{def}{=} (k5product, k_5).\text{Raf-1}^*/\text{RKIP}/\text{ERK-PP}_\text{L} + (k4react, k_4).\text{Raf-1}^*/\text{RKIP}/\text{ERK-PP}_\text{L}$$

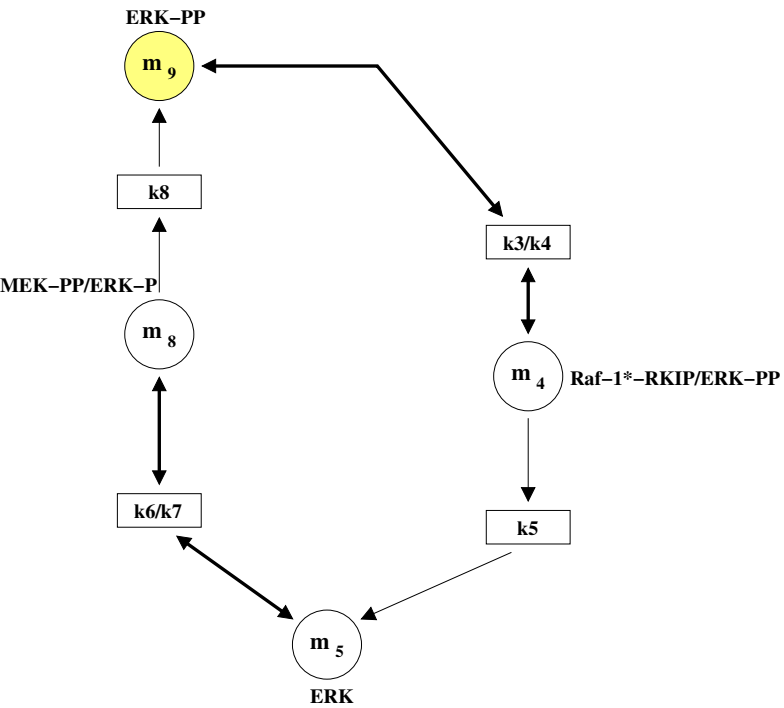
$$\text{Raf-1}^*/\text{RKIP}/\text{ERK-PP}_\text{L} \stackrel{def}{=} (k3react, k_3).\text{Raf-1}^*/\text{RKIP}/\text{ERK-PP}_\text{H}$$

Each reagent gives rise to a pair of PEPA definitions, one for high concentration and one for low concentration.

Configuration of the reagent-centric model



PEPA components of the pathway-centric model



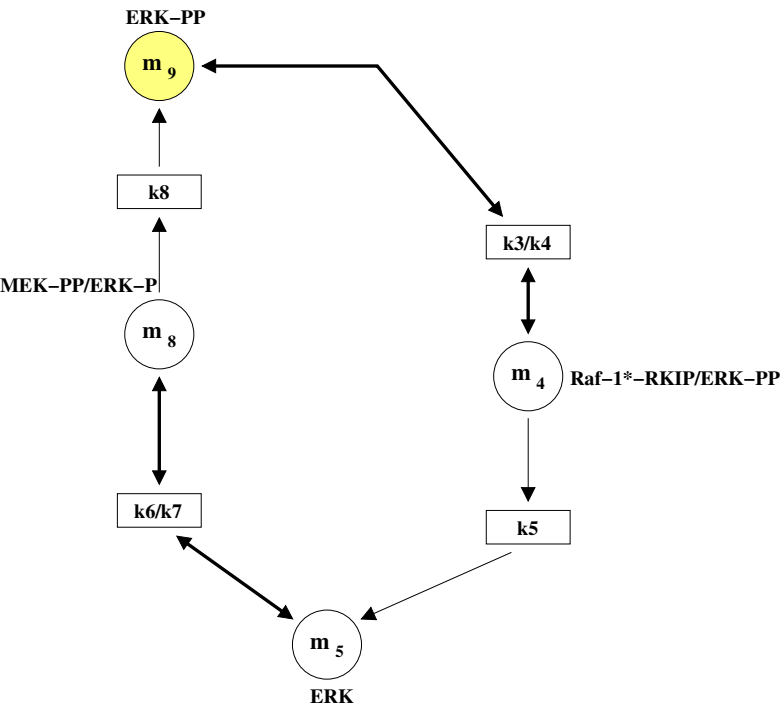
$$Pathway_{30} \stackrel{def}{=} (k3react, k_3).Pathway_{31}$$

$$Pathway_{31} \stackrel{def}{=} (k5product, k_5).Pathway_{32} + (k4react, k_4).Pathway_{30}$$

$$Pathway_{32} \stackrel{def}{=} (k6react, k_6).Pathway_{33}$$

$$Pathway_{33} \stackrel{def}{=} (k8product, k_8).Pathway_{30} + (k7react, k_7).Pathway_{32}$$

PEPA components of the pathway-centric model



$$
\begin{aligned}
Pathway_{30} &\stackrel{def}{=} (k3react, k_3).Pathway_{31} \\
Pathway_{31} &\stackrel{def}{=} (k5product, k_5).Pathway_{32} \\
&\quad +(k4react, k_4).Pathway_{30} \\
Pathway_{32} &\stackrel{def}{=} (k6react, k_6).Pathway_{33} \\
Pathway_{33} &\stackrel{def}{=} (k8product, k_8).Pathway_{30} \\
&\quad +(k7react, k_7).Pathway_{32}
\end{aligned}
$$

For each reagent that has an initial concentration we define the sub-pathway generated by the progression of that reagent.

Configuration of the pathway-centric model

To complete the model we must ensure that the pathways are constrained to interact appropriately, i.e. an association can only occur when all necessary reagents are in a state to make the reaction.

$$((((Pathway_{50} \underset{\{k12react, k13react, k14product\}}{\bowtie} Pathway_{40}) \underset{\{k6react, k7react, k8product, k3react, k4react, k5product\}}{\bowtie} Pathway_{30}) \underset{\{k1react, k2react, k3react, k4react, k5product\}}{\bowtie} Pathway_{20}) \underset{\{k9react, k10react, k11product\}}{\bowtie} Pathway_{10})$$

Note that this is much simpler in this case than for the reagent model.

Commentary on the models

- Neither model currently "correctly" captures the rate of interaction – concentrations are discretized and rates are assumed to be constant within levels:

Commentary on the models

- Neither model currently "correctly" captures the rate of interaction – concentrations are discretized and rates are assumed to be constant within levels: In these examples only using high and low to modify rates in the sense of enabling or disabling activities.

Commentary on the models

- Neither model currently "correctly" captures the rate of interaction – concentrations are discretized and rates are assumed to be constant within levels: In these examples only using high and low to modify rates in the sense of enabling or disabling activities.
- The reagent-centric model can be regarded as a fine-grained view of the system.

Commentary on the models

- Neither model currently "correctly" captures the rate of interaction – concentrations are discretized and rates are assumed to be constant within levels: In these examples only using high and low to modify rates in the sense of enabling or disabling activities.
- The reagent-centric model can be regarded as a fine-grained view of the system.
- The pathway-centric model can be regarded as a more structural, coarse-grained view of the system.

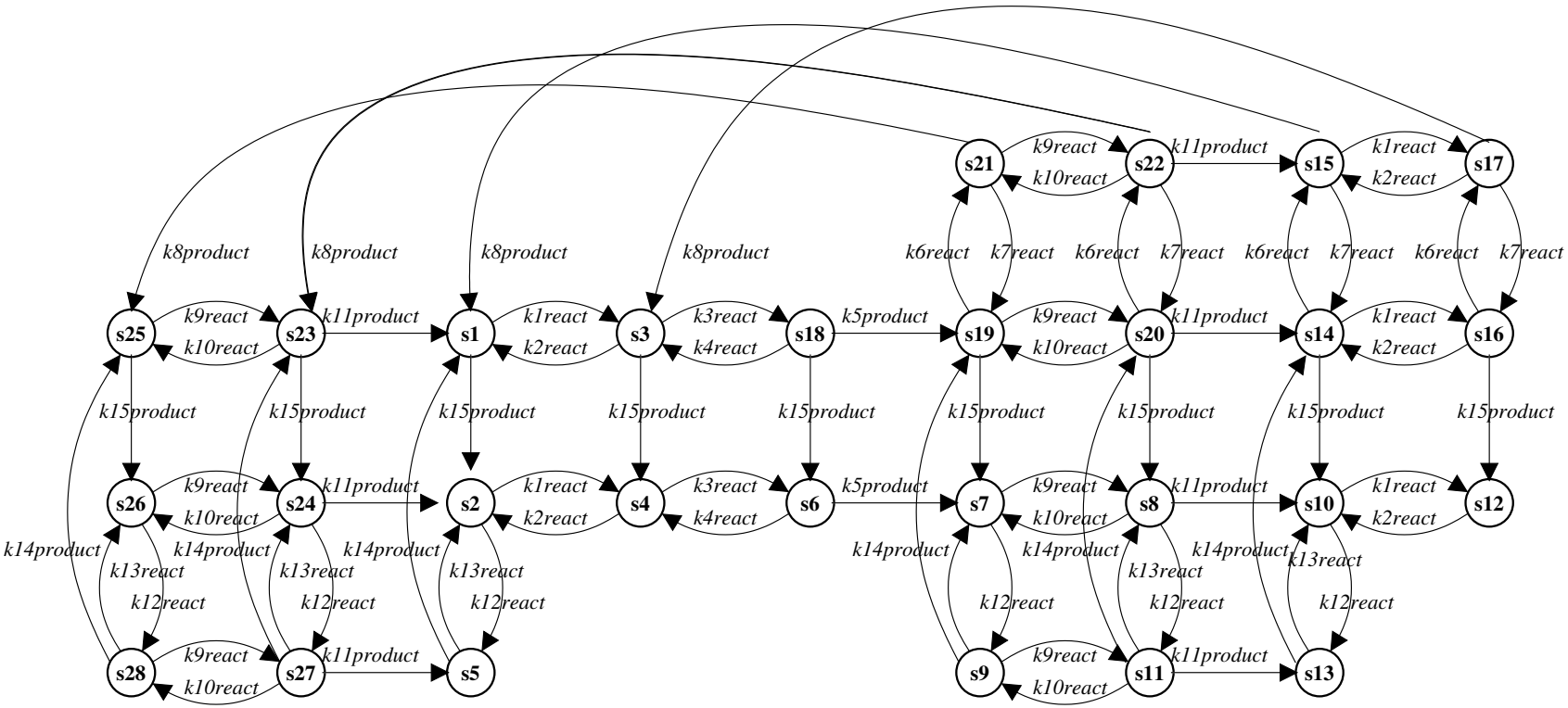
Commentary on the models

- Neither model currently "correctly" captures the rate of interaction – concentrations are discretized and rates are assumed to be constant within levels: In these examples only using high and low to modify rates in the sense of enabling or disabling activities.
- The reagent-centric model can be regarded as a fine-grained view of the system.
- The pathway-centric model can be regarded as a more structural, coarse-grained view of the system.
- Applying the structured operational semantics reveals that they are strongly bisimilar

Commentary on the models

- Neither model currently "correctly" captures the rate of interaction – concentrations are discretized and rates are assumed to be constant within levels: In these examples only using high and low to modify rates in the sense of enabling or disabling activities.
- The reagent-centric model can be regarded as a fine-grained view of the system.
- The pathway-centric model can be regarded as a more structural, coarse-grained view of the system.
- Applying the structured operational semantics reveals that they are strongly bisimilar (in fact, in this case, isomorphic).

The state space



The bisimulation

s_1	$(\text{Raf-1}^*_{\text{H}}, \text{RKIP}_{\text{H}}, \text{Raf-1}^*/\text{RKIP}_{\text{L}}, \text{Raf-1}^*/\text{RKIP/ERK-PP}_{\text{L}}, \text{ERK}_{\text{L}}, \text{RKIP-P}_{\text{L}}, \text{RKIP-P/RP}_{\text{L}}, \text{RP}_{\text{H}}, \text{MEK}_{\text{L}}, \text{MEK/Raf-1}^*_{\text{L}}, \text{MEK-PP}_{\text{H}}, \text{MEK-PP/ERK}_{\text{L}}, \text{ERK-PP}_{\text{H}})$	$(Pathway_{50}, Pathway_{40}, Pathway_{30}, Pathway_{20}, Pathway_{10})$

The bisimulation

s_1	$(\text{Raf-1}^*_{\text{H}}, \text{RKIP}_{\text{H}}, \text{Raf-1}^*/\text{RKIP}_{\text{L}}, \text{Raf-1}^*/\text{RKIP/ERK-PP}_{\text{L}}, \text{ERK}_{\text{L}}, \text{RKIP-P}_{\text{L}}, \text{RKIP-P/RP}_{\text{L}}, \text{RP}_{\text{H}}, \text{MEK}_{\text{L}}, \text{MEK/Raf-1}^*_{\text{L}}, \text{MEK-PP}_{\text{H}}, \text{MEK-PP/ERK}_{\text{L}}, \text{ERK-PP}_{\text{H}})$	$(Pathway_{50}, Pathway_{40}, Pathway_{30}, Pathway_{20}, Pathway_{10})$
s_2	$(\text{Raf-1}^*_{\text{H}}, \text{RKIP}_{\text{H}}, \text{Raf-1}^*/\text{RKIP}_{\text{L}}, \text{Raf-1}^*/\text{RKIP/ERK-PP}_{\text{L}}, \text{ERK}_{\text{L}}, \text{RKIP-P}_{\text{L}}\text{RKIP-P/RP}_{\text{L}}, \text{RP}_{\text{H}}, \text{MEK}_{\text{H}}, \text{MEK/Raf-1}^*_{\text{L}}, \text{MEK-PP}_{\text{L}}, \text{MEK-PP/ERK}_{\text{L}}, \text{ERK-PP}_{\text{H}})$	$(Pathway_{51}, Pathway_{40}, Pathway_{30}, Pathway_{20}, Pathway_{10})$

The bisimulation

s_1	$(\text{Raf-1}^*_{\text{H}}, \text{RKIP}_{\text{H}}, \text{Raf-1}^*/\text{RKIP}_{\text{L}}, \text{Raf-1}^*/\text{RKIP/ERK-PP}_{\text{L}}, \text{ERK}_{\text{L}}, \text{RKIP-P}_{\text{L}}, \text{RKIP-P}/\text{RP}_{\text{L}}, \text{RP}_{\text{H}}, \text{MEK}_{\text{L}}, \text{MEK}/\text{Raf-1}^*_{\text{L}}, \text{MEK-PP}_{\text{H}}, \text{MEK-PP}/\text{ERK}_{\text{L}}, \text{ERK-PP}_{\text{H}})$	$(Pathway_{50}, Pathway_{40}, Pathway_{30}, Pathway_{20}, Pathway_{10})$
s_2	$(\text{Raf-1}^*_{\text{H}}, \text{RKIP}_{\text{H}}, \text{Raf-1}^*/\text{RKIP}_{\text{L}}, \text{Raf-1}^*/\text{RKIP/ERK-PP}_{\text{L}}, \text{ERK}_{\text{L}}, \text{RKIP-P}_{\text{L}}\text{RKIP-P}/\text{RP}_{\text{L}}, \text{RP}_{\text{H}}, \text{MEK}_{\text{H}}, \text{MEK}/\text{Raf-1}^*_{\text{L}}, \text{MEK-PP}_{\text{L}}, \text{MEK-PP}/\text{ERK}_{\text{L}}, \text{ERK-PP}_{\text{H}})$	$(Pathway_{51}, Pathway_{40}, Pathway_{30}, Pathway_{20}, Pathway_{10})$
s_3	$(\text{Raf-1}^*_{\text{L}}, \text{RKIP}_{\text{L}}, \text{Raf-1}^*/\text{RKIP}_{\text{H}}, \text{Raf-1}^*/\text{RKIP/ERK-PP}_{\text{L}}, \text{ERK}_{\text{L}}, \text{RKIP-P}_{\text{L}}, \text{RKIP-P}/\text{RP}_{\text{L}}, \text{RP}_{\text{H}}, \text{MEK}_{\text{L}}, \text{MEK}/\text{Raf-1}^*_{\text{L}}, \text{MEK-PP}_{\text{H}}, \text{MEK-PP}/\text{ERK}_{\text{L}}, \text{ERK-PP}_{\text{H}})$	$(Pathway_{50}, Pathway_{41}, Pathway_{30}, Pathway_{21}, Pathway_{10})$

The bisimulation

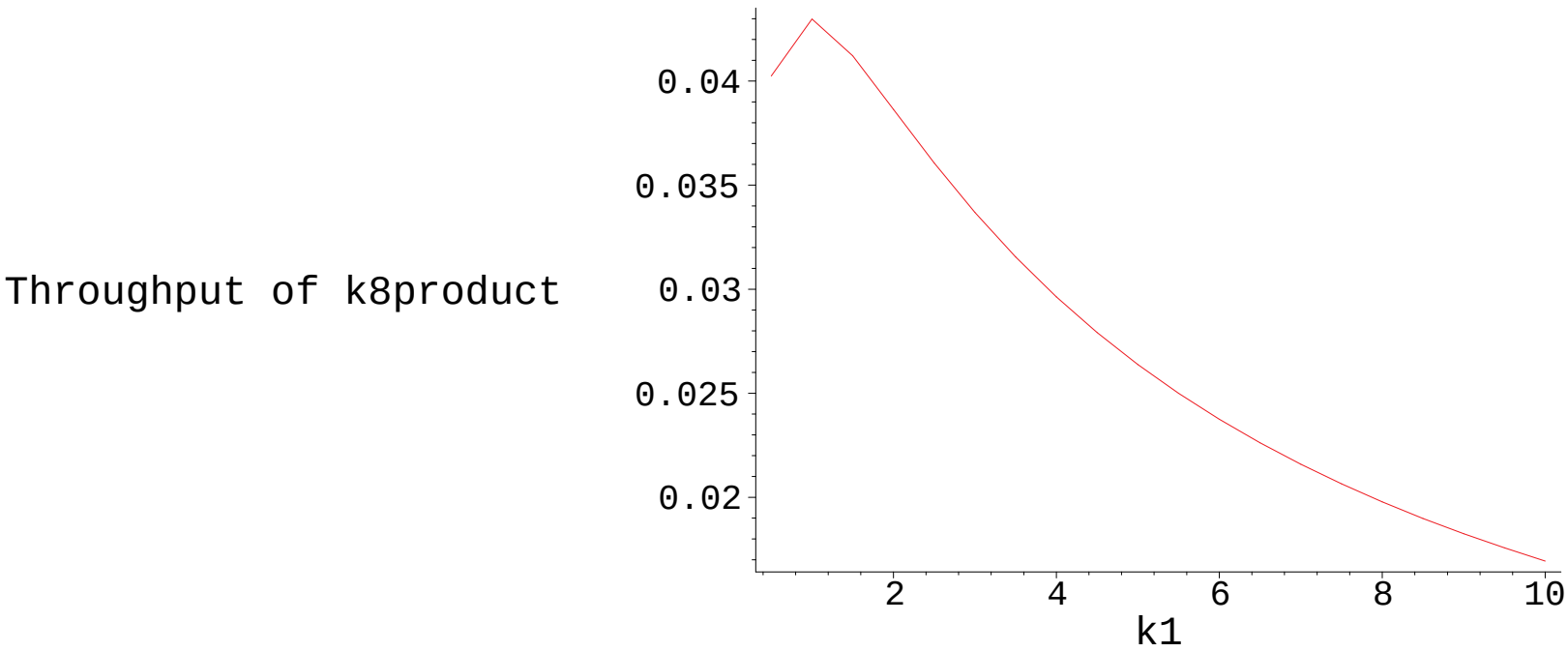
s_1	$(\text{Raf-1}^*_{\text{H}}, \text{RKIP}_{\text{H}}, \text{Raf-1}^*/\text{RKIP}_{\text{L}}, \text{Raf-1}^*/\text{RKIP/ERK-PP}_{\text{L}}, \text{ERK}_{\text{L}}, \text{RKIP-P}_{\text{L}}, \text{RKIP-P/RP}_{\text{L}}, \text{RP}_{\text{H}}, \text{MEK}_{\text{L}}, \text{MEK/Raf-1}^*_{\text{L}}, \text{MEK-PP}_{\text{H}}, \text{MEK-PP/ERK}_{\text{L}}, \text{ERK-PP}_{\text{H}})$	$(Pathway_{50}, Pathway_{40}, Pathway_{30}, Pathway_{20}, Pathway_{10})$
s_2	$(\text{Raf-1}^*_{\text{H}}, \text{RKIP}_{\text{H}}, \text{Raf-1}^*/\text{RKIP}_{\text{L}}, \text{Raf-1}^*/\text{RKIP/ERK-PP}_{\text{L}}, \text{ERK}_{\text{L}}, \text{RKIP-P}_{\text{L}}\text{RKIP-P/RP}_{\text{L}}, \text{RP}_{\text{H}}, \text{MEK}_{\text{H}}, \text{MEK/Raf-1}^*_{\text{L}}, \text{MEK-PP}_{\text{L}}, \text{MEK-PP/ERK}_{\text{L}}, \text{ERK-PP}_{\text{H}})$	$(Pathway_{51}, Pathway_{40}, Pathway_{30}, Pathway_{20}, Pathway_{10})$
s_3	$(\text{Raf-1}^*_{\text{L}}, \text{RKIP}_{\text{L}}, \text{Raf-1}^*/\text{RKIP}_{\text{H}}, \text{Raf-1}^*/\text{RKIP/ERK-PP}_{\text{L}}, \text{ERK}_{\text{L}}, \text{RKIP-P}_{\text{L}}, \text{RKIP-P/RP}_{\text{L}}, \text{RP}_{\text{H}}, \text{MEK}_{\text{L}}, \text{MEK/Raf-1}^*_{\text{L}}, \text{MEK-PP}_{\text{H}}, \text{MEK-PP/ERK}_{\text{L}}, \text{ERK-PP}_{\text{H}})$	$(Pathway_{50}, Pathway_{41}, Pathway_{30}, Pathway_{21}, Pathway_{10})$
$\vdots$	$\vdots$	$\vdots$

The bisimulation

s_1	$(\text{Raf-1}^*_{\text{H}}, \text{RKIP}_{\text{H}}, \text{Raf-1}^*/\text{RKIP}_{\text{L}}, \text{Raf-1}^*/\text{RKIP/ERK-PP}_{\text{L}}, \text{ERK}_{\text{L}}, \text{RKIP-P}_{\text{L}}, \text{RKIP-P/RP}_{\text{L}}, \text{RP}_{\text{H}}, \text{MEK}_{\text{L}}, \text{MEK/Raf-1}^*_{\text{L}}, \text{MEK-PP}_{\text{H}}, \text{MEK-PP/ERK}_{\text{L}}, \text{ERK-PP}_{\text{H}})$	$(Pathway_{50}, Pathway_{40}, Pathway_{30}, Pathway_{20}, Pathway_{10})$
s_2	$(\text{Raf-1}^*_{\text{H}}, \text{RKIP}_{\text{H}}, \text{Raf-1}^*/\text{RKIP}_{\text{L}}, \text{Raf-1}^*/\text{RKIP/ERK-PP}_{\text{L}}, \text{ERK}_{\text{L}}, \text{RKIP-P}_{\text{L}}\text{RKIP-P/RP}_{\text{L}}, \text{RP}_{\text{H}}, \text{MEK}_{\text{H}}, \text{MEK/Raf-1}^*_{\text{L}}, \text{MEK-PP}_{\text{L}}, \text{MEK-PP/ERK}_{\text{L}}, \text{ERK-PP}_{\text{H}})$	$(Pathway_{51}, Pathway_{40}, Pathway_{30}, Pathway_{20}, Pathway_{10})$
s_3	$(\text{Raf-1}^*_{\text{L}}, \text{RKIP}_{\text{L}}, \text{Raf-1}^*/\text{RKIP}_{\text{H}}, \text{Raf-1}^*/\text{RKIP/ERK-PP}_{\text{L}}, \text{ERK}_{\text{L}}, \text{RKIP-P}_{\text{L}}, \text{RKIP-P/RP}_{\text{L}}, \text{RP}_{\text{H}}, \text{MEK}_{\text{L}}, \text{MEK/Raf-1}^*_{\text{L}}, \text{MEK-PP}_{\text{H}}, \text{MEK-PP/ERK}_{\text{L}}, \text{ERK-PP}_{\text{H}})$	$(Pathway_{50}, Pathway_{41}, Pathway_{30}, Pathway_{21}, Pathway_{10})$
$\vdots$	$\vdots$	$\vdots$
s_{28}	$(\text{Raf-1}^*_{\text{L}}, \text{RKIP}_{\text{L}}, \text{Raf-1}^*/\text{RKIP}_{\text{L}}, \text{Raf-1}^*/\text{RKIP/ERK-PP}_{\text{L}}, \text{ERK}_{\text{L}}, \text{RKIP-P}_{\text{H}}, \text{RKIP-P/RP}_{\text{L}}, \text{RP}_{\text{H}}, \text{MEK}_{\text{L}}, \text{MEK/Raf-1}^*_{\text{H}}, \text{MEK-PP}_{\text{L}}, \text{MEK-PP/ERK}_{\text{L}}, \text{ERK-PP}_{\text{H}})$	$(Pathway_{50}, Pathway_{40}, Pathway_{30}, Pathway_{20}, Pathway_{10})$

Quantified analysis – *k8product*

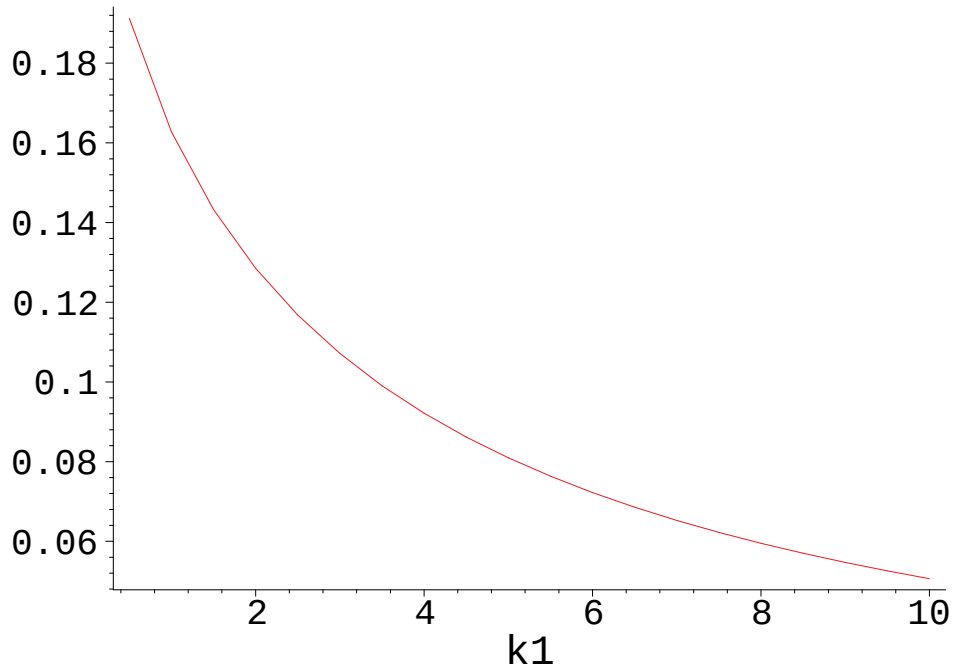
Approximating a variation in the initial concentration of RKIP by varying the rate constant k1, we can assess the impact on the production of ERK-PP.



Quantified analysis – *k14product*

Similarly we can assess the impact on the production of MEK-PP.

Throughput of k14product



Outline

Deriving differential equation: overview

- The ODEs are the familiar mathematical model for the biochemists.

Deriving differential equation: overview

- The ODEs are the familiar mathematical model for the biochemists.
- There should be one equation for each reagent/concentration, indicating how the concentration varies over time.

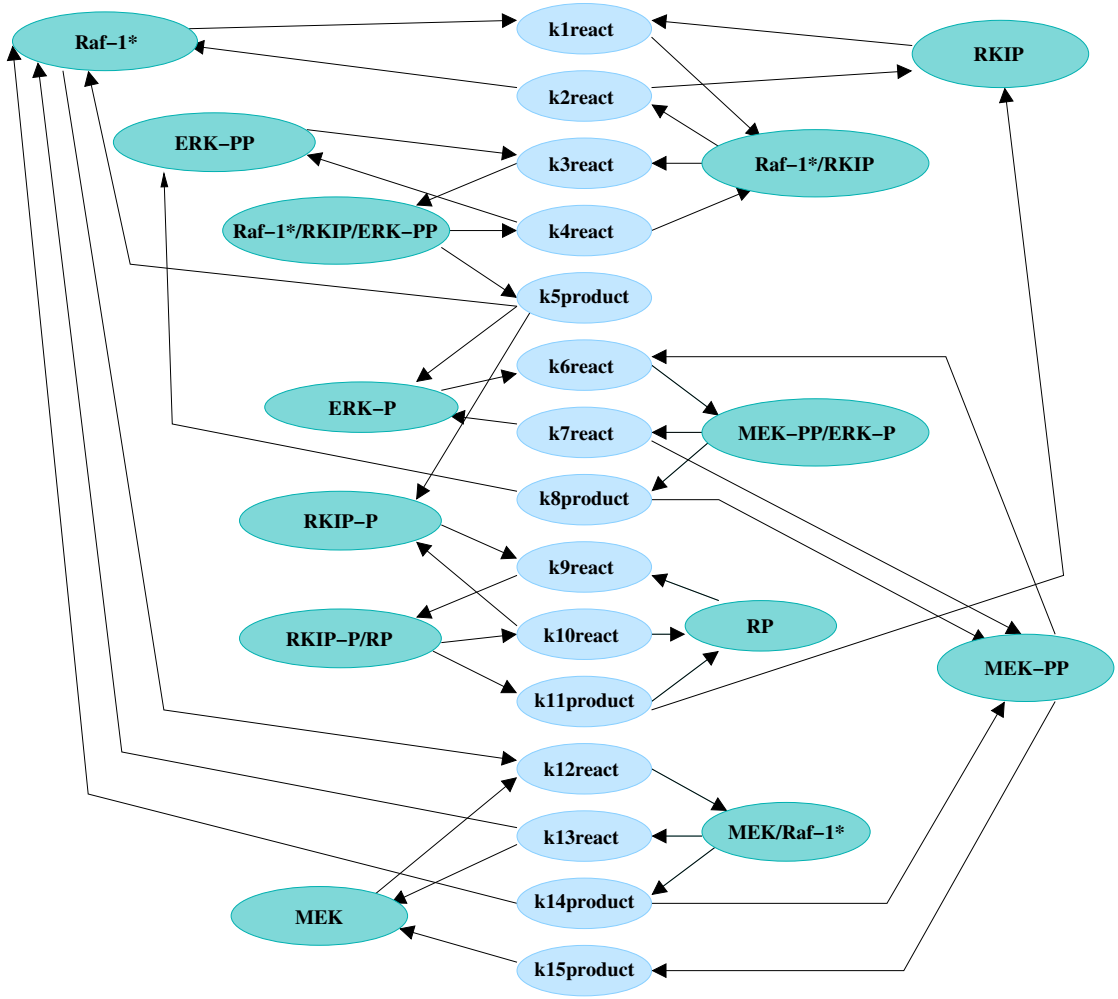
Deriving differential equation: overview

- The ODEs are the familiar mathematical model for the biochemists.
- There should be one equation for each reagent/concentration, indicating how the concentration varies over time.
- Standard solution tools are available for solution of this equations, which are known and trusted by the biologists.

Deriving differential equation: overview

- The ODEs are the familiar mathematical model for the biochemists.
- There should be one equation for each reagent/concentration, indicating how the concentration varies over time.
- Standard solution tools are available for solution of this equations, which are known and trusted by the biologists.
- From the reagent-centric PEPA model we can see the influence of the reactions on each concentration – this is the basis of the derivation via a directed bipartite graph termed the activity graph.

Deriving differential equations: activity graph



Deriving differential equations: activity matrix

	k1	k2	k3	k4	k5	k6	k7	k8	k9	k10	k11	k12	k13	k14	k15
Raf-1*	-1	+1	0	0	+1	0	0	0	0	0	0	-1	+1	+1	0
RKIP	-1	+1	0	0	0	0	0	0	0	0	+1	0	0	0	0
Raf-1*/RKIP	+1	-1	-1	+1	0	0	0	0	0	0	0	0	0	0	0
Raf-1*/RKIP/ERK-PP	0	0	+1	-1	-1	0	0	0	0	0	0	0	0	0	0
ERK	0	0	0	0	+1	-1	+1	0	0	0	0	0	0	0	0
RKIP-P	0	0	0	0	+1	0	0	0	-1	+1	0	0	0	0	0
MEK-PP	0	0	0	0	0	-1	+1	+1	0	0	0	0	0	+1	-1
MEK-PP/ERK	0	0	0	0	0	+1	-1	-1	0	0	0	0	0	0	0
ERK-PP	0	0	-1	+1	0	0	0	+1	0	0	0	0	0	0	0
RP	0	0	0	0	0	0	0	0	-1	+1	+1	0	0	0	0
RKIP-P/RP	0	0	0	0	0	0	0	0	+1	-1	-1	0	0	0	0
MEK	0	0	0	0	0	0	0	0	0	0	0	-1	+1	0	+1
MEK/Raf-1*	0	0	0	0	0	0	0	0	0	0	0	+1	-1	-1	0

Each row corresponds to a single reagent; the entries in a row indicate whether an activity (column) increases the concentration (+1), decreases it (-1) or has no impact (0). Each column corresponds to a single reaction; the negative entries indicate those substrates which are *used* in the reaction.

Deriving differential equations: activity matrix

	$k1$	$k2$	$k3$	$k4$	$k5$	$k6$	...	$conc.$
Raf-1*	-1	$+1$	0	0	$+1$	0	...	m_1
RKIP	-1	$+1$	0	0	0	0	...	m_2
Raf-1*/RKIP	$+\mathbf{1}$	$-\mathbf{1}$	$-\mathbf{1}$	$+\mathbf{1}$	0	0	...	$(\mathbf{m_3})$
Raf-1*/RKIP/ERK-PP	0	0	$+1$	-1	-1	0	...	m_4
ERK	0	0	0	0	$+1$	-1	...	m_5
RKIP-P	0	0	0	0	$+1$	0	...	m_6
MEK-PP	0	0	0	0	0	-1	...	m_7
MEK-PP/ERK	0	0	0	0	0	$+1$	...	m_8
ERK-PP	0	0	-1	$+1$	0	0	...	m_9
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	

$$\frac{d\,m_3(t)}{dt} = k_1\, m_1(t) m_2(t) - k_2\, m_3(t) - k_3\, m_3(t) m_9(t) + k_4\, m_4(t)$$

Differential equations

Using this approach, we can validate our PEPA model against the system of ODEs indepedently derived by the biochemists:

$$\begin{aligned}
\frac{dm_1(t)}{dt} &= -k_1 m_1(t) m_2(t) + k_2 m_3(t) + k_5 m_4(t) - k_{12} m_1(t) m_{12}(t) \\
&\quad + k_{13} m_{13}(t) + k_{14} m_{13}(t) \\
\frac{dm_2(t)}{dt} &= -k_1 m_1(t) m_2(t) + k_2 m_3(t) + k_{11} m_{11}(t) \\
\frac{dm_3(t)}{dt} &= k_1 m_1(t) m_2(t) - k_2 m_3(t) - k_3 m_3(t) m_9(t) + k_4 m_4(t) \\
&\vdots \qquad\qquad \vdots \\
\frac{dm_{13}(t)}{dt} &= k_{12} m_1(t) m_1 2(t) - k_{13} m_{13}(t) - k_{14} m_{13}(t)
\end{aligned}$$

ODEs via PEPA

There are several advantages to be gained by introducing a process algebra model as an intermediary to the derivation of the ODEs.

ODEs via PEPA

There are several advantages to be gained by introducing a process algebra model as an intermediary to the derivation of the ODEs.

- The ODEs can be automatically generated from the descriptive process algebra model, thus reducing human error.

ODEs via PEPA

There are several advantages to be gained by introducing a process algebra model as an intermediary to the derivation of the ODEs.

- The ODEs can be automatically generated from the descriptive process algebra model, thus reducing human error.
- The formality of the process algebra model and its underlying semantics allow us to derive properties of the model, such as freedom from deadlock, before numerical analysis is carried out.

ODEs via PEPA

There are several advantages to be gained by introducing a process algebra model as an intermediary to the derivation of the ODEs.

- The ODEs can be automatically generated from the descriptive process algebra model, thus reducing human error.
- The formality of the process algebra model and its underlying semantics allow us to derive properties of the model, such as freedom from deadlock, before numerical analysis is carried out.
- The algebraic formulation of the model makes clear the interactions between the biochemical entities, or substrates. The style of modelling is descriptive, close to informal graphical representations that biochemists already use.

Performance evaluation: new mathematical structures

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

- Closed form analytical models;

Performance evaluation: new mathematical structures

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

- Closed form analytical models;
- Simulations; or

Performance evaluation: 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)

Performance evaluation: 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

- 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.

Differential equations from arbitrary PEPA models

- 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 the state of the system as the count of the current number of each possible local derivative or component type.
- We can approximate the behaviour of the model by treating the number of 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 ordinary differential equation (assuming rates are deterministic).

Differential equations from arbitrary PEPA models

- 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 the state of the system as the count of the current number of each possible local derivative or component type.
- We can approximate the behaviour of the model by treating the number of 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 ordinary differential equation (assuming rates are deterministic).

Differential equations from arbitrary PEPA models

- 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 the state of the system as the count of the current number of each possible local derivative or component type.
- We can approximate the behaviour of the model by treating the number of 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 ordinary differential equation (assuming rates are deterministic).

Differential equations from arbitrary PEPA models

- 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 the state of the system as the count of the current number of each possible local derivative or component type.
- We can approximate the behaviour of the model by treating the number of 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 ordinary differential equation (assuming rates are deterministic).

Differential equations from arbitrary PEPA models

- 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.

Differential equations from arbitrary PEPA models

- 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.

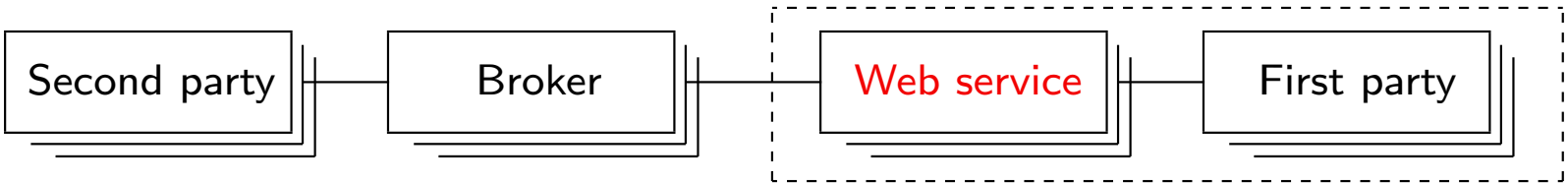
Differential equations from arbitrary PEPA models

- 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.

Derivation of the system of ODES representing the PEPA model then proceeds via an activity matrix in much the same way as for the high/low biochemical pathway model.

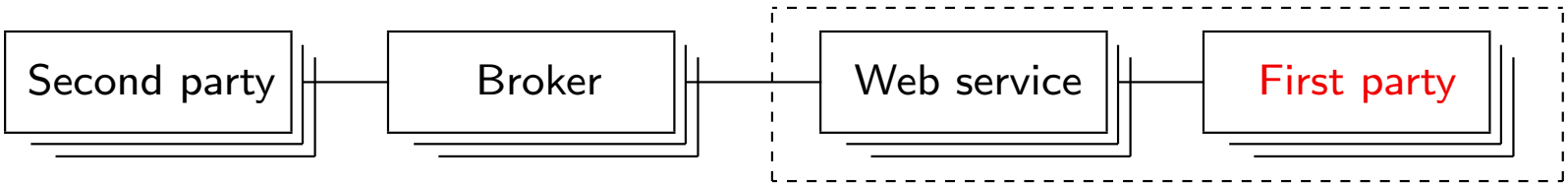
Outline

Example: Secure Web Service use



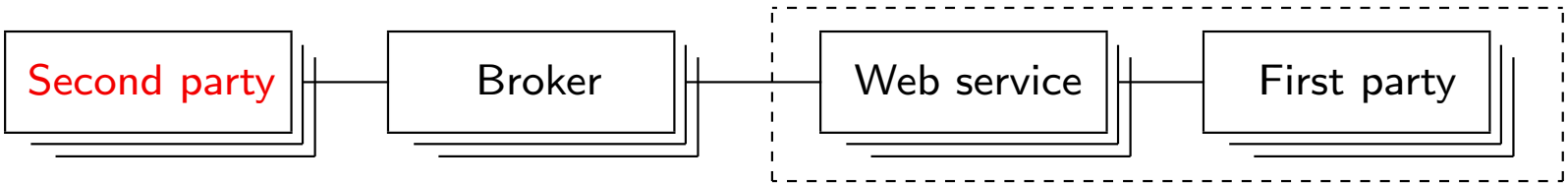
- The example which we consider is a Web service which has two types of clients:
  - first party application clients which access the web service across a secure intranet, and
  - second party browser clients which access the Web service across the Internet.
- Second party clients route their service requests via trusted brokers.

Example: Secure Web Service use



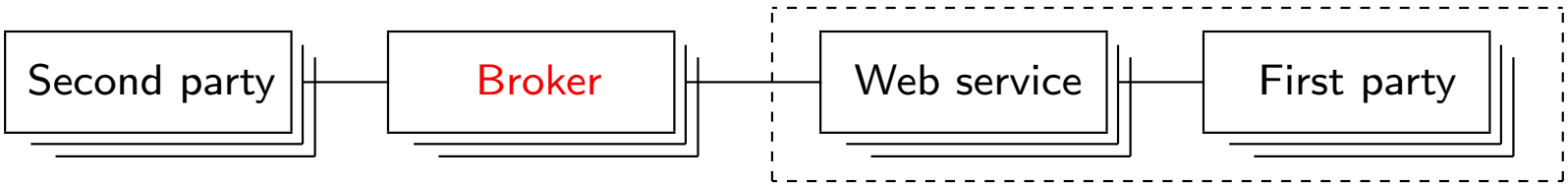
- The example which we consider is a Web service which has two types of clients:
  - first party application clients which access the web service across a secure intranet, and
  - second party browser clients which access the Web service across the Internet.
- Second party clients route their service requests via trusted brokers.

Example: Secure Web Service use



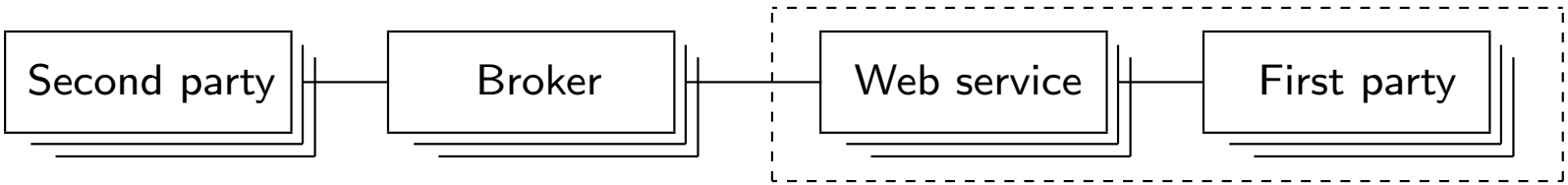
- The example which we consider is a Web service which has two types of clients:
  - first party application clients which access the web service across a secure intranet, and
  - second party browser clients which access the Web service across the Internet.
- Second party clients route their service requests via trusted brokers.

Example: Secure Web Service use



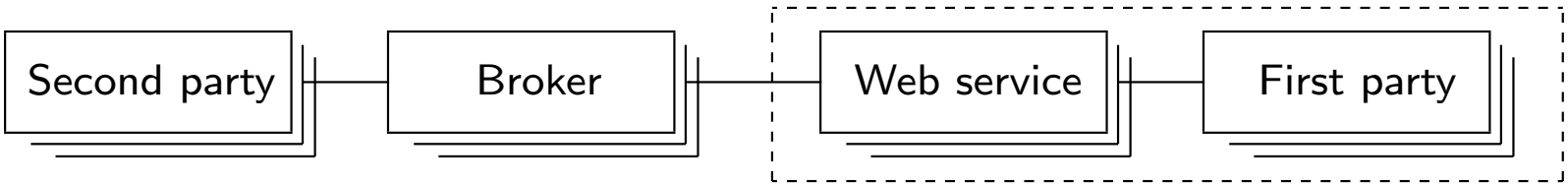
- The example which we consider is a Web service which has two types of clients:
  - first party application clients which access the web service across a secure intranet, and
  - second party browser clients which access the Web service across the Internet.
- Second party clients route their service requests via trusted brokers.

Scalability and replication



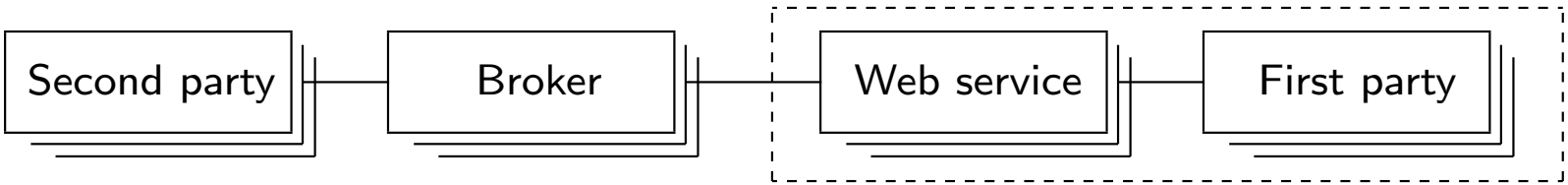
- To ensure scalability the Web service is replicated across multiple hosts.

Scalability and replication



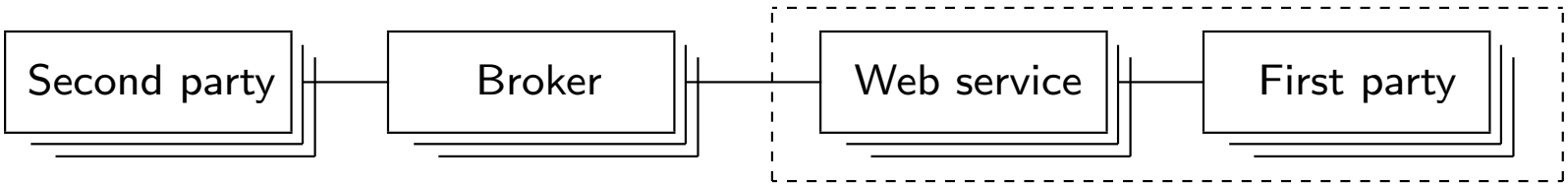
- To ensure scalability the Web service is replicated across multiple hosts.
- Multiple brokers are available.

Scalability and replication



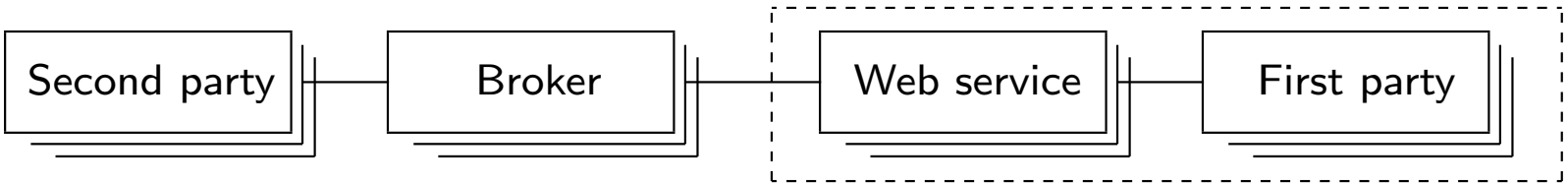
- To ensure scalability the Web service is replicated across multiple hosts.
- Multiple brokers are available.
- There are numerous first party clients behind the firewall using the service via remote method invocations across the secure intranet.

Scalability and replication



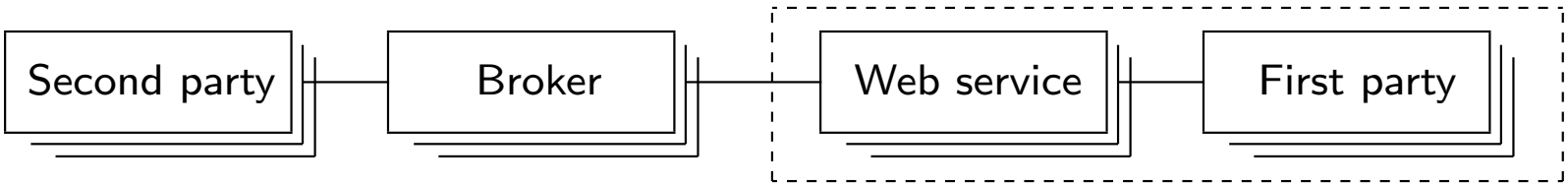
- To ensure scalability the Web service is replicated across multiple hosts.
- Multiple brokers are available.
- There are numerous first party clients behind the firewall using the service via remote method invocations across the secure intranet.
- There are numerous second party clients outside the firewall.

Security and use of encryption



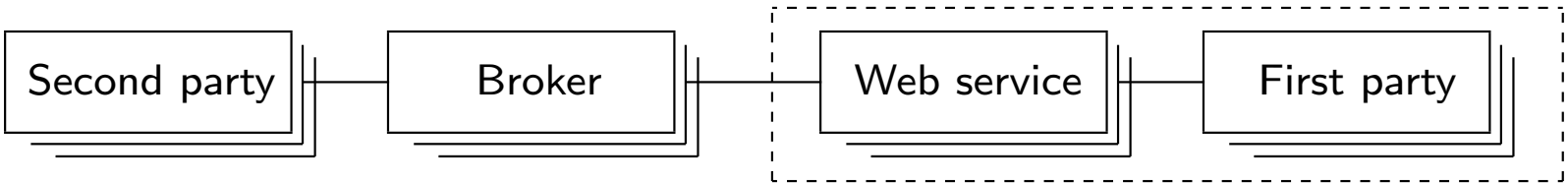
- Second party clients need to use encryption to ensure authenticity and confidentiality. First party clients do not.

Security and use of encryption



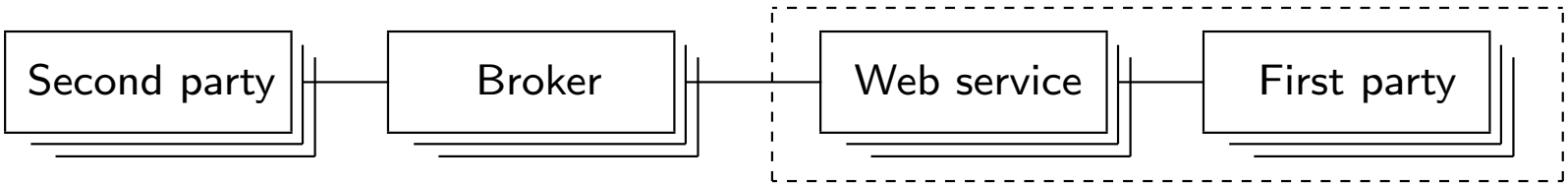
- Second party clients need to use encryption to ensure authenticity and confidentiality. First party clients do not.
- Brokers add decryption and encryption steps to build end-to-end security from point-to-point security.

Security and use of encryption



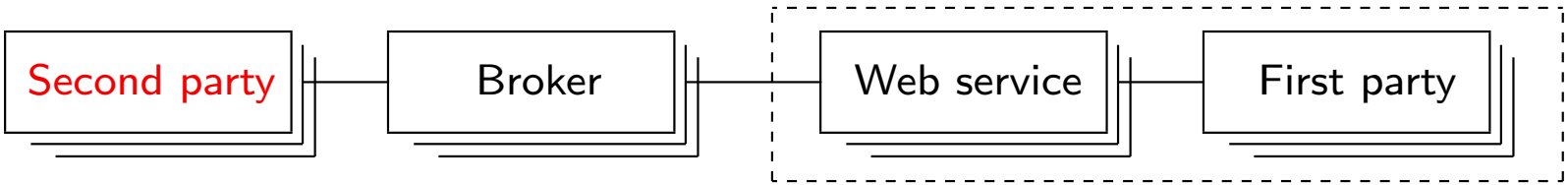
- Second party clients need to use encryption to ensure authenticity and confidentiality. First party clients do not.
- Brokers add decryption and encryption steps to build end-to-end security from point-to-point security.
  - When processing a request from a second party client brokers decrypt the request before re-encrypting it for the Web service.

Security and use of encryption



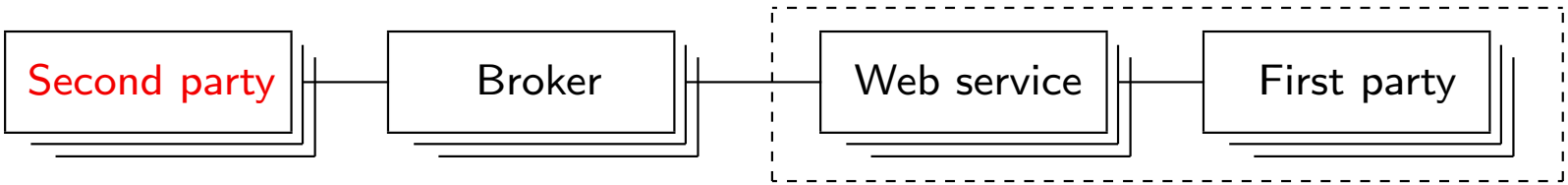
- Second party clients need to use encryption to ensure authenticity and confidentiality. First party clients do not.
- Brokers add decryption and encryption steps to build end-to-end security from point-to-point security.
  - When processing a request from a second party client brokers decrypt the request before re-encrypting it for the Web service.
  - When the response to a request is returned to the broker it decrypts the response before re-encrypting it for the client.

PEPA model: Second party clients



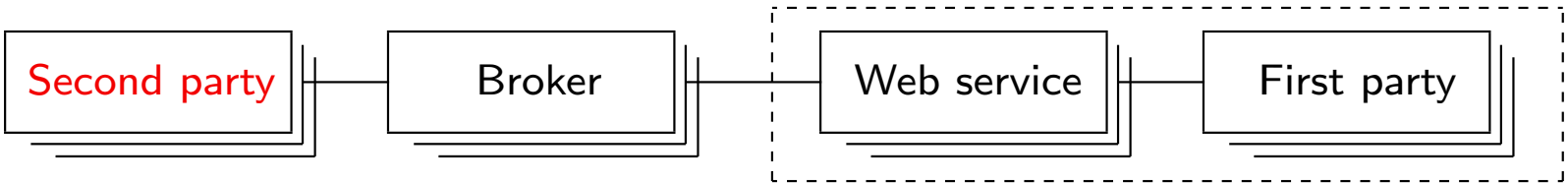
- A second party client composes service requests, encrypts these and sends them to its broker.

PEPA model: Second party clients



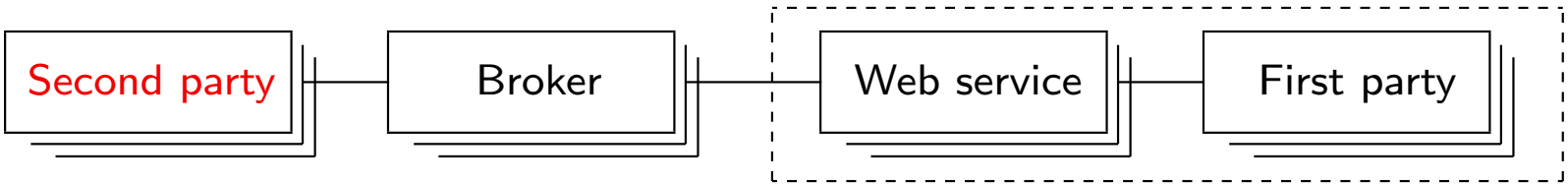
- A second party client composes service requests, encrypts these and sends them to its broker.
- It then waits for a response from the broker.

PEPA model: Second party clients



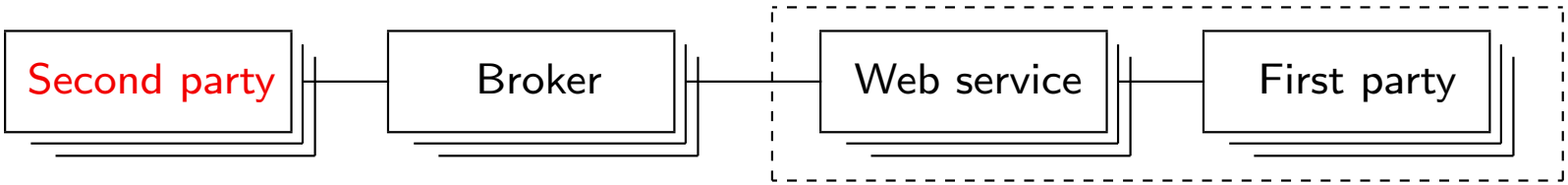
- A second party client composes service requests, encrypts these and sends them to its broker.
- It then waits for a response from the broker.
- The rate at which the first three activities happen is under the control of the client.

PEPA model: Second party clients



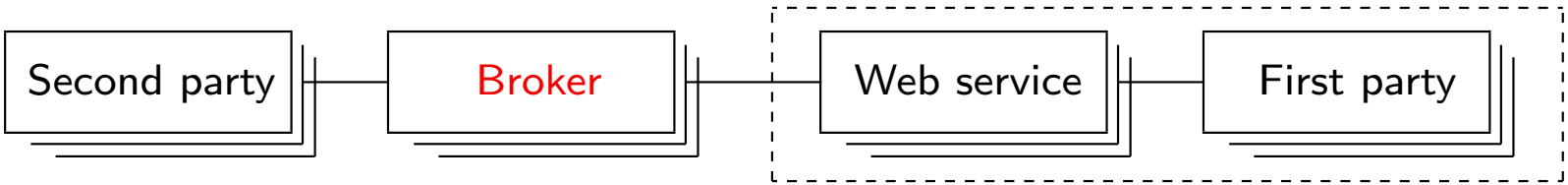
- A second party client composes service requests, encrypts these and sends them to its broker.
- It then waits for a response from the broker.
- The rate at which the first three activities happen is under the control of the client.
- The rate at which responses are produced is determined by the interaction of the broker and the service endpoint.

PEPA model: Second party clients



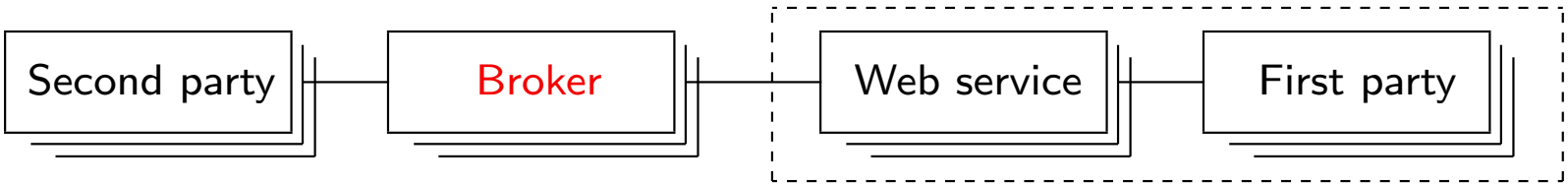
$$
\begin{aligned}
SPC_{idle} &\stackrel{def}{=} (compose_{sp}, r_{sp\_cmp}).SPC_{enc} \\
SPC_{enc} &\stackrel{def}{=} (encrypt_b, r_{sp\_encb}).SPC_{sending} \\
SPC_{sending} &\stackrel{def}{=} (request_b, r_{sp\_req}).SPC_{waiting} \\
SPC_{waiting} &\stackrel{def}{=} (response_b, \top).SPC_{dec} \\
SPC_{dec} &\stackrel{def}{=} (decrypt_b, r_{sp\_decb}).SPC_{idle}
\end{aligned}
$$

PEPA model: Brokers



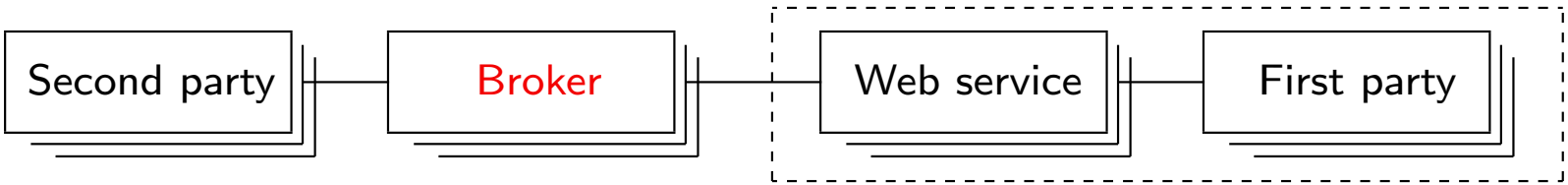
- The broker is inactive until it receives a request.

PEPA model: Brokers



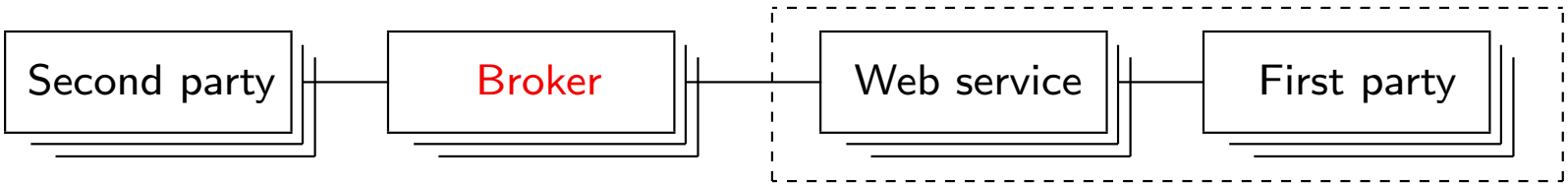
- The broker is inactive until it receives a request.
- It then decrypts the request before re-encrypting it for the Web service to ensure end-to-end security.

PEPA model: Brokers



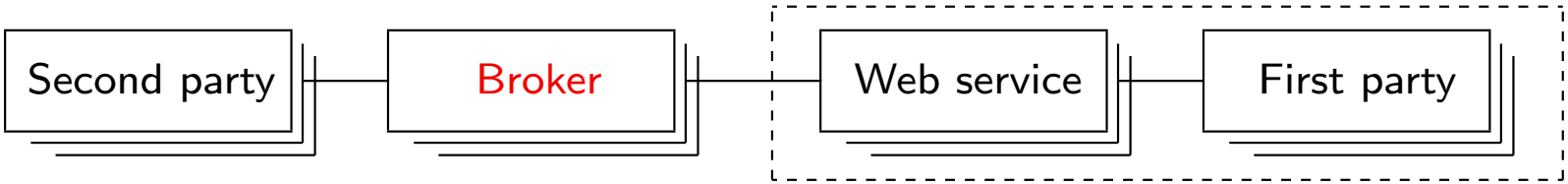
- The broker is inactive until it receives a request.
- It then decrypts the request before re-encrypting it for the Web service to ensure end-to-end security.
- It forwards the request to the Web service and then waits for a response.

PEPA model: Brokers



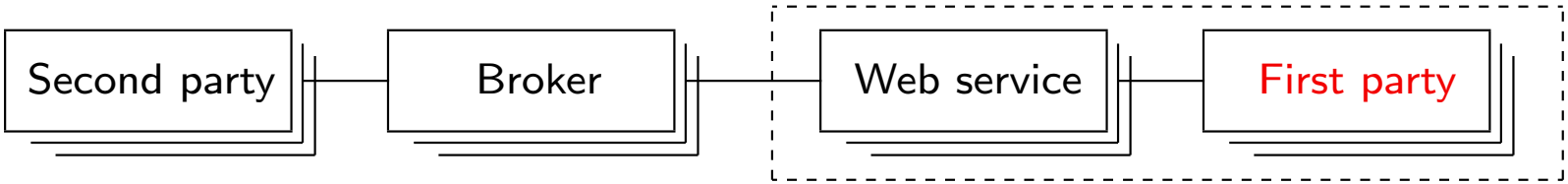
- The broker is inactive until it receives a request.
- It then decrypts the request before re-encrypting it for the Web service to ensure end-to-end security.
- It forwards the request to the Web service and then waits for a response.
- The corresponding decryption and re-encrytion are performed before returning the response to the client.

PEPA model: Brokers



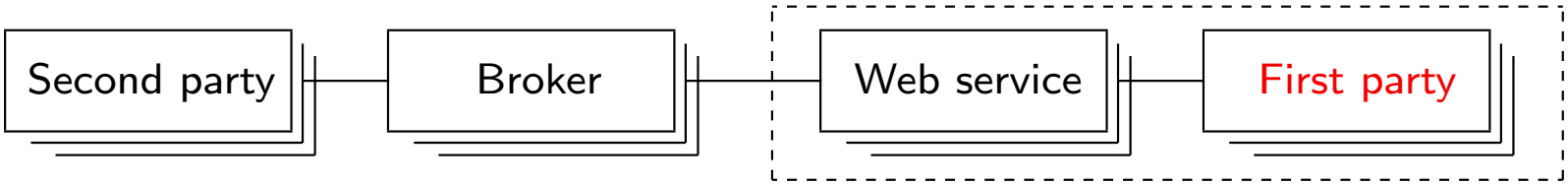
$$
\begin{aligned}
Broker_{idle} &\stackrel{def}{=} (request_b, \top).Broker_{dec\_input} \\
Broker_{dec\_input} &\stackrel{def}{=} (decrypt_{sp}, r_{b\_dec\_sp}).Broker_{enc\_input} \\
Broker_{enc\_input} &\stackrel{def}{=} (encrypt_{ws}, r_{b\_enc\_ws}).Broker_{sending} \\
Broker_{sending} &\stackrel{def}{=} (request_{ws}, r_{b\_req}).Broker_{waiting} \\
Broker_{waiting} &\stackrel{def}{=} (response_{ws}, \top).Broker_{dec\_resp} \\
Broker_{dec\_resp} &\stackrel{def}{=} (decrypt_{ws}, r_{b\_dec\_ws}).Broker_{enc\_resp} \\
Broker_{enc\_resp} &\stackrel{def}{=} (encrypt_{sp}, r_{b\_enc\_sp}).Broker_{replying} \\
Broker_{replying} &\stackrel{def}{=} (response_b, r_{b\_resp}).Broker_{idle}
\end{aligned}
$$

PEPA model: First party clients



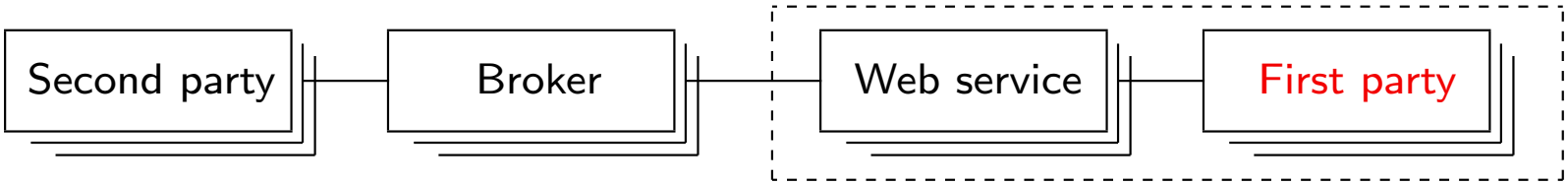
- The lifetime of a first party client mirrors that of a second party client except that encryption need not be used when all of the communication is conducted across a secure intranet.

PEPA model: First party clients



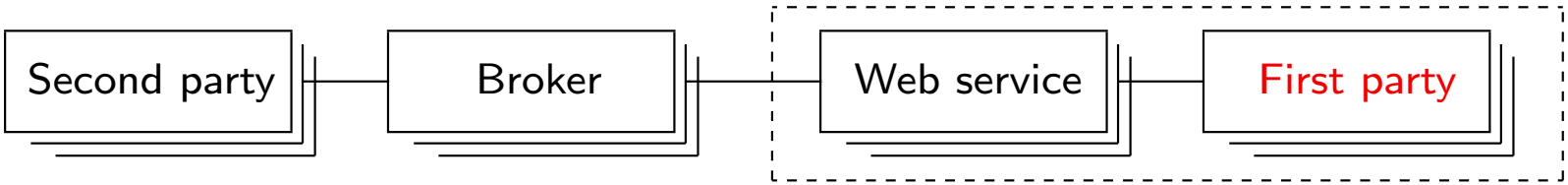
- The lifetime of a first party client mirrors that of a second party client except that encryption need not be used when all of the communication is conducted across a secure intranet.
- Also the service may be invoked by a remote method invocation to the host machine instead of via HTTP.

PEPA model: First party clients



- The lifetime of a first party client mirrors that of a second party client except that encryption need not be used when all of the communication is conducted across a secure intranet.
- Also the service may be invoked by a remote method invocation to the host machine instead of via HTTP.
- Thus the first party client experiences the Web service as a blocking remote method invocation.

PEPA model: First party clients

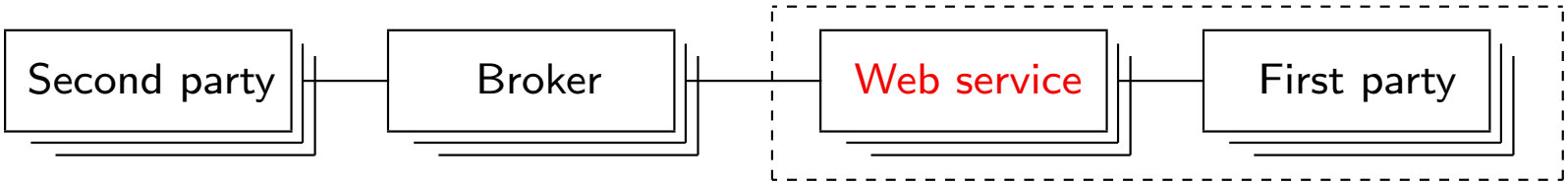


$$FPC_{idle} \stackrel{def}{=} (compose_{fp}, r_{fp\_cmp}).FPC_{calling}$$

$$FPC_{calling} \stackrel{def}{=} (invoke_{ws}, r_{fp\_inv}).FPC_{blocked}$$

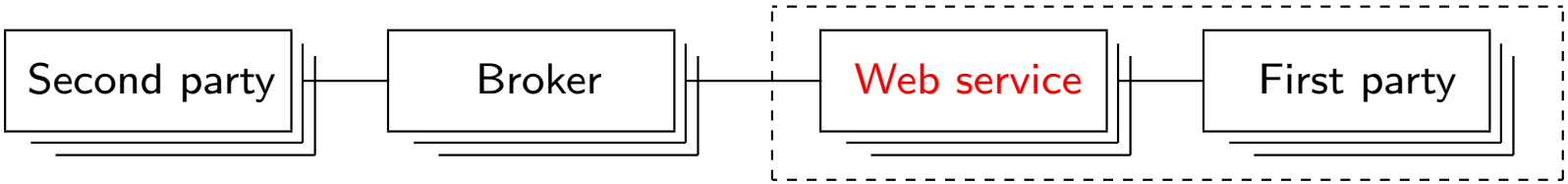
$$FPC_{blocked} \stackrel{def}{=} (result_{ws}, \top).FPC_{idle}$$

PEPA model: Web service



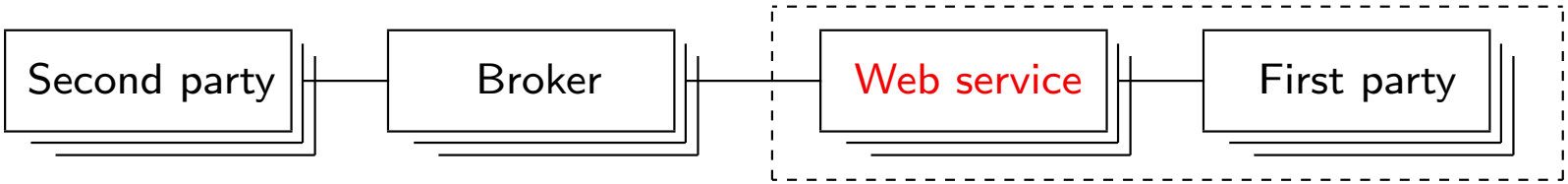
- There are two ways in which the service is executed, leading to a choice in the process algebra model taking the service process into one or other of its two modes of execution.

PEPA model: Web service



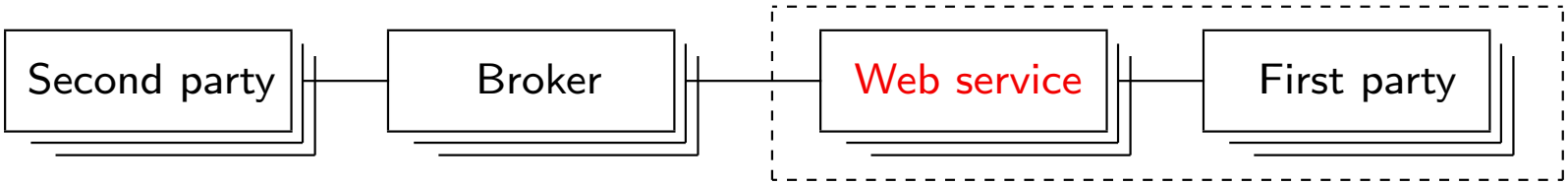
- There are two ways in which the service is executed, leading to a choice in the process algebra model taking the service process into one or other of its two modes of execution.
- In either case, the duration of the execution of the service itself is unchanged.

PEPA model: Web service



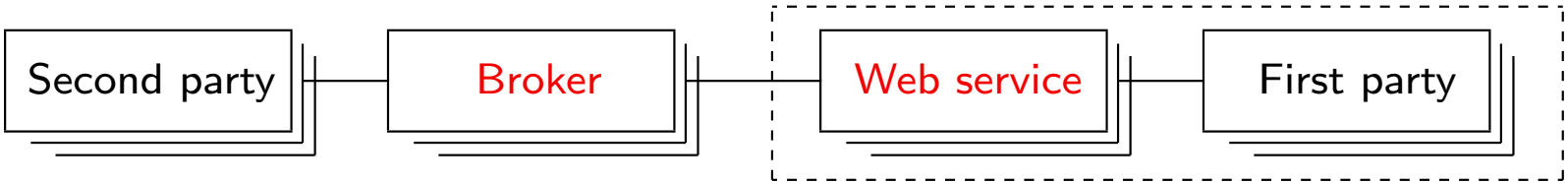
- There are two ways in which the service is executed, leading to a choice in the process algebra model taking the service process into one or other of its two modes of execution.
- In either case, the duration of the execution of the service itself is unchanged.
- The difference is only in whether encryption is needed and whether the result is delivered via HTTP or not.

PEPA model: Web service



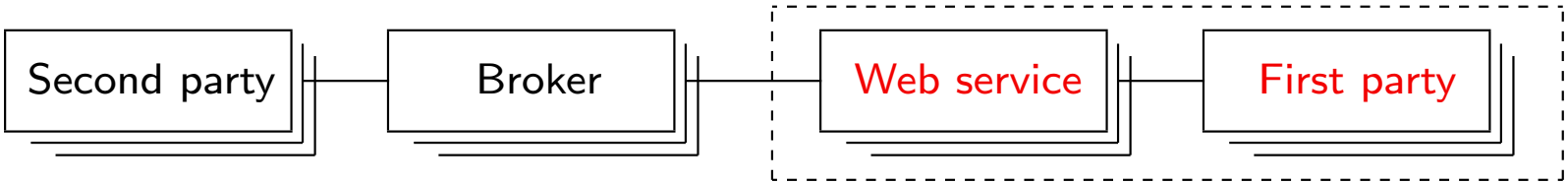
$$
\begin{aligned}
WS_{idle} &\stackrel{def}{=} (request_{ws}, \top).WS_{decoding} \\
&\quad + \; (invoke_{ws}, \top).WS_{method} \\
WS_{decoding} &\stackrel{def}{=} (decryptReq_{ws}, r_{ws\_dec\_b}).WS_{execution} \\
WS_{execution} &\stackrel{def}{=} (execute_{ws}, r_{ws\_exec}).WS_{securing} \\
WS_{securing} &\stackrel{def}{=} (encryptResp_{ws}, r_{ws\_enc\_b}).WS_{responding} \\
WS_{responding} &\stackrel{def}{=} (response_{ws}, r_{ws\_resp\_b}).WS_{idle} \\
WS_{method} &\stackrel{def}{=} (execute_{ws}, r_{ws\_exec}).WS_{returning} \\
WS_{returning} &\stackrel{def}{=} (result_{ws}, r_{ws\_res}).WS_{idle}
\end{aligned}
$$

PEPA model: Web service



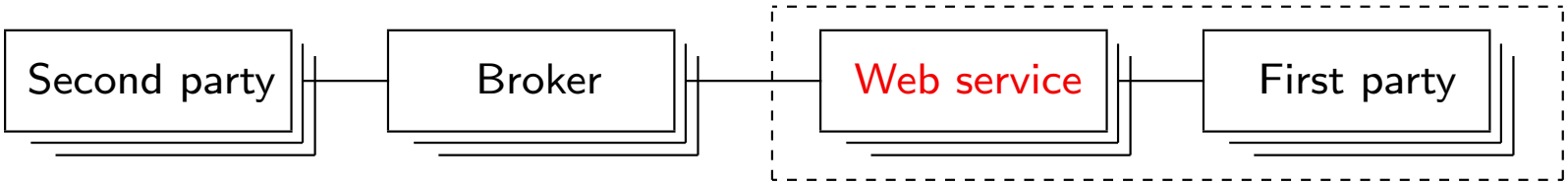
$$
\begin{aligned}
WS_{idle} &\stackrel{def}{=} (request_{ws}, \top).WS_{decoding} \\
&\quad + \; (invoke_{ws}, \top).WS_{method} \\
WS_{decoding} &\stackrel{def}{=} (decryptReq_{ws}, r_{ws\_dec\_b}).WS_{execution} \\
WS_{execution} &\stackrel{def}{=} (execute_{ws}, r_{ws\_exec}).WS_{securing} \\
WS_{securing} &\stackrel{def}{=} (encryptResp_{ws}, r_{ws\_enc\_b}).WS_{responding} \\
WS_{responding} &\stackrel{def}{=} (response_{ws}, r_{ws\_resp\_b}).WS_{idle} \\
WS_{method} &\stackrel{def}{=} (execute_{ws}, r_{ws\_exec}).WS_{returning} \\
WS_{returning} &\stackrel{def}{=} (result_{ws}, r_{ws\_res}).WS_{idle}
\end{aligned}
$$

PEPA model: Web service



$$
\begin{aligned}
WS_{idle} &\stackrel{def}{=} (request_{ws}, \top).WS_{decoding} \\
&\quad + (invoke_{ws}, \top).WS_{method} \\
WS_{decoding} &\stackrel{def}{=} (decryptReq_{ws}, r_{ws\_dec\_b}).WS_{execution} \\
WS_{execution} &\stackrel{def}{=} (execute_{ws}, r_{ws\_exec}).WS_{securing} \\
WS_{securing} &\stackrel{def}{=} (encryptResp_{ws}, r_{ws\_enc\_b}).WS_{responding} \\
WS_{responding} &\stackrel{def}{=} (response_{ws}, r_{ws\_resp\_b}).WS_{idle} \\
WS_{method} &\stackrel{def}{=} (execute_{ws}, r_{ws\_exec}).WS_{returning} \\
WS_{returning} &\stackrel{def}{=} (result_{ws}, r_{ws\_res}).WS_{idle}
\end{aligned}
$$

PEPA model: Web service



$$
\begin{aligned}
WS_{idle} &\stackrel{def}{=} (request_{ws}, \top).WS_{decoding} \\
&\quad + \; (invoke_{ws}, \top).WS_{method} \\
WS_{decoding} &\stackrel{def}{=} (decryptReq_{ws}, r_{ws\_dec\_b}).WS_{execution} \\
WS_{execution} &\stackrel{def}{=} (execute_{ws}, r_{ws\_exec}).WS_{securing} \\
WS_{securing} &\stackrel{def}{=} (encryptResp_{ws}, r_{ws\_enc\_b}).WS_{responding} \\
WS_{responding} &\stackrel{def}{=} (response_{ws}, r_{ws\_resp\_b}).WS_{idle} \\
WS_{method} &\stackrel{def}{=} (execute_{ws}, r_{ws\_exec}).WS_{returning} \\
WS_{returning} &\stackrel{def}{=} (result_{ws}, r_{ws\_res}).WS_{idle}
\end{aligned}
$$

PEPA model: System composition

In the initial state of the system model we represent each of the four component types being initially in their idle state.

$$System \stackrel{def}{=} (SPC_{idle} \underset{\mathcal{K}}{\bowtie} Broker_{idle}) \underset{\mathcal{L}}{\bowtie} (WS_{idle} \underset{\mathcal{M}}{\bowtie} FPC_{idle})$$

$$\begin{array}{lrcl} \text{where} & \mathcal{K} & = & \{ request_b, response_b \} \\ & \mathcal{L} & = & \{ request_{ws}, response_{ws} \} \\ & \mathcal{M} & = & \{ invoke_{ws}, result_{ws} \} \end{array}$$

PEPA model: System composition

In the initial state of the system model we represent each of the four component types being initially in their idle state.

$$System \stackrel{def}{=} (SPC_{idle} \underset{\mathcal{K}}{\bowtie} Broker_{idle}) \underset{\mathcal{L}}{\bowtie} (WS_{idle} \underset{\mathcal{M}}{\bowtie} FPC_{idle})$$

$$\begin{array}{lrcl} \text{where} & \mathcal{K} & = & \{ request_b, response_b \} \\ & \mathcal{L} & = & \{ request_{ws}, response_{ws} \} \\ & \mathcal{M} & = & \{ invoke_{ws}, result_{ws} \} \end{array}$$

This model represents the smallest possible instance of the system, where there is one instance of each component type. We evaluate the system as the number of clients, brokers, and copies of the service increase.

Cost of analysis

- Performance models admit many different types of analysis. Some have lower evaluation cost, but are less informative, such as steady-state analysis. Others have higher evaluation cost, but are more informative, such as transient analysis.

Cost of analysis

- Performance models admit many different types of analysis. Some have lower evaluation cost, but are less informative, such as steady-state analysis. Others have higher evaluation cost, but are more informative, such as transient analysis.
- We compare ODE-based evaluation against other techniques which could be used to analyse the model.

Cost of analysis

- Performance models admit many different types of analysis. Some have lower evaluation cost, but are less informative, such as steady-state analysis. Others have higher evaluation cost, but are more informative, such as transient analysis.
- We compare ODE-based evaluation against other techniques which could be used to analyse the model.
- We compare against steady-state and transient analysis as implemented by the PRISM probabilistic model-checker, which provides PEPA as one of its input languages. We also compare against Monte Carlo Markov Chain simulation.

Comparison of analysis types

- We report only a single run of the transient analysis and simulation. In practice, due to the stochastic nature of the analyses, these would need to be re-run multiple times to produce results comparable to the ODE-based analysis.

Comparison of analysis types

- We report only a single run of the transient analysis and simulation. In practice, due to the stochastic nature of the analyses, these would need to be re-run multiple times to produce results comparable to the ODE-based analysis.
- Moreover, note that the number of ODEs is constant regardless of the number of components in the system, whilst the state space grows dramatically.

Running times from analyses (in seconds)

Second party clients	Brokers	Web service instances	First party clients	Number of states in the full state-space	Number of states in the aggregated state-space	Sparse matrix steady-state	Matrix/MTBDD steady-state	Transient solution for time $t = 100$	MCMC simulation one run to $t = 100$	ODE solution
1	1	1	1	48	48	1.04	1.10	1.01	2.47	2.81

Running times from analyses (in seconds)

Second party clients	Brokers	Web service instances	First party clients	Number of states in the full state-space	Number of states in the aggregated state-space	Sparse matrix steady-state	Matrix/MTBDD steady-state	Transient solution for time $t = 100$	MCMC simulation one run to $t = 100$	ODE solution
1	1	1	1	48	48	1.04	1.10	1.01	2.47	2.81
2	2	2	2	6,304	860	2.15	2.26	2.31	2.45	2.81

Running times from analyses (in seconds)

Second party clients	Brokers	Web service instances	First party clients	Number of states in the full state-space	Number of states in the aggregated state-space	Sparse matrix steady-state	Matrix/MTBDD steady-state	Transient solution for time $t = 100$	MCMC simulation one run to $t = 100$	ODE solution
1	1	1	1	48	48	1.04	1.10	1.01	2.47	2.81
2	2	2	2	6,304	860	2.15	2.26	2.31	2.45	2.81
3	3	3	3	1,130,496	161,296	172.48	255.48	588.80	2.48	2.83

Running times from analyses (in seconds)

Second party clients	Brokers	Web service instances	First party clients	Number of states in the full state-space	Number of states in the aggregated state-space	Sparse matrix steady-state	Matrix/MTBDD steady-state	Transient solution for time $t = 100$	MCMC simulation one run to $t = 100$	ODE solution
1	1	1	1	48	48	1.04	1.10	1.01	2.47	2.81
2	2	2	2	6,304	860	2.15	2.26	2.31	2.45	2.81
3	3	3	3	1,130,496	161,296	172.48	255.48	588.80	2.48	2.83
4	4	4	4	>234M	–	–	–	–	2.44	2.85

Running times from analyses (in seconds)

Second party clients	Brokers	Web service instances	First party clients	Number of states in the full state-space	Number of states in the aggregated state-space	Sparse matrix steady-state	Matrix/MTBDD steady-state	Transient solution for time $t = 100$	MCMC simulation one run to $t = 100$	ODE solution
1	1	1	1	48	48	1.04	1.10	1.01	2.47	2.81
2	2	2	2	6,304	860	2.15	2.26	2.31	2.45	2.81
3	3	3	3	1,130,496	161,296	172.48	255.48	588.80	2.48	2.83
4	4	4	4	>234M	–	–	–	–	2.44	2.85
100	100	100	100	–	–	–	–	–	2.78	2.78

Running times from analyses (in seconds)

Second party clients	Brokers	Web service instances	First party clients	Number of states in the full state-space	Number of states in the aggregated state-space	Sparse matrix steady-state	Matrix/MTBDD steady-state	Transient solution for time $t = 100$	MCMC simulation one run to $t = 100$	ODE solution
1	1	1	1	48	48	1.04	1.10	1.01	2.47	2.81
2	2	2	2	6,304	860	2.15	2.26	2.31	2.45	2.81
3	3	3	3	1,130,496	161,296	172.48	255.48	588.80	2.48	2.83
4	4	4	4	>234M	–	–	–	–	2.44	2.85
100	100	100	100	–	–	–	–	–	2.78	2.78
1000	100	500	1000	–	–	–	–	–	3.72	2.77

Running times from analyses (in seconds)

Second party clients	Brokers	Web service instances	First party clients	Number of states in the full state-space	Number of states in the aggregated state-space	Sparse matrix steady-state	Matrix/MTBDD steady-state	Transient solution for time $t = 100$	MCMC simulation one run to $t = 100$	ODE solution
1	1	1	1	48	48	1.04	1.10	1.01	2.47	2.81
2	2	2	2	6,304	860	2.15	2.26	2.31	2.45	2.81
3	3	3	3	1,130,496	161,296	172.48	255.48	588.80	2.48	2.83
4	4	4	4	>234M	–	–	–	–	2.44	2.85
100	100	100	100	–	–	–	–	–	2.78	2.78
1000	100	500	1000	–	–	–	–	–	3.72	2.77
1000	1000	1000	1000	–	–	–	–	–	5.44	2.77

Running times from analyses (in seconds)

Second party clients	Brokers	Web service instances	First party clients	Number of states in the full state-space	Number of states in the aggregated state-space	Sparse matrix steady-state	Matrix/MTBDD steady-state	Transient solution for time $t = 100$	MCMC simulation one run to $t = 100$	ODE solution
1	1	1	1	48	48	1.04	1.10	1.01	2.47	2.81
2	2	2	2	6,304	860	2.15	2.26	2.31	2.45	2.81
3	3	3	3	1,130,496	161,296	172.48	255.48	588.80	2.48	2.83
4	4	4	4	>234M	–	–	–	–	2.44	2.85
100	100	100	100	–	–	–	–	–	2.78	2.78
1000	100	500	1000	–	–	–	–	–	3.72	2.77
1000	1000	1000	1000	–	–	–	–	–	5.44	2.77

Time series analysis via ODEs

- We now consider the results from our solution of the PEPA Web Service model as a system of ODEs with the number of clients of both kinds, brokers, and web service instances all 1000.

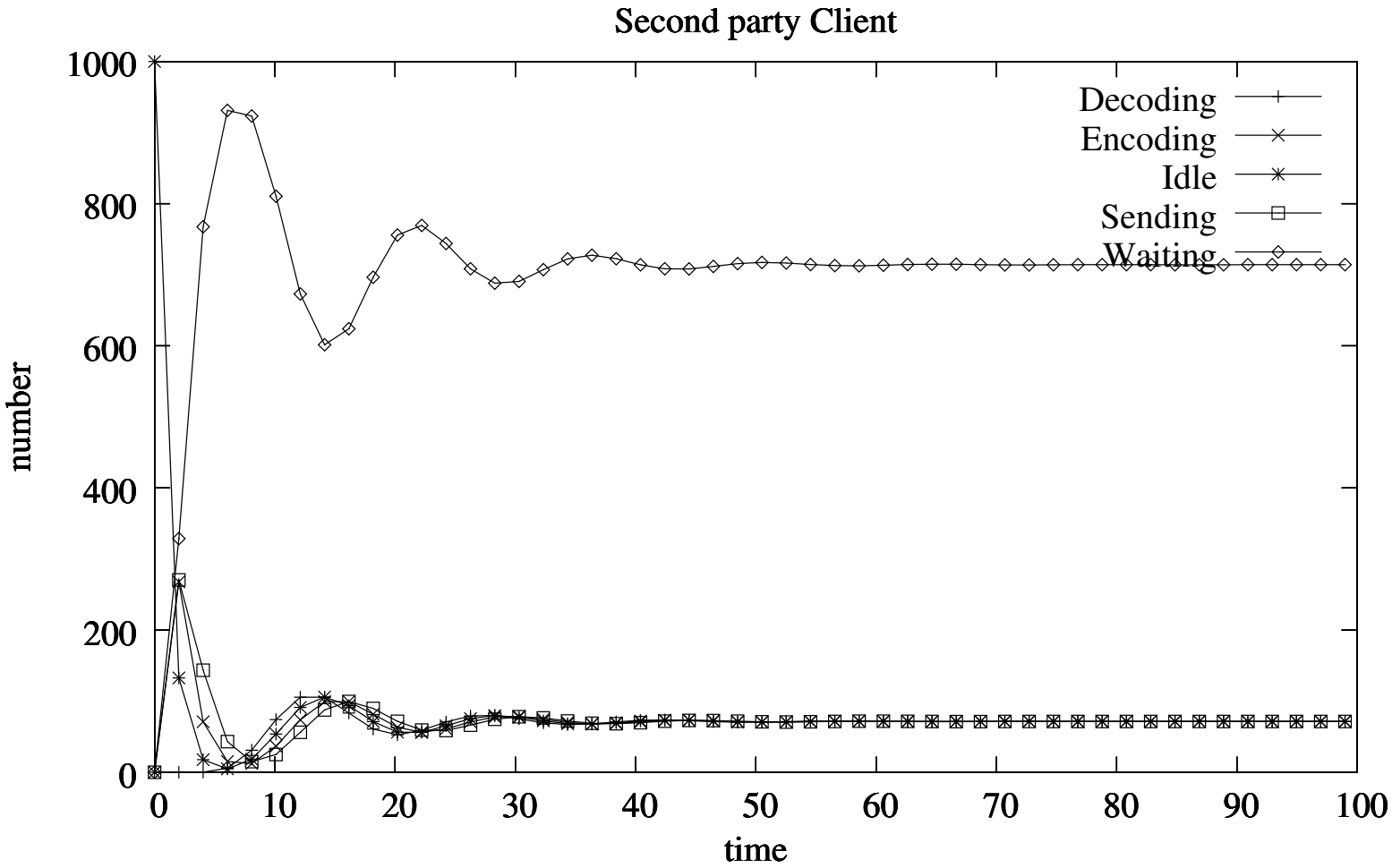
Time series analysis via ODEs

- We now consider the results from our solution of the PEPA Web Service model as a system of ODEs with the number of clients of both kinds, brokers, and web service instances all 1000.
- The results as presented from our ODE integrator are time-series plots of the number of each type of component behaviour as a function of time.

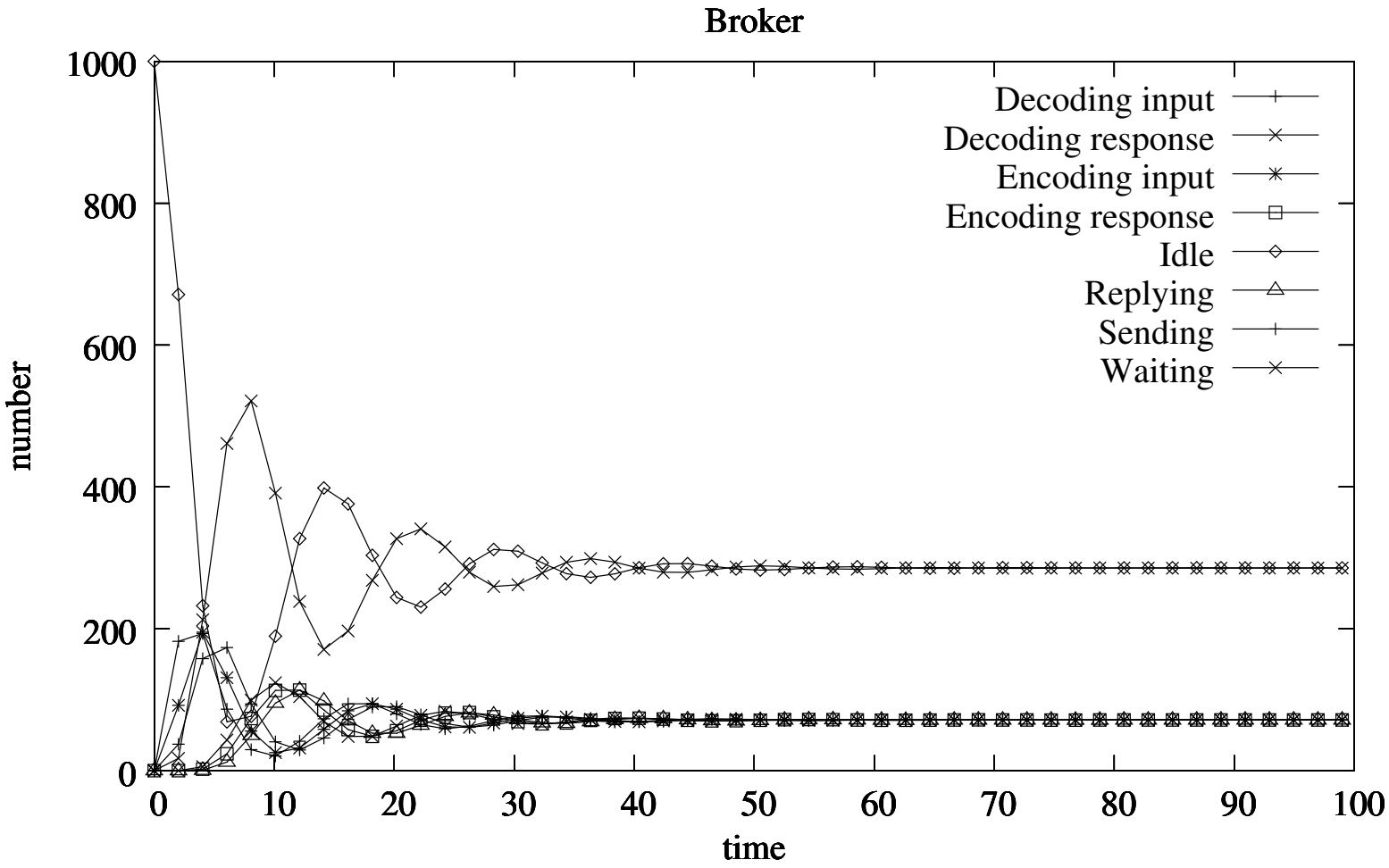
Time series analysis via ODEs

- We now consider the results from our solution of the PEPA Web Service model as a system of ODEs with the number of clients of both kinds, brokers, and web service instances all 1000.
- The results as presented from our ODE integrator are time-series plots of the number of each type of component behaviour as a function of time.
- The graphs show fluctuations in the numbers of components with respect to time from $t = 0$ to $t = 100$ for estimated values of rates for the activities of the system. We can observe an initial flurry of activity until the system stabilises into its steady-state equilibrium at time (around) $t = 50$.

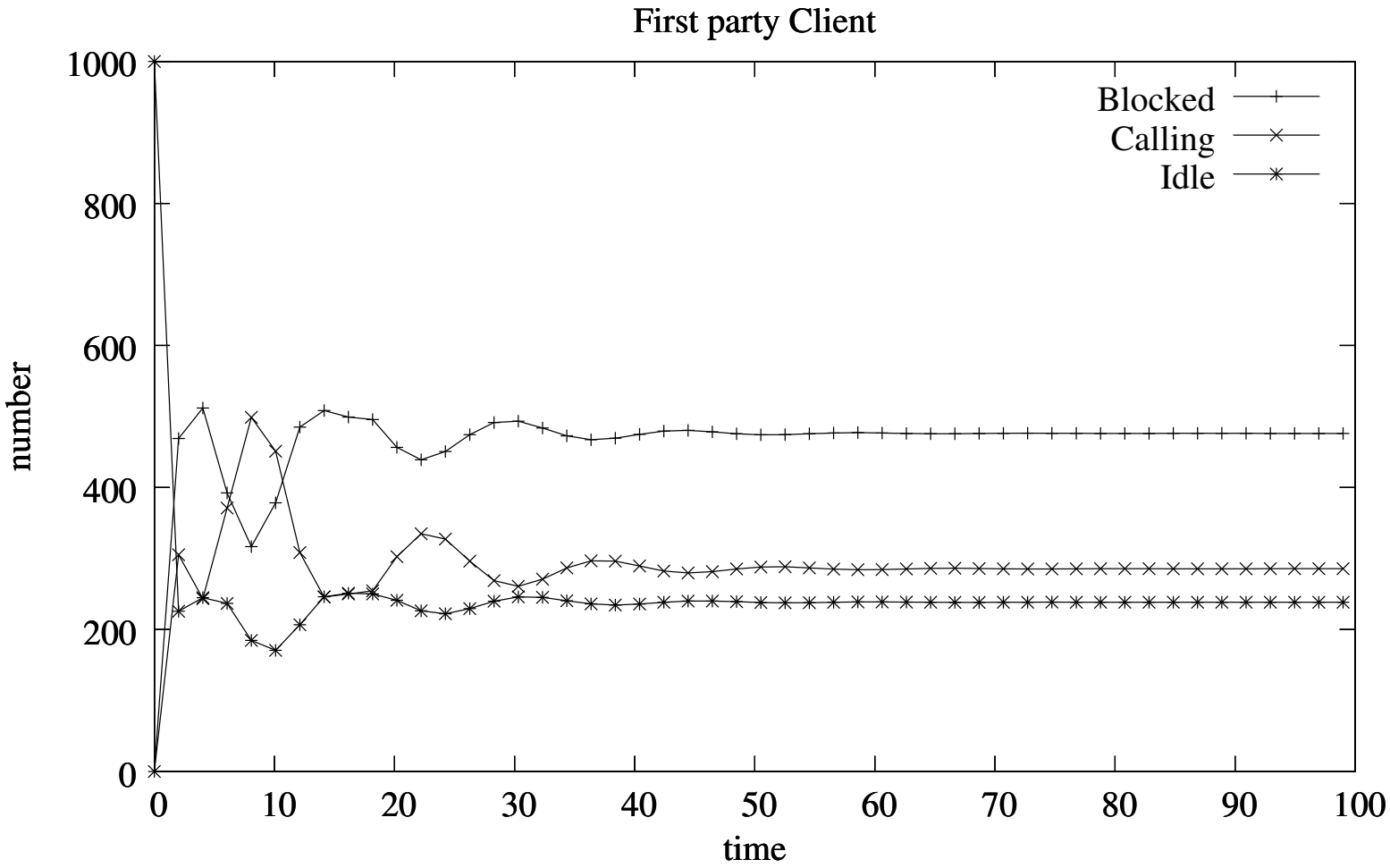
Second party clients



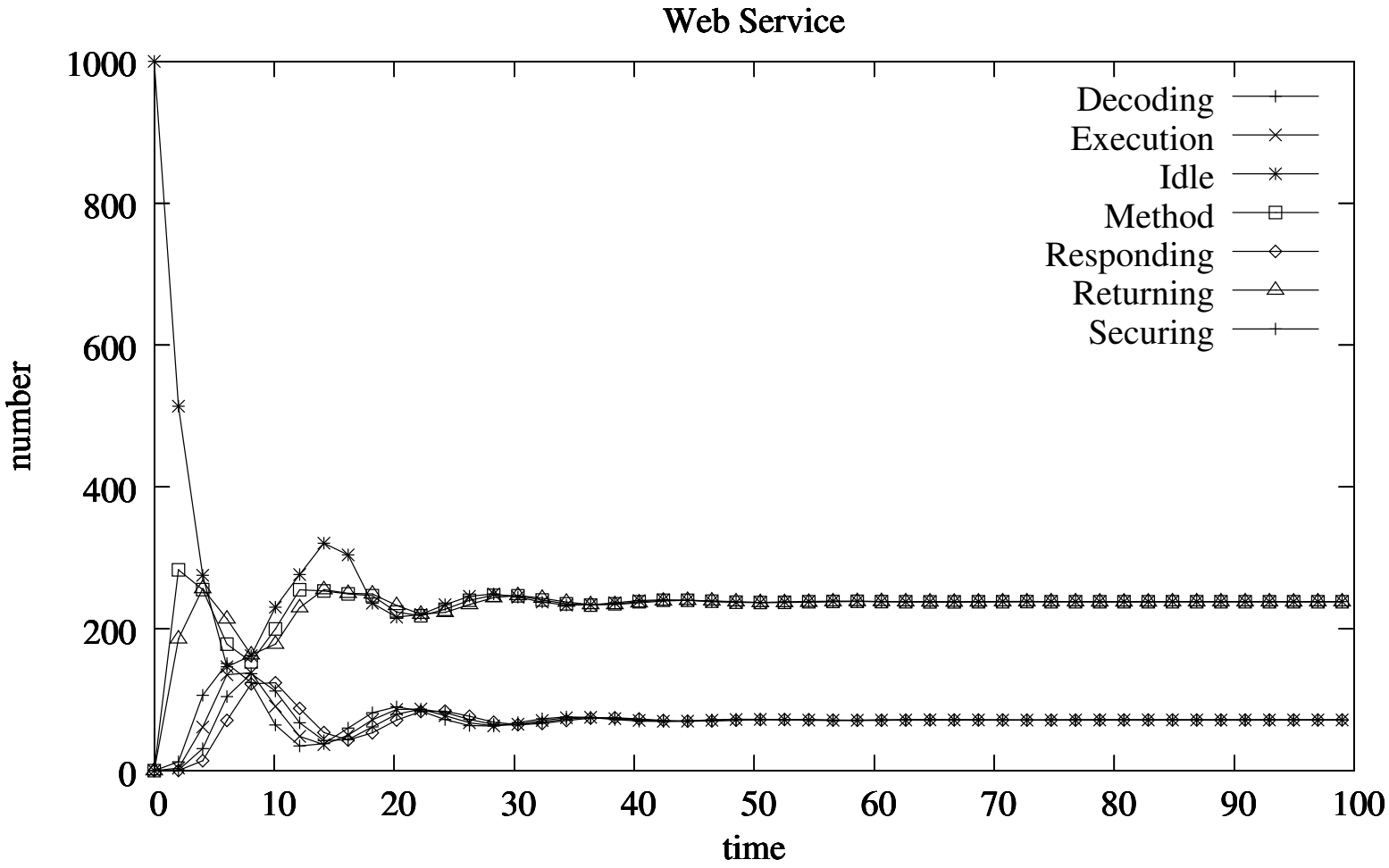
Brokers


Broker
number
time
0
200
400
600
800
1000
0
10
20
30
40
50
60
70
80
90
100
Decoding input
Decoding response
Encoding input
Encoding response
Idle
Replying
Sending
Waiting

First party clients



Web service


Web Service
number
time
0
200
400
600
800
1000
0
10
20
30
40
50
60
70
80
90
100
Decoding
Execution
Idle
Method
Responding
Returning
Securing

Comparison with Continuous-Time Markov Chain solution

- For a system without blocking (1000 instances of all components) we compared the steady-state values of state variables with values derived from the the steady-state probabilities computed by the PEPA Workbench for the model with one instance of each component.

Comparison with Continuous-Time Markov Chain solution

- For a system without blocking (1000 instances of all components) we compared the steady-state values of state variables with values derived from the the steady-state probabilities computed by the PEPA Workbench for the model with one instance of each component.
  - The steady state probabilities of the PEPA Workbench were scaled by 1000.

Comparison with Continuous-Time Markov Chain solution

- For a system without blocking (1000 instances of all components) we compared the steady-state values of state variables with values derived from the the steady-state probabilities computed by the PEPA Workbench for the model with one instance of each component.
  - The steady state probabilities of the PEPA Workbench were scaled by 1000.
- We found good agreement with the results obtained by ODE solution after the system stabilises into its steady-state equilibrium.

Outline

Conclusions

The excursion into systems biology has been very worthwhile:

Conclusions

The excursion into systems biology has been very worthwhile:

- the reaction of the biochemists from the Beatson Institute have been very positive:

Conclusions

The excursion into systems biology has been very worthwhile:

- the reaction of the biochemists from the Beatson Institute have been very positive:
  - the presentation of the system in two contrasting styles, which can be proved to be equivalent;

Conclusions

The excursion into systems biology has been very worthwhile:

- the reaction of the biochemists from the Beatson Institute have been very positive:
  - the presentation of the system in two contrasting styles, which can be proved to be equivalent;
  - the quantified reasoning at the Markov chain level – even with crude approximation of concentrations – giving estimations of transient properties such as throughput over time, and first passage time distributions.

Conclusions

The excursion into systems biology has been very worthwhile:

- the reaction of the biochemists from the Beatson Institute have been very positive:
  - the presentation of the system in two contrasting styles, which can be proved to be equivalent;
  - the quantified reasoning at the Markov chain level – even with crude approximation of concentrations – giving estimations of transient properties such as throughput over time, and first passage time distributions.
- the derivation of differential equations appears to offer an interesting alternative to existing modelling approaches to performance evaluation of large scale models.