PEPA
Performance Evaluation Process Algebra

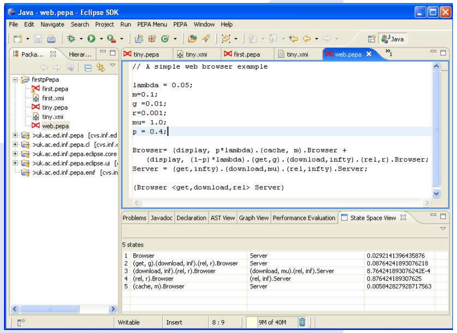
Professor Jane Hillston

Jane.Hillston@ed.ac.uk

http://homepages.inf.ed.ac.uk/jeh/

Quantitative modelling and analysis

The PEPA language is a stochastic process algebra used to analyse both natural and artificial systems such as the biochemical pathways in living organisms and the performance characteristics of computer and communication systems. PEPA is a concise modelling language in the tradition of process algebras such as CCS, designed in Edinburgh by Robin Milner, founder of the Laboratory for Foundations of Computer Science. It enriches the CCS language with timing information about the activities performed by the model, enabling the system under study to be analysed both behaviourally and quantitatively.



Analysis via continuous-time Markov chains provides a thorough stochastic treatment of performance models

PEPA has been successfully applied to many different systems

- including protocols in fault-tolerant systems,
- switch behaviour in active networks,
- robotic workcells,
- locks and movable bridges in inland shipping in Belgium,
- signal transduction pathways for intracellular communication.

The formalism has been used by researchers and practitioners around the world and over 100 papers have been published about PEPA and its use.

Semantics of the language

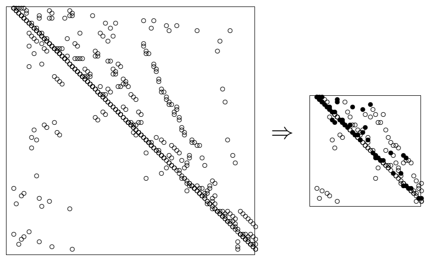
The PEPA language was designed with a small-step structured operation semantics in the style made famous by Gordon Plotkin. The semantics formally mapped PEPA models onto continuous-time Markov chains, bringing the machinery of Markovian analysis to the world of process algebra for the first time.

$$\frac{E \xrightarrow{(\alpha,r_1)} E' \quad F \xrightarrow{(\alpha,r_2)} F'}{E \bowtie_L F \xrightarrow{(\alpha,R)} E' \bowtie_L F'} \qquad (\alpha \in L)$$

where $R = \frac{r_1}{r_\alpha(E)}\frac{r_2}{r_\alpha(F)}\min(r_\alpha(E), r_\alpha(F))$
and $r_\alpha(E)$ is the apparent rate of α in E

Aggregation

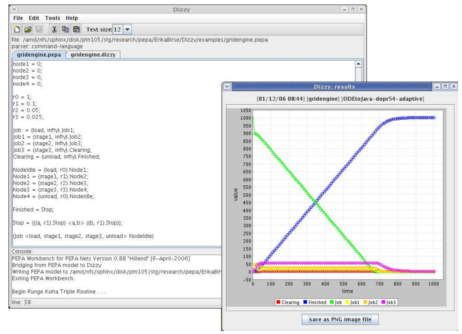
A rich and meaningful connection was found between the world of process algebra and the world of Markov chains. The *bisimulation* relation defined for the PEPA process algebra was proven to correspond with the Markovian notion of *lumpability*, meaning that complex models could be subjected to compositional reduction, allowing larger and more complex models to be analysed.



However, even though they can be reduced in size, Markovian models are still prone to the well-known problem of *state-space explosion*. A new and radical approach was needed in order to tackle much more complex Internet-scale performance problems.

Collective dynamics

Process algebras, and CTMCs, by focusing on the behaviour of individuals, give a fine-grained, discrete state account of a system in terms of the individual entities in the system and their interactions. However, there are some circumstances in which it is sufficient to understand what is happening at a population level, rather than keeping track of each individual, for example for biochemical species or clients in scalability studies. To facilitate such analyses PEPA has recently been equipped with a continuous semantics which maps a PEPA model to a set of coupled nonlinear ordinary differential equations.



System scalability can be assessed by mapping onto ordinary differential equations and applying numerical integration

History of the PEPA project

Jane Hillston defined the PEPA language in her PhD thesis, undertaken in the Laboratory for Foundations of Computer Science at Edinburgh. Her PhD thesis was awarded the British Computer Society/Conference of Professors and Heads of Computing Distinguished Dissertation award in 1995 and is published by Cambridge University Press. In 2004 Prof Hillston won the BCS/Microsoft Roger Needham award for her work on PEPA and compositional approaches to performance modelling. In May 2005 she was awarded a five-year Advanced Research Fellowship from the Engineering and Physical Sciences Research Council to support her research on quantified methods and process algebras.

School of informatics