Population models from PEPA descriptions

Jane Hillston

LFCS, The University of Edinburgh, Edinburgh EH9 3JZ, Scotland.
Email: jeh@inf.ed.ac.uk

1 Introduction

Stochastic process algebras (e.g. PEPA [10], EMPA [1], TIPP [9]) emerged about 15 years ago as system description techniques for performance modelling. They have enjoyed some considerable success in this arena. For example, PEPA has been used to study the performance of a wide variety of systems [12, 2, 3, 18, 13].

This analysis has been based on the generation of a continuous time Markov chain (CTMC) underlying the labelled transition system of the process algebra model, derived via the interleaving structured operational semantics. The CTMC facilitates steady state and transient analysis numerically. From these distributions many performance metrics can be derived, such as utilization, throughput, and mean time to congestion. Unfortunately, as with all state-based modelling techniques, CTMCs, and consequently stochastic process algebras, suffer from problems of *state space explosion*. Such models can be regarded as being at the *individual* level as all details of all the components of the model are recorded in the state of the model and its subsequent analysis.

Although developed for performance modelling of computer and communication systems, stochastic process algebras have proved to be useful for modelling other systems as well. In particular, in recent years there has been considerable interest in using stochastic process algebras and similar formalisms for expressing models for systems biology [16, 17, 4, 6]. These models also exhibit problems of state space explosion, as both the types of components and the number of components of each type are typically large.

In recent work we have been exploring an alternative mapping from a system description in the PEPA stochastic process algebra. In this mapping we aim to capture the behaviour of PEPA components at a *population* level. Rather than capturing individual behaviours as happens in the CTMC semantics, we instead map to a set of non-linear ordinary differential equations (ODEs) [5, 11]. This incurs some loss of information with respect to, for example, the steady state probability distribution of the CTMC. Nevertheless the solution of the set of ODEs can still give us useful quantitative information about the system.

2 Individual vs. Population models

In most performance models the focus is on the individual entities within the model. These individuals may be customers or tasks within the system, as well as

Dagstuhl Seminar Proceedings 06161
Simulation and Verification of Dynamic Systems
http://drops.dagstuhl.de/opus/volltexte/2006/707

the software and/or hardware entities of which the system is comprised. However in many modern systems the number of types of entities is not necessarily very big, while the number of entities of each type is large. This suggests that it may be better to instead view the system in terms of a small number of populations rather than individuals.

Of course making this switch to the population level results in some loss of information. It is no longer possible to precisely record the state of each entity. This information is aggregated into information about the statistics of the state of the population to which the entity belongs. However, for some performance measures at least, this population level view is sufficient. Such an abstract view has previously been adopted in the *diffusion* approximation technique for queueing networks [14, 15].

These alternative styles of model are already available in the systems biology arena:

Stochastic Simulations (Gillespie *et al.* [7]) are individual-based models in which each molecule is treated separately;

Ordinary Differential Equations (ODEs) are population-based models in which populations of molecules are represented abstractly as concentrations.

We view these not as competing approaches but as alternative analysis techniques and we would like access to both whilst also being able to carry out process algebra-based analyses such as model checking. We have modified versions of the PEPA workbench [8] which will extract underlying models amenable for three distinct forms of analysis, representing different interpretations of the PEPA system description:

- CTMC model suitable for numerical solution for steady state and transient analysis or model checking against properties expressed in the CSL modal logic;
- A representation of a CTMC suitable for simulation using Gillespie's stochastic simulation technique;
- A set of ODEs which may be solved numerically using a variety of techniques.

The BioCHAM formalism offers a similar spectrum of interpretations and analysis techniques [6].

3 Deriving the ODEs

In this section we give a brief overview of how a set of ODEs may be derived from a PEPA model. The interested reader is referred to [11] for more details. There are two crucial steps. The first is to establish a more abstract representation of the state of the system than that typically used as the basis for Markovian analysis. The second is to assume that state change is a continuous rather than a discrete process.

When taking the population view of the system, rather than record the local state of each entity (current derivative) and treat the global state as the complete

set of local states, we instead record the local state of each subpopulation (the number of each possible derivative currently exhibited) and treat the global state as the set of subpopulation states. Thus we first identify the distinct entity types within the system and the local states or *derivatives* of each one. Each such derivative is associated with an entry in a *state count vector*. The number in the corresponding place in the vector records the current number of derivatives of that type.

This represents a considerably more compact state representation when there are more repeated copies of components, than there are local derivatives. However this state representation is still inherently discrete. The population view is really obtained when we make the assumption that these state variables are subject to continuous, rather than discrete, change. The evolution of each count variable can then be described by an ODE. Note that there is an implicit assumption being made here that the rate of the activity is deterministic rather than exponentially distributed.

The PEPA definitions of the component specify the activities which can *increase* or *decrease* the number of components exhibited in the current state. For any derivative the *entry activities* are those that bring the component into this state, while the *exit activities* are those enabled in this state and when performed result in a change of derivative. The cooperations show when the number of instances of another component will have an influence on the evolution of this component. This will introduce non-linearity into the set of ODEs.

Let us consider one entry of the state count vector: $N(\mathcal{C}_{i_j}, t)$ denotes the number of $\mathcal{C}_{i_j}$ derivatives at time t.

Consider the change in a small time δt:

$$N(\mathcal{C}_{i_j}, t+\delta t) \; - \; N(\mathcal{C}_{i_j}, t) = \\ \underbrace{- \sum_{(\alpha,r)\in Ex(\mathcal{C}_{i_j})} r \times \min_{\mathcal{C}_{k_l}\in Ex(\alpha,r)} (N(\mathcal{C}_{k_l}, t))\, \delta t}_{\text{exit activities}} \\ \underbrace{+ \sum_{(\alpha,r)\in En(\mathcal{C}_{i_j})} r \times \min_{\mathcal{C}_{k_l}\in Ex(\alpha,r)} (N(\mathcal{C}_{k_l}, t))\, \delta t}_{\text{entry activities}}$$

Dividing by δt and taking the limit, $\delta t \longrightarrow 0$ we obtain:

$$\frac{dN(C_{i_j}, t)}{dt} = - \sum_{(\alpha,r)\in Ex(\mathcal{C}_{i_j})} r \times \min_{\mathcal{C}_{k_l}\in Ex(\alpha,r)} (N(C_{k_l}, t)) \\ + \sum_{(\alpha,r)\in En(\mathcal{C}_{i_j})} r \times \min_{\mathcal{C}_{k_l}\in Ex(\alpha,r)} (N(C_{k_l}, t))$$

In practice we do not wish to derive the system of ODEs representing a PEPA model from first principles in this way. Instead the derivation proceeds

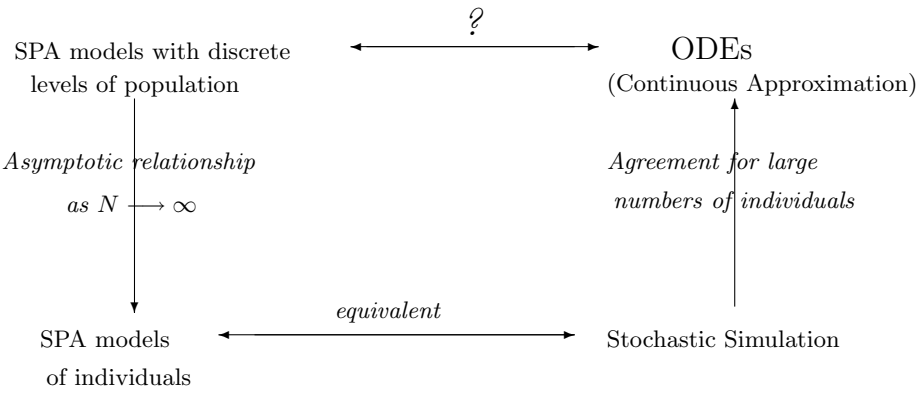

Fig. 1. Alternative representations and the relationships between them

via an intermediate data structure, termed the *activity matrix*, which records the influence of each activity on each component type/derivative. The matrix has one row for each component type and one column for each activity type. One ODE is generated corresponding to each row of the matrix, taking into account the *negative entries* in the non-zero columns as these are the components for which this is an exit activity [11].

4 Conclusions

The use of stochastic process algebra models for Markovian based modelling is well-established, but hampered by the problem of state space explosion. In some models the problems arise because there are many instances of components of the same type. In this case the use of more abstract mathematical models which focus on the statistics of the population rather than the individuals seems promising. We have recently established a mapping to such a mathematical model for the stochastic process algebra PEPA [11]. This is an alternative, not a replacement for existing techniques.

We are currently exploring the relationship between models derived in this way and other models which can be derived from the PEPA description. Our PEPA models of biochemical pathways have been experimenting with different levels of abstraction. Whereas treating each molecule in the system as a distinct entity or component in the model corresponds to the stochastic simulation view of the system, we have developed models with discrete levels of concentration, corresponding to the ODE view [4]. These models can also be interpreted as CTMCs and we are investigating the relationship between the ODE and the CTMC obtained in these cases (see Figure 1).

References

1. M. Bernardo and R. Gorrieri. A tutorial on EMPA: a theory of concurrent processes with n ondet erminism, priorities, probabilities and time. *TCS*, 202:1–54, 1998.
2. H. Bowman, J. Bryans, and J. Derrick. Analysis of a multimedia stream using stochastic process algebra. In C. Priami, editor, *Sixth International Workshop on Process Algebras and Performance Modelling*, pages 51–69, Nice, September 1998.
3. J.T. Bradley, N.J. Dingle, S.T. Gilmore, and W.J. Knottenbelt. Derivation of passage-time densities in PEPA models using ipc: the Imperial PEPA Compiler. In Gabriele Kotsis, editor, *Proc. of the 11th IEEE/ACM Int. Symp. on Modeling, Analysis and Simulation of Computer and Telecommunications Systems*, pages 344–351, University of Central Florida, October 2003. IEEE Computer Society Press.
4. M. Calder, S. Gilmore, and J. Hillston. Modelling the influence of RKIP on the ERK signaling path way using the stochastic process algebra PEPA. In *Proceedings of BioConcur'04*, London, England, August 2004.
5. M. Calder, S. Gilmore, and J. Hillston. Automatically deriving ODEs from process algebra models of signalling pathways. In G. Plotkin, editor, *Proc. of International Workshop on Computational Methods in Systems Biology*, pages 204–215, April 2005.
6. N. Chabrier-Rivier, M. Chiaverini, V. Danos, F. Fages, and V. Schchter. Modeling and querying biomolecular interaction networks. *Theoretical Computer Science*, 325(1):25–44, 2004.
7. D.T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340 –2361, 1977.
8. S. Gilmore and J. Hillston. The PEPA Workbench: A Tool to Support a Process Algebra-based Approach to Performance Modelling. In *Proceedings of the Seventh International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, number 794 in Lecture Notes in Computer Science, pages 353–368, Vienna, May 1994. Springer-Verlag.
9. N. Götz, U. Herzog, and M. Rettelbach. Multiprocessor and Distributed System Design: The Integration of Functional Specification and Performance Analysis using Stochastic Process Algebras. In *Performance'93*, 1993.
10. J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.
11. J. Hillston. Fluid flow approximations of PEPA models. In *Proceedings of the International Conference on Quantitative Evaluation of Systems (QEST) 2005*, pages 33–42, Torino, September 2005. IEEE Computer Society Press.
12. J. Hillston. Tuning systems: From composition to performance. *The Computer Journal*, 48(4):385–400, 2005.
13. J. Hillston, L. Kloul, and A. Mokhtari. Towards a feasible active networking scenario. *Telecommunication Systems*, 27(2–4):413–438, 2004.
14. H. Kobayashi. Application of the diffusion approximation to queueing networks I: Equilibrium queue distributions. *J. ACM*, 21(2):316–328, 1974.
15. H. Kobayashi. Application of the diffusion approximation to queueing networks II: Nonequilibrium distributions and applications to computer modeling. *J. ACM*, 21(3):459–469, 1974.
16. C. Priami, A. Regev, W. Silverman, and E. Shapiro. Application of a stochastic name passing calculus to representation and simulation of molecular processes. *Information Processing Letters*, 80:25–31, 2001.

17. A. Regev, W. Silverman, and E. Shapiro. Representation and simulation of biochemical processes using π-calculus process algebra. In *Pacific Symposium on Biocomputing 2001 (PSB 2001)*, pages 459–470, 2001.
18. N. Thomas, J.T. Bradley, and W.J. Knottenbelt. Stochastic analysis of scheduling strategies in a grid-based resource model. *IEE Software Engineering*, 151(5):232–239, September 2004.