Structural and Fluid Analysis for Large Scale PEPA Models — With Applications to Content Adaptation Systems

Jie Ding

A thesis submitted for the degree of Doctor of Philosophy.
The University of Edinburgh.
January 2010

Abstract

The stochastic process algebra PEPA is a powerful modelling formalism for concurrent systems, which has enjoyed considerable success over the last decade. Such modelling can help designers by allowing aspects of a system which are not readily tested, such as protocol validity and performance, to be analysed before a system is deployed. However, model construction and analysis can be challenged by the size and complexity of large scale systems, which consist of large numbers of components and thus result in state-space explosion problems. Both structural and quantitative analysis of large scale PEPA models suffers from this problem, which has limited wider applications of the PEPA language. This thesis focuses on developing PEPA, to overcome the state-space explosion problem, and make it suitable to validate and evaluate large scale computer and communications systems, in particular a content adaption framework proposed by the Mobile VCE.

In this thesis, a new representation scheme for PEPA is proposed to numerically capture the structural and timing information in a model. Through this numerical representation, we have found that there is a Place/Transition structure underlying each PEPA model. Based on this structure and the theories developed for Petri nets, some important techniques for the structural analysis of PEPA have been given. These techniques do not suffer from the state-space explosion problem. They include a new method for deriving and storing the state space and an approach to finding invariants which can be used to reason qualitatively about systems. In particular, a novel deadlock-checking algorithm has been proposed to avoid the state-space explosion problem, which can not only efficiently carry out deadlock-checking for a particular system but can tell when and how a system structure lead to deadlocks.

In order to avoid the state-space explosion problem encountered in the quantitative analysis of a large scale PEPA model, a fluid approximation approach has recently been proposed, which results in a set of ordinary differential equations (ODEs) to approximate the underlying CTMC. This thesis presents an improved mapping from PEPA to ODEs based on the numerical representation scheme, which extends the class of PEPA models that can be subjected to fluid approximation. Furthermore, we have established the fundamental characteristics of the derived ODEs, such as the existence, uniqueness, boundedness and nonnegativeness of the solution. The convergence of the solution as time tends to infinity for several classes of PEPA models, has been proved under some mild conditions. For general PEPA models, the convergence is proved under a particular condition, which has been revealed to relate to some famous constants of Markov chains such as the spectral gap and the Log-Sobolev constant. This thesis has established the consistency between the fluid approximation and the underlying CTMCs for PEPA, i.e. the limit of the solution is consistent with the equilibrium probability distribution corresponding to a family of underlying density dependent CTMCs.

These developments and investigations for PEPA have been applied to both qualitatively and quantitatively evaluate the large scale content adaptation system proposed by the Mobile VCE. These analyses provide an assessment of the current design and should guide the development of the system and contribute towards efficient working patterns and system optimisation.

Declaration of originality

I hereby declare that the research recorded in this thesis and the thesis itself was composed and originated entirely by myself in the School of Engineering and the School of Informatics at the University of Edinburgh.

Jie Ding

Acknowledgements

First and foremost, I deeply thank my supervisors Prof. Jane Hillston and Dr David I. Laurenson. Without their invaluable guidance and assistance during my PhD student life, I would not complete this thesis. I appreciate Dr Allan Clark's help on the experiments using ipc/Hydra, the results of which are presented in Figure 2.6 and 2.7 in this thesis.

I gratefully acknowledge the financial support from the Mobile VCE, without which I would not have been in a position to commence and complete this work.

Finally, I am forever indebted to my parents and my sister, who have given me constant support and encouragement. During the course of my PhD research, my mother passed away, with the regret of having no chances to see my thesis. This thesis is in memory of my mother. Om mani padme hum!

In memory of my mother

Contents

Declaration of originality . . . iii
Acknowledgements . . . iv
Contents . . . vi
List of figures . . . x
List of tables . . . xii
Acronyms and abbreviations . . . xiii

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Contribution of the Thesis . . . 3
1.3 Organisation of the Thesis . . . 6
1.4 Publication List and Some Notes . . . 8

2 Background **11**
2.1 Introduction . . . 11
2.2 Content Adaptation Framework by Mobile VCE . . . 11
2.2.1 Content adaptation . . . 11
2.2.2 Mobile VCE Project . . . 13
2.2.3 Content adaptation framework by Mobile VCE . . . 13
2.2.4 The working cycle . . . 15
2.3 Introduction to PEPA . . . 16
2.3.1 Components and activities . . . 17
2.3.2 Syntax . . . 18
2.3.3 Execution strategies, apparent rate and operational semantics . . . 20
2.3.4 CTMC underlying PEPA model . . . 24
2.3.5 Attractive features of PEPA . . . 24
2.4 Performance Measures and Performance Evaluation for Small Scale Content Adaptation Systems . . . 24
2.4.1 PEPA model and parameter settings . . . 25
2.4.2 Performance measures and performance evaluation: throughput and utilisation . . . 28
2.4.3 Performance measures and performance evaluation: response time . . . 32
2.4.4 Enhancing PEPA to evaluate large scale content adaptation systems . . . 36
2.5 Related work . . . 37
2.5.1 Decomposition technique . . . 37
2.5.2 Tensor representation technique . . . 40
2.5.3 Abstraction and stochastic bound techniques . . . 41
2.5.4 Fluid approximation technique . . . 41
2.6 Summary . . . 44

3 New Representation for PEPA: from Syntactical to Numerical **45**
3.1 Introduction . . . 45

3.2 Numerical Vector Form . . . 46
3.2.1 State-space explosion problem: an illustration by a tiny example . . . 46
3.2.2 Definition of numerical vector form . . . 47
3.2.3 Efficiency of numerical vector form . . . 49
3.2.4 Model 1 continued . . . 51
3.3 Labelled Activity and Activity Matrix . . . 53
3.3.1 Original definition of activity matrix . . . 53
3.3.2 Labelled activity and modified activity matrix . . . 55
3.4 Transition Rate Function . . . 61
3.4.1 Model 2 continued . . . 61
3.4.2 Definitions of transition rate function . . . 63
3.4.3 Algorithm for deriving activity matrix and transition rate functions . . . 65
3.5 Associated Methods for Qualitative and Quantitative Analysis of PEPA Models 68
3.5.1 Numerical and aggregated representation for PEPA . . . 68
3.5.2 Place/Transition system . . . 69
3.5.3 Aggregated CTMC and ODEs . . . 70
3.6 Summary . . . 72

4 Structural Analysis for PEPA Models 73
4.1 Introduction . . . 73
4.2 Place/Transtion Structure underlying PEPA Models . . . 74
4.2.1 Dynamics of PEPA models . . . 74
4.2.2 Place/Transition Structure in PEPA Models . . . 77
4.2.3 Some terminology . . . 79
4.3 Invariance in PEPA models . . . 81
4.3.1 What are invariants . . . 81
4.3.2 How to find invariants . . . 83
4.3.3 Conservation law as a kind of invariance . . . 86
4.4 Linearisation of State Space for PEPA . . . 88
4.4.1 Linearisation of state space . . . 88
4.4.2 Example . . . 93
4.5 Improved Deadlock-Checking Methods for PEPA . . . 94
4.5.1 Preliminary . . . 95
4.5.2 Equivalent deadlock-checking . . . 96
4.5.3 Deadlock-checking algorithm in LRS$^{\text{Psf}}$ . . . 98
4.5.4 Examples . . . 99
4.6 Summary . . . 103

5 Fluid Analysis for Large Scale PEPA Models—Part I: Probabilistic Approach 105
5.1 Introduction . . . 105
5.2 Fluid Approximations for PEPA Models . . . 106
5.2.1 Deriving ODEs from PEPA models . . . 107
5.2.2 Example . . . 110
5.2.3 Existence and uniqueness of ODE solution . . . 112
5.3 Convergence of ODE Solution: without Synchronisations . . . 114
5.3.1 Features of ODEs without synchronisations . . . 114
5.3.2 Convergence and consistency for the ODEs . . . 118

5.4 Relating to Density Dependent CTMCs . . . 119
5.4.1 Density dependent Markov chains from PEPA models . . . 120
5.4.2 Consistency between the derived ODEs and the aggregated CTMCs . . 123
5.4.3 Boundedness and nonnegativeness of ODE solutions . . . 124
5.5 Convergence of ODE Solution: under a Particular Condition . . . 125
5.6 Investigation of the Particular Condition . . . 129
5.6.1 An important estimation in the context of Markov kernel . . . 130
5.6.2 Investigation of the particular condition . . . 131
5.7 Summary . . . 134

6 Fluid Analysis for Large Scale PEPA Models—Part II: Analytical Approach 137
6.1 Introduction . . . 137
6.2 Analytical Proof of Boundedness and Nonnegativeness . . . 137
6.2.1 Features of the derived ODEs . . . 138
6.2.2 Boundedness and nonnegativeness of solutions . . . 139
6.3 A Case Study on Convergence with Two Synchronisations . . . 141
6.3.1 ODEs derived from an interesting model . . . 141
6.3.2 Numerical study for convergence . . . 148
6.4 Convergence For Two Component Types and One Synchronisation (I): A Special Case . . . 151
6.4.1 A previous model and the derived ODE . . . 151
6.4.2 Outline of proof . . . 153
6.4.3 Proofs not relying on explicit expressions . . . 161
6.5 Convergence For Two Component Types and One Synchronisation (II): General Case . . . 168
6.5.1 Features of coefficient matrix . . . 168
6.5.2 Eigenvalues of Q_1 and Q_2 . . . 173
6.5.3 Convergence theorem . . . 177
6.6 Summary . . . 179

7 Deriving Performance Measures for Large Scale Content Adaptation Systems 181
7.1 Introduction . . . 181
7.2 Fluid Approximation of the PEPA Model of Content Adaptation Systems . . . 182
7.2.1 ODEs derived from the PEPA model of content adaptation systems . . 182
7.2.2 The properties of the solution of the derived ODEs . . . 185
7.3 Deriving Quantitative Performance Measures through Different Approaches . . 192
7.3.1 Deriving performance measures through fluid approximation approach . 192
7.3.2 Deriving average response time via Little's Law . . . 196
7.3.3 Deriving performance measures through stochastic simulation approach 197
7.3.4 Comparison of performance measures through different approaches . . 201
7.4 Performance Analysis for Large Scale Content Adaptation Systems . . . 207
7.4.1 Scalability analysis . . . 209
7.4.2 Capacity planning . . . 210
7.4.3 Sensitivity analysis . . . 212
7.5 Structural Analysis of A Subsystem . . . 215
7.5.1 Adaptation management model . . . 215
7.5.2 Invariants . . . 217

7.5.3 Deadlock-checking . . . 220
7.6 Summary . . . 220

8 Conclusions **223**
8.1 Introduction . . . 223
8.2 Summary . . . 223
8.3 Limitations of the thesis and Future work . . . 226

References **229**

A From Process Algebra to Stochastic Process Algebra **243**
A.1 Process algebra . . . 243
A.2 Timed process algebra . . . 244
A.3 Probabilistic process algebra . . . 245
A.4 Stochastic process algebra . . . 245

B Two Proofs in Chapter 3 **247**
B.1 Proof of consistency between transition rate function and PEPA semantics . . . 247
B.2 Proof of Proposition 3.4.3 . . . 248

C Some Theorems and Functional Analysis of Markov chains **251**
C.1 Some theorems . . . 251
C.2 Spectral gaps and Log-Sobolev constants of Markov chains . . . 252

D Proofs and Some Background Theories in Chapter 6 **257**
D.1 Some basic results in mathematical analysis . . . 257
D.2 Some theories of differential equations . . . 258
D.2.1 The Jordan Canonical Form . . . 258
D.2.2 Some obtained results . . . 261
D.3 Eigenvalue properties of coefficient matrices of Model 3 . . . 264
D.4 Eigenvalue property for more general cases . . . 265
D.5 A proof of (6.42) in Section 6.4.2.2 . . . 266
D.6 A proof of Lemma 6.4.1 . . . 270

List of figures

1.1 A diagram of the work for PEPA 4
1.2 Reading order of chapters 8

2.1 Logical architecture of content adaptation framework 14
2.2 Working cycle of content adaptation management 16
2.3 Operational semantics of PEPA 23
2.4 Throughput of the CA adaptation ($(M, N, P, Q) = 1$) 30
2.5 Utilisation of the CA 31
2.6 Response time as a function of adaptation rate and AM decision rate 34
2.7 Response time changes with the number of PDEs 35
2.8 Throughput and utilisation changes with the number of PDEs 35
2.9 Adaptation rate and the number of PDEs' impact on the system performance . 35

3.1 Transition between states (Model 1) 52
3.2 Transition vectors form an activity matrix (Model 1) 54
3.3 Transition diagram of Model 2 56

4.1 Transition diagram of Model 1 ($M = N = 2$) 75
4.2 Transition systems of the components of Model 3 82

5.1 Transition diagram of Model 2 110
5.2 Transition diagram of Model 5 117
5.3 Convergence and consistency diagram for derived ODEs 128

6.1 Transition systems of the components of Model 3 142
6.2 Numerical study for Model 3: rates $(1, 1, 1, 1, 1, 1)$ 148
6.3 Numerical study for Model 3: rates $(1, 1, 1, 10, 1, 10)$ 149
6.4 Numerical study for Model 3: rates $(1, 1, 10, 1, 1, 1)$ 149
6.5 Numerical study for Model 3: rates $(20, 20, 1, 1, 1, 1)$ 150
6.6 Illustration of derived ODEs with component types and one synchronisation . . 170

7.1 Concentrated density dependent CTMCs (concentration level one) approximate the ODEs 187
7.2 Concentrated density dependent CTMCs (concentration level two) approximate the ODEs 188
7.3 Concentrated density dependent CTMCs approximate the ODEs ($(M, N, P, Q) = (30, 20, 20, 20)$) 189
7.4 Concentrated density dependent CTMCs approximate the ODEs ($(M, N, P, Q) = (300, 20, 20, 20)$) 190
7.5 Comparison between three approaches to derive response time 203
7.6 Fluid approximation and stochastic simulation 206
7.7 Throughput vs the number of PDEs 208

7.8 Impact of the number of C/S Providers on performance 210
7.9 Impact of the number of AMs on performance 211
7.10 Impact of the number of CAs on performance 212
7.11 Impact of assimilation rate on performance 213
7.12 Impact of decision rate on performance 213
7.13 Impact of adaptation rate on performance 214
7.14 Working cycle of content adaption model 216

List of tables

2.1 Comparison between PEPA and other paradigms . . . 24
2.2 Parameter settings (unit of duration: millisecond) . . . 28

3.1 Elapsed time of state pace derivation . . . 53
3.2 Originally defined activity matrix of Model 2 . . . 56
3.3 Modified activity matrix of Model 2 . . . 57
3.4 From syntactical and separated to numerical and aggregated representation for PEPA . . . 68

4.1 Activity matrix and pre activity matrix of Model 1 . . . 75
4.2 Comparison between three approaches . . . 79
4.3 P/T structure in PEPA models . . . 81
4.4 Activity matrix of Model 3 . . . 84
4.5 Elapsed time of state space derivation . . . 88
4.6 Activity matrix and pre activity matrix of Model 4 . . . 101

5.1 Comparison with respect to restrictions . . . 108
5.2 Activity matrix and transition rate functions of Model 2 . . . 110
5.3 Activity matrix and transition rate function of Model 5 . . . 117
5.4 Activity matrix and transition rate function of Model 1 . . . 122
5.5 Fundamental characteristics of derived ODEs from PEPA models . . . 135

6.1 A summary for the convergence of Model 3 . . . 148
6.2 Complex dynamical behaviour of Model 3: starting state (1,1,5,0,0,5) . . . 150

7.1 Percentage error between CTMCs and ODEs . . . 188
7.2 Deriving performance measures through stochastic simulation . . . 200
7.3 Running times (sec.) of small scale experiments . . . 204
7.4 Factors and effects on paths and performance . . . 205
7.5 Running time of large scale experiments . . . 205
7.6 Activity matrix $\mathbf{C}$ of the sub content management model . . . 218
7.7 Invariants of the sub content management model . . . 219

Acronyms and abbreviations

ACP	Algebra of Communicating Processes
AM	Adaptation Manager
CA	Content Adaptor
CCS	Calculus of Communicating Systems
CLS	Calculus of Looping Sequences
CSP	Communicating Sequential Processes
C/S Provider	Content/Service Provider
CTMC	continuous-time Markov chain
DME	Device Management Entity
EMPA	Extended Markovian Process Algebra
EQ	equal conflict
FC	free choice
IMC	interactive Markov chains
Log-Sobolev	logarithm Sobolev
LOTOS	Language of Temporal Ordering Specifications
Mobile VCE	the virtual centre of excellence in mobile communications
ODEs	ordinary differential equations
OWL	Web Ontology Language
PAA	Personal Assistant Agent
PCM	Personal Content Manager
PDE	Personal Distributed Environment
PEPA	Performance Evaluation Process Algebra
P/T	Place/Transition
RCAT	reversed compound agent theorem
SDEs	stochastic differential equations
SOAP	Simple Object Access Protocol
SSA	stochastic simulation algorithm
TIPP	TImed Process for Performance Evaluation
UDDI	Universal Description, Discovery and Integration

WSDL	Web Services Description Language

Chapter 1
Introduction

1.1 Motivation

In the new era of wireless, mobile connectivity, there has been a great increase in the heterogeneity of devices and network technologies. For instance, mobile terminals may significantly vary in their software, hardware and network connectivity characteristics. Meanwhile, there is an increased variety of services being offered to meet users' preferences and needs, for example, mobile TV services, and shopping services such as Ebay. The service may embody functionality and deliver multiple content items to mobile end users in a specific manner. However, the mismatch between the diversity of content and the heterogeneity of devices presents a research challenge [LL02b]. *Content adaptation* has emerged as a potential effective solution to cope with the problem of delivering services and content to users in a variety of contexts.

The virtual centre of excellence in mobile communications (Mobile VCE) is addressing this area in the programme entitled "Removing the Barriers to Ubiquitous Services". The programme has been investigating the tools and techniques essential to hiding complexity in the heterogeneous communications environment that is becoming a reality. In particular, the work makes use of agents that manage personal preferences, and control the adaptation of content to meet the system requirements for a user to view content they have requested. The interaction between the entities in the user-controlled devices and the network that is required to achieve this then becomes a significant issue.

Performance modelling provides an important route to gaining insight about how systems will perform both qualitatively and quantitatively. Such modelling can help designers by allowing aspects of a system which are not readily tested, such as protocol validity and performance, to be analysed before a system is deployed. This thesis will discuss and present a high-level modelling formalism—the stochastic process algebra PEPA developed by Hillston [Hil96]—to validate and evaluate the potential designs and configurations of a content adaptation framework proposed by the Mobile VCE.

Stochastic process algebras are powerful modelling formalisms for concurrent systems, which have enjoyed considerable success over the last decade. As a process algebra, PEPA is a compositional description technique which allows a model of system to be developed as a number of interacting *components* which undertake *activities*. In addition to the system description aspects the process algebra is equipped with techniques for manipulating and analysing models, all implemented in tools [TDG09]. Thus analysis of the model becomes automatic once the description is completed. In a *stochastic* process algebra additional information is incorporated into the model, representing the expected duration of actions and the relative likelihood of alternative behaviours. This is done by associating an exponentially distributed random variable with each action in the model. This quantification allows quantified reasoning to be carried out on the model. Thus, whereas a process algebra model can be analysed to assess whether it behaves correctly, a stochastic process algebra model can be analysed both with respect to correctness and timeliness of behaviour.

Once a PEPA model has been constructed two different analysis approaches are accessible from the single model:

- The model may be used to derive a corresponding (discrete state) continuous time Markov chain (CTMC) which can be solved for both transient and equilibrium behaviour, allowing the calculation of measures such as expected throughput, utilisation and response time distributions.
- Desirable properties of the system can be expressed as logical formulae which may be automatically checked against the formal description of the system, to test whether the property holds. This can be particularly useful in checking that protocols behave appropriately and that certain desired properties of the system are not violated.

However, these two basic types of analysis can be challenged by the size and complexity of large scale systems. In fact, a realistic system may consist of large numbers of users and other entities, which results in the size of the state space underlying the system being too large to allow analysis. This problem is termed the *state-space explosion problem*. Both qualitative and quantitative analysis of stochastic process algebras and many other formal modelling approaches suffer this problem. For instance, the current deadlock-checking algorithm of PEPA relies on exploring the entire state space to find whether a deadlock exists. For large scale PEPA models, deadlock-checking becomes impossible due to the state-space explosion problem.

For quantitative analysis of PEPA models, a novel approach—fluid approximation—to avoid this problem has recently been developed by Hillston [Hil05a], which results in a set of ordinary differential equations (ODEs) to approximate the underlying CTMC. However, this approach is restricted to a class of models and needs to be extended. Furthermore, the approach gives rise to some fundamental theoretical questions. For example, whether the solution of the ODEs converges to a finite limit as time tends to infinity? What is the relationship between the derived ODEs and the underlying CTMC? etc. Solving these problems can not only bring confidence in the new approach, but can also provide new insight into, as well as a profound understanding of, performance formalisms.

Therefore, it is an important issue and this thesis focuses on this topic, to both technically and theoretically develop the stochastic process algebra PEPA to overcome the state-space explosion problem, and make it suitable to validate and evaluate large scale computer and communications systems, in particular the content adaption framework proposed by the Mobile VCE.

1.2 Contribution of the Thesis

In this section we outline the work which has been undertaken, highlighting the primary contributions of the thesis. These include both theoretical underpinnings for large scale modelling and an application to the evaluation of large scale content adaptation systems.

A PEPA model is constructed to approximately and abstractly represent a system while hiding its implementation details. Based on the model, performance properties of the dynamic behaviour of the system can be assessed, through some techniques and computational methods. This process is referred to as the *performance modelling* of the system, which mainly involves three levels: model construction and representation, technical computation and performance derivation. Our enhancement for PEPA embodies these three aspects, which are illustrated by Figure 1.1. At the first level, we propose a new representation scheme to numerically describe any given PEPA model, which provides a platform to directly employ a variety of approaches to analyse the model. These approaches are shown at the second level. At this level, the current fluid approximation method for the quantitative analysis of PEPA is expanded, as well as investigated, mainly with respect to its convergence and the consistency and comparison between this method and the underlying CTMC. Moreover, a Place/Transition (P/T) structure-based approach is proposed to qualitatively analyse the model. At the third level, both qualitative and

quantitative performance measures can be derived from the model through those approaches. In particular, we demonstrate what kind of performance measures can be derived through the fluid approximation approach. A stochastic simulation algorithm based on the numerical representation scheme is proposed to obtain general performance metrics. Moreover, we can determine some structural properties of the model such as invariance and deadlock-freedom without suffering the state-space explosion problem.

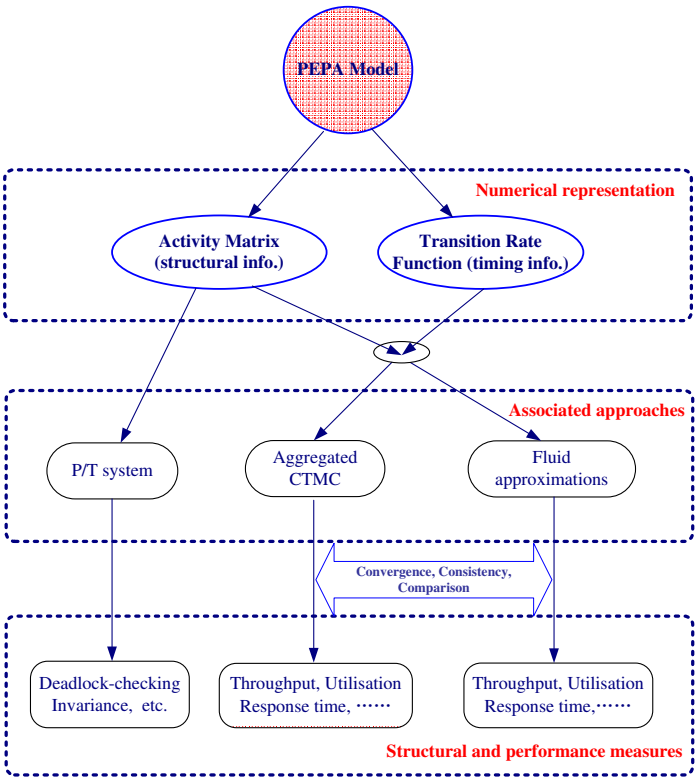


Figure 1.1: *A diagram of the work for PEPA*

These achievements, as well as an application to evaluate large scale content adaptation systems, are detailed in the following:

1. **New numerical representation scheme for PEPA**: This thesis proposes a new numerical representation scheme for PEPA. In this scheme, labelled activities are defined to cope with the difference between actions in PEPA and transitions in the underlying CTMC, so that the correspondence between them is one-to-one. Modified activity matrices based on the labelled activities are defined to capture structural information about PEPA models. Moreover, transition rate functions are proposed to capture the timing information. These concepts numerically describe and represent a PEPA model, and provide a platform for

conveniently and easily exposing and simulating the underlying CTMC, deriving the fluid approximation, as well as leading to an underlying P/T structure. These definitions have been proved consistent with the original semantics of PEPA. An algorithm for automatically deriving these definitions from any given PEPA model has been provided. Some good characteristics of this numerical representation have been revealed. For example, using numerical vector forms the exponential increase of the size of the state space with the number of components can be reduced to at most a polynomial increase.

2. **Efficient techniques for qualitative analysis of PEPA**: Through the numerical representation of PEPA, we have found that there is a P/T structure underlying each PEPA model, which reveals tight connections between stochastic process algebras and stochastic Petri nets. Based on this structure and the theories developed for Petri nets, several powerful techniques and approaches for structural analysis of PEPA are proposed. For instance, we give a method of deriving and storing the state space which avoids the problems associated with populations of components, and an approach to find invariants which can be used to qualitatively reason about systems. Moreover, a structure-based deadlock-checking algorithm is proposed, which can avoid the state-space explosion problem.

3. **Technical and theoretical developments of fluid-flow analysis of PEPA**: Based on the numerical representation scheme, we have proposed a new approach for the fluid approximation of PEPA, which extends the current semantics of mapping PEPA models to ODEs by relaxing previous restrictions. The derived ODEs through our approach can be considered as the limit of a family of density dependent CTMCs underlying the given PEPA model. The fundamental characteristics of the derived ODEs have been established, including the existence, uniqueness, boundedness and nonnegativeness of the solution. We have revealed consistency between the deterministic ODEs and the underlying stochastic CTMCs for general PEPA models: if the solution of the derived ODEs converges as time tends to infinity, then the limit is an expectation in terms of the steady-state probability distributions of the corresponding density dependent CTMCs. The convergence of the solution of the ODEs has been proved under a particular condition, which relates the convergence problem to some well-known constants of Markov chains such as the spectral gap and the Log-Sobolev constant. For several classes of PEPA models, the convergence has been demonstrated under some mild conditions, and the coefficient matrices of the derived ODEs have been exposed to have the following property: all eigenvalues are either zeros or have negative real parts. In particular, invariants in the PEPA models

have been shown to play an important role in the proof of convergence.

4. **Performance derivation methods for large scale PEPA models**: We have shown what kind of performance metrics can be derived from a PEPA model through the approach of fluid approximation and how this can be done. For the measures that cannot be derived by this approach, we have presented a stochastic simulation algorithm which is based on the numerical representation scheme. Detailed comparisons between these two approaches, in terms of both computational cost and accuracy, have been provided.

5. **Performance validation and evaluation framework for large scale content adaptation systems**: We have proposed a formal approach as well as associated techniques and methods to validate (e.g. check deadlocks) and evaluate content adaptation systems, particularly at large scales. We have developed powerful techniques for future qualitative analysis, including qualitative reasoning techniques through invariants as well as structure-based methods for protocol validation, and so on. Quantitative analysis, in terms of the response time of the system, has been carried out to assess the current design. In particular, we have shown that the average response time is approximately governed by a set of corresponding nonlinear algebra equations, based on which scalability and sensitivity analysis, as well as capacity planning and system optimisation, can be carried out simply and efficiently.

1.3 Organisation of the Thesis

The remaining chapters of this thesis are organised as follows:

Chapter 2 (Background): This chapter will present some background to the Mobile VCE project and an introduction to PEPA, as well as some performance analyses for small scale content adaptation systems.

Chapter 3 (Numerical representation for PEPA): This chapter will demonstrate a numerical presentation scheme for PEPA. The definitions of labelled activities, which form a modified activity matrix, and transition rate functions as well as their corresponding properties will be given. Moreover, this chapter will provide an algorithm for deriving this scheme from any PEPA model.

Chapter 4 (Structural analysis for PEPA): This chapter will reveal that there is a P/T structure underlying each PEPA model. Based on this structure and the theories developed for Petri nets, structural analysis for PEPA will be carried out. This chapter will provide powerful methods to derive and store the state space and to find invariants in PEPA models. In particular, a new deadlock-checking approach for PEPA will be proposed, to avoid the state-space explosion problem.

Chapter 5 (Fluid analysis for PEPA (I)—through a probabilistic approach): In this chapter, an improved mapping from PEPA to ODEs will be given, which extends the current mapping semantics by relaxing certain restrictions. Some fundamental characteristics such as the existence and uniqueness of the solution of the derived ODEs will be presented. For PEPA models without synchronisations, the solution of the ODEs converges to a limit which coincides with the stable probability distribution of the underlying CTMC.

Chapter 6 (Fluid analysis for PEPA (II)—through an analytic approach): This chapter will present a purely analytical proof of the boundedness and nonnegativeness of the solution of the derived ODEs from PEPA models. A case study will show the important role of invariance in the proof of convergence. For a class of PEPA models, i.e. models with two component types and one synchronisation, we will demonstrate the convergence under some mild conditions.

Chapter 7 (Deriving performance measures for large scale content adaptation systems): In this chapter, we will show the kind of performance measures available from the fluid approximation of a PEPA model and how these measures can be derived. A stochastic simulation algorithm for deriving performance based on the numerical representation scheme will also be presented. This chapter will present applications of enhanced PEPA to validate and evaluate large scale content adaptation systems. We will carry out scalability and sensitivity analysis, as well as capacity planning for content adaptation systems, to assess the performance. The computational cost and accuracy of different approaches for PEPA analysis, particularly the fluid approximation and the simulation approaches, will be experimentally compared and studied. In addition, some structural analysis for a subsystem will be demonstrated.

Chapter 8 (Conclusions): This chapter will conclude the thesis and propose future work.

The reading order of this thesis is illustrated by Figure 1.2.

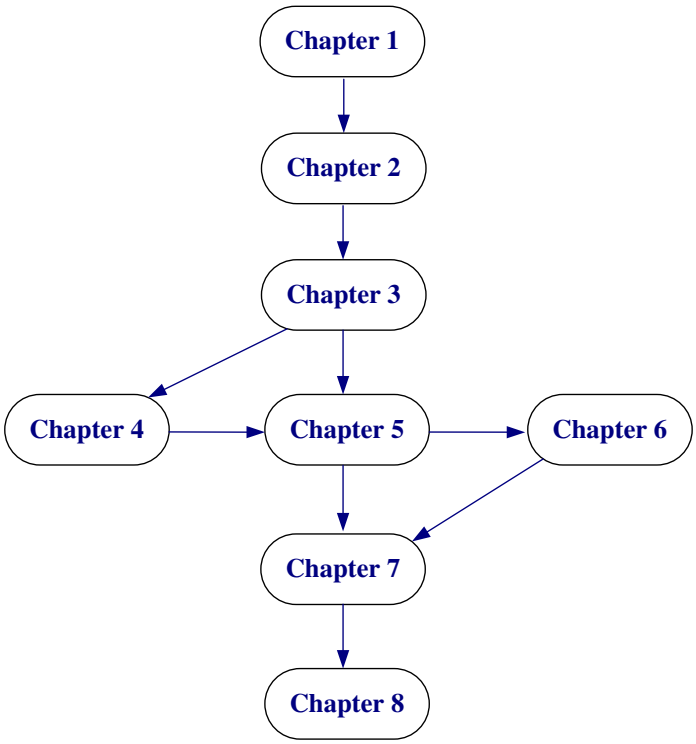


Figure 1.2: *Reading order of chapters*

1.4 Publication List and Some Notes

This section gives a publication list, with some notes indicating the correspondence between these papers and the content of this thesis.

1. Jie Ding, J. Hillston, D. Laurenson, *Evaluating the response time of large scale content adaptation systems*, accepted by the International Communication Conference (2010), Cape Town, South Africa.

   (This paper presents the simulated results of large scale content adaptation systems. The analysis of these results is similar to the discussions in Section 7.4 of Chapter 7, although the results in Section 7.4 are mainly derived through the approach of fluid approximation.)

2. Jie Ding, J. Hillston, D. Laurenson, *Performance modelling of content adaptation for a personal distributed environment*, Wireless Personal Communications: An International

Journal, Volume 48, Issue 1, Jan. 2009.

(This paper presents the performance modelling of a small scale content adaptation system, in which some analysis and discussions appear in Section 2.4 of Chapter 2.)

3. Jie Ding, J. Hillston, *A new deadlock checking algorithm for PEPA*, 8th Workshop on Process Algebra and Stochastically Timed Activities (PASTA'09), Edinburgh, UK.

   (A brief introduction to the numerical representation scheme of PEPA and based on which some structural analysis of PEPA models, particularly the deadlock-checking method, have been presented in this paper. These materials are mainly shown in Chapter 3 and Chapter 4 in this thesis.)

4. A. Attou, Jie Ding, D. Laurenson, and K. Moessner, *Performance modelling and evaluation of an adaptation management system*, International Symposium on Performance Evaluation of Computer and Telecommunication Systems 2008 (SPECTS'08), Edinburgh, UK.

   (This paper presents the performance modelling and evaluation for the design of the entity Adaptation Manager. In the interest of brevity the main content of this paper is not included in this thesis.)

5. Jie Ding, J. Hillston, *Convergence of the fluid approximations of PEPA Models*, 7th Workshop on Process Algebra and Stochastically Timed Activities (PASTA'08), Edinburgh, UK.

   (This paper presents the relationship between the fluid approximation and the density dependent Markov chains underlying the same PEPA model, as well as the convergence of the solution of the derived ODEs as time goes to infinity under some conditions. These results are mainly presented in Chapter 5.)

6. Jie Ding, J. Hillston, *on ODEs from PEPA models*, 6th Workshop on Process Algebra and Stochastically Timed Activities (PASTA'07), London, UK.

   (The fundamental characteristics of fluid approximation of PEPA models including the existence, uniqueness, boundedness and nonegativeness of the solutions of the ODEs derived from a class of PEPA models, have been established in this paper, while they are extended in Chapter 6 in this thesis.)

Chapter 2
Background

2.1 Introduction

This chapter will give an introduction to the Mobile VCE project and the stochastic process algebra PEPA, which are shown in Section 2 and 3 respectively. Then, in Section 4, we present performance measures and performance evaluation for small scale content adaptation systems which are based on the content adaptation framework proposed by the Mobile VCE. A literature review of the techniques developed to deal with the state-space explosion problem will be presented in Section 5. Finally, we conclude this chapter in Section 6.

2.2 Content Adaptation Framework by Mobile VCE

In this section, we will give an introduction to content adaptation and the Mobile VCE project, as well as describing a content adaptation framework proposed by the Mobile VCE.

2.2.1 Content adaptation

As networks become more sophisticated, both in terms of their underlying technology and the applications running upon them, it is crucial that users' expectations and requirements are anticipated and met. In particular, users are basically not concerned with the technological aspects of communications. However, at present, they need to be aware of a multitude of details about equipment and benefits of one communication strategy over another, as well as how to connect systems together to achieve the communication that they desire. As the number of possible communication strategies increases, so does the complexity of negotiating the most appropriate method of delivering content that the user wishes to access.

Users, requesting content from a provider, wish that content to be usable in a specific device or devices. Currently a content provider may provide a number of formats of a particular content to suit a selection of devices, or they may only provide a single format of the data. With the

rapidly growing variety of devices that a user may expect to use for delivering a particular content, providing content tailored to each device becomes an infeasible task for the content provider [LL02a]. Thus, in order for a user, who needs the content in a different format, to be able to make use of that content, a transformation needs to take place. In the wider context, not only may transformation from one format to another be required, but additionally the content may need to be modified, for example its bit-rate reduced, in order to meet quality of service constraints. This process is called *content adaptation*. The adaptation, itself, may take place within the domain of the service provider, the domain of the user, or may take place within the network as a third party service.

Content adaptation can be defined as "the set of measures taken against a dynamically changing context, for the purpose of maintaining a user experience of the delivered content as close to that of the original content as possible" [Dey00]. Several techniques have been developed for content adaptation. One technique is transcoding, which changes the content coding format while preserving the same information. For example, to reduce the bit-rate or save device storage, a JPEG image is transcoded to PNG format. Another main technique is cross-modal adaptation. This transforms content from one modality to another, such as text to speech adaptation. There are other techniques for adaptation such as content recomposition for small displays [CWW07] where, for instance, useful regions in a video or image are extracted and re-composed in an image or video. Moreover, content adaptation management mechanisms have been incorporated into content distribution networks [MBC+00, KM06, EKBS04], to minimize the interference of adaptation with replication effectiveness [BB].

According to the location where the adaptation takes place, content adaptation techniques can be classified into three categories [BGGW02]: provider-based, client-based, and proxy-based. When the adaptation takes place on provider side (e.g. [MSCS99, PKP03]), the content provider could have a central control over how the content and service are presented to users. Client-side adaptation (e.g. [BHR+99, FAD+97]) is controlled by the end terminal. The user can impose his preference of the final result, but adaptation is very limited due to the limitation of devices. If adaptation occurs on a proxy site (e.g. [CEV00, FGC+97, LH05, YL03, JTW+07]), it will reduce the complexity at the client and provider sides but may lose the advantage of end-to-end security solutions.

In a ubiquitous environment, the adaptation should be context-aware, i.e. taking into consideration context covering user location and preference, device characteristics, network conditions

such as bandwidth, delay, QoS, content provider's digital rights, natural environment characteristics and content properties etc. As pointed out in [DLM08], context-aware application and system design has evolved from early ad-hoc application-specific [WHFG92] or toolkit-based [Dey00] design, to infrastructure-based design [Che04] which supports context-aware application in distributed and heterogeneous environment. Content adaptation has been widely acknowledged as one of the most important aspects for context-aware ubiquitous content delivery. The techniques of context acquisition and formatting and adaptation decision taking, have been applied to adaptation management [MSCS99, PKP03, LH05, YL03, JTW+07]. Some surveys on the content adaptation technologies can be found in [VCH03, Li06].

2.2.2 Mobile VCE Project

Before presenting a content adaptation framework put forward by the Mobile VCE, we first introduce the Mobile VCE project. The Mobile VCE is the operating name of the Virtual Centre of Excellence in Mobile and Personal Communications Ltd, a collaborative partnership of around 20 of the world's most prominent mobile communications companies and 7 UK universities each having long standing specialist expertise in relevant areas. Mobile VCE engages in industrially-led, long term, research in mobile and personal communications [htt].

Ubiquitous service represents a major future revenue stream for service providers, telecommunication operators and pervasive technology manufacturers, since Bluetooth, WiFi, WiMAX, UWB and more, are bringing the dream of ubiquitous access closer to reality. The "Removing the Barriers to Ubiquitous Services" programme aims at hiding the complexity involved in the communication of the content, and its delivery mechanism, from the user, empowering the user to access anything, at anytime, from anywhere.

2.2.3 Content adaptation framework by Mobile VCE

This subsection introduces the content adaptation framework proposed by the Mobile VCE. This introduction is based on the papers [Bus06, BID06, LM06]. For details, please refer to them.

A design of a content adaptation framework for a personal distributed environment, being developed under the auspices of the Mobile VCE, has been presented in [Bus06]. The concept of a Personal Distributed Environment (PDE), developed by the third programme of the Mo-

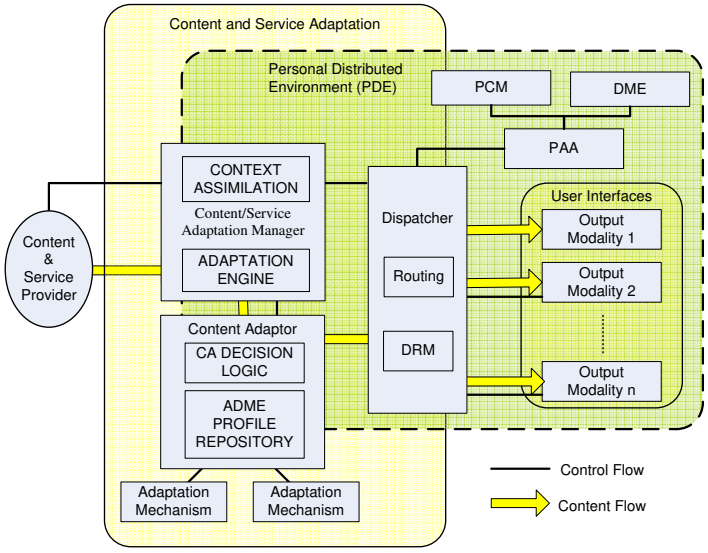


Figure 2.1: *Logical architecture of content adaptation framework [Bus06]*

bile VCE, core 3, is a user-centric view of communications in a heterogeneous environment, and is defined as those nodes over which a user has control. At the user side, the Personal Assistant Agent (PAA) is proposed to reduce the perceived complexity of future multi-device personal area networks by proactively managing the modes, the functions, the applications, and the connectivity of the user's devices. In addition, employing the Personal Content Manager (PCM) can effectively store content throughout the user's environments, maximizing availability and efficiency as well as retrieving the content in the most appropriate manner. The Device Management Entity (DME) acts as platform for the PAA and PCM to operate over.

The logical architecture for content adaptation based on this concept is depicted in Figure 2.1. There are two major functional entities within the framework to accomplish the content adaptation management, that is, the Adaptation Manager (AM) and the Content Adaptor (CA). The AM provides the required functionality to assimilate and distill user-related and content-related contexts into relevant rules so that actions may be determined by the decision logic. The CA organises the actual adaptation processes based on maintaining a profile of all the available adaptation mechanisms, contacting external adaptation mechanism providers and consolidating their capabilities to meet the adaptation requests, passed down from the AM. Semantic Web-based technologies, such as OWL [OWL] and Description Logic [Des], are used to represent contextual information and thus facilitate autonomous adaptation decision-making. Web Service technologies, such as WSDL [OWL], SOAP [OWL], UDDI [UDD], together with Semantic Web Service technologies, such as WSDL-S and OWL-S [OWL], are used to develop

adaptation mechanisms which carry out the actual content adaptation. The reader may refer to [AM07, LM07, TBID08] for details of using these technologies within the adaptation management framework.

The Dispatcher acts as a buffer, transporting the context information from the PAA to the AM, and delivering the content from the Content/Service Provider (C/S Provider) or the CA to the PDE, which forms the logical interface between the personal entities and the content and service adaptation framework as a whole.

2.2.4 The working cycle

A working scenario of the adaptation management based on the logical architecture is illustrated in Figure 2.2. The operation can be described as follows (for convenience, the component Dispatcher is omitted in the scenario):

1. When an external content/service request is activated, i.e. the user requests some content with specific preferences, the PDE will forward this request to the AM.

2. After receiving the request from the PDE, the AM asks for and receives the corresponding content/service information from the C/S Provider.

3. The AM assimilates and analyses the information from the user and the C/S Provider. If the C/S Provider can provide the desired content, then the AM requests the C/S Provider to directly forward the content to the PDE. Otherwise the content needs to be adapted to satisfy the user's requirement, so the AM asks for the CA's information for the purpose of content adaptation.

4. Based on the collected information from the user, the C/S Provider, and the CA, the AM ascertains appropriate options for content/service translation from the available options and constraints. Then the AM makes an adaptation plan and forwards it to the C/S Provider, which includes the adaptation authorization, adaptation schedule and network routing.

5. According to the received information from the AM, the C/S Provider makes a choice: either providing the content to the PDE directly or sending the content with the adaptation plan to the CA for adaptation.

6. After receiving the content with the adaptation plan, the CA starts the content adaptation and then forwards the adapted content to the PDE after the process is finished.

7. The PDE forwards the received content to the user interface.

In the following, we will use the PEPA language to describe and model the working cycle, and then to derive the performance measures. But first, we will present an introduction to PEPA.

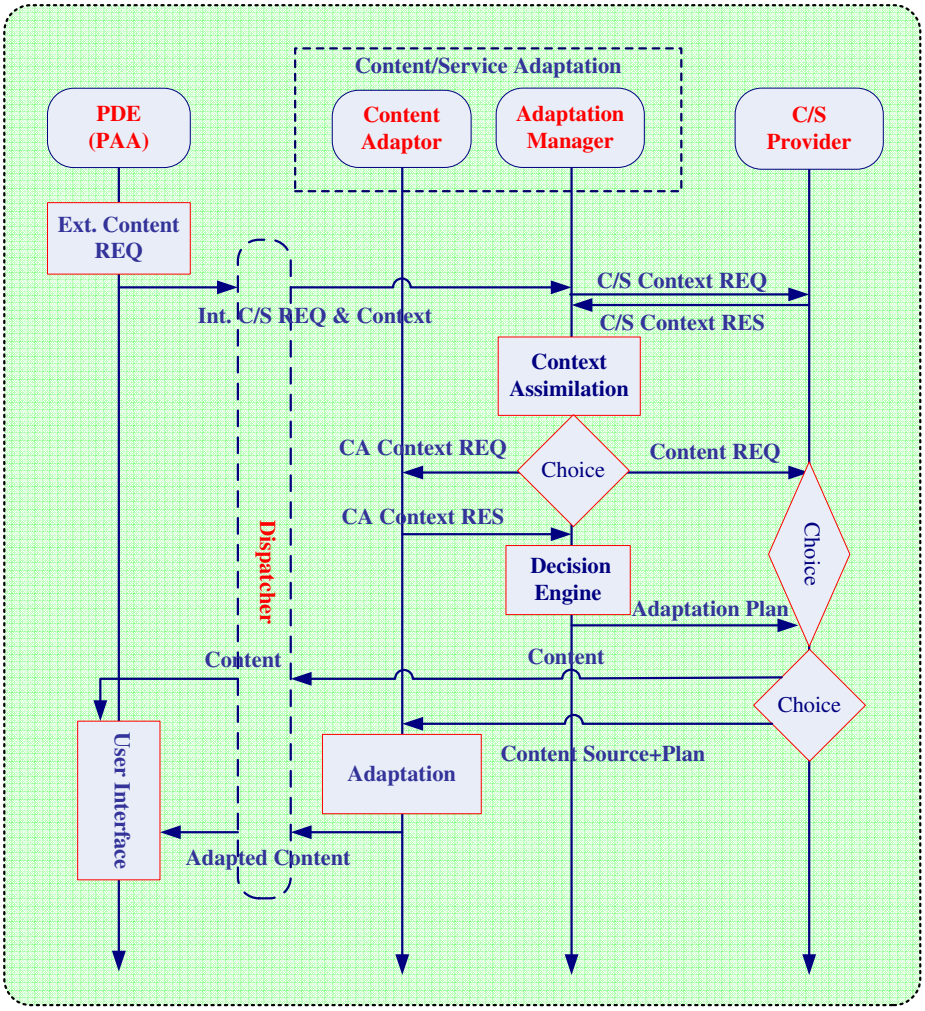


Figure 2.2: *Working cycle of content adaptation management*

2.3 Introduction to PEPA

This section presents an introduction to the PEPA (Performance Evaluation Process Algebra) language, which was developed by Hillston in the 1990s. For more details about PEPA, please

refer to [Hil96]. An overview of the history from the origin of process algebras to the current development of stochastic process algebras (e.g. Bio-PEPA), is presented in Appendix A.

2.3.1 Components and activities

PEPA is a high-level model specification language for low-level stochastic models, which allows a model of a system to be developed as a number of interacting *components* which undertake *activities*. A PEPA model has a finite set of components that correspond to identifiable parts of a system, or roles in the behaviour of the system. For example, the content adaptation system mentioned in the previous subsection has four types of components: the PDE, the AM, the CA and the C/S Provider. We usually use $\mathcal{C}$ to denote the set of all components.

The behaviour of each component in a model is captured by its activities. For instance, the component CA can perform "$ca\_adaptation$", i.e. the activity of content adaptation. In the PEPA language, each activity has a type, called *action type* (or simply *type*), and a duration, represented by *activity rate* (or simply *rate*). The duration of this activity satisfies a negative exponential distribution[1] which takes the rate as its parameter. For example, the adaptation activity in the above example can be written as $(ca\_adaptation, r_{ca\_adaptation})$, where $ca\_adaptation$ is the action type and $r_{ca\_adaptation}$ is the activity rate. The delay of the adaptation is determined by the exponential distribution with the parameter $r_{ca\_adaptation}$ or with the mean $\frac{1}{r_{ca\_adaptation}}$. Therefore, the probability that this activity happens within a period of time of length t is $F(t) = 1 - e^{-tr_{ca\_adaptation}}$. The set of all action types which a component P may next engage in is denoted by $\mathcal{A}(P)$ while the multiset of all activities which P may next fire is denoted by $\mathcal{A}ct(P)$. Then the sets of all possible action types and all possible activities are written as $\mathcal{A}$ and $\mathcal{A}ct$ respectively. If a component P completes an activity $\alpha \in \mathcal{A}(P)$ and then behaves as a component Q, then Q is called a *derivative* of P and this transition can be written as $P \xrightarrow{\alpha} Q$ or $P \xrightarrow{(\alpha,r)} Q$.

There is a special action type in PEPA, *unknown* type τ, which is used to represent an unknown or unimportant action. A special activity rate in PEPA is the *passive* rate, denoted by $\top$, which is unspecified.

[1] In the reminder of this thesis, "negative exponential distribution" is shorted as "exponential distribution" for convenience.

2.3.2 Syntax

This subsection presents the name and interpretation of combinators used in the PEPA language, which express the individual behaviours and interactions of the components.

Prefix: The prefix combinator "." is a basic mechanism by which the first behaviour of a component is designated. The component $(\alpha, r).P$, which has action type α and a duration which satisfies an exponential distribution with parameter r (mean $1/r$), carries out the activity (α, r) and subsequently behaves as component P. The time taken for completing the activity will be some Δt, sampled from the distribution.

For example, in the working cycle of the content adaptation system presented in Section 2.2.4, a component which can launch an external content request and then behaves as PDE_2, can be expressed by $(pde\_ext\_cont\_req, r_{pde\_ext\_cont\_req}).PDE_2$, where the rate $r_{pde\_ext\_cont\_req}$ reflects the expected rate at which the user will submit requests for the desired content or service. We would like to denote this component by PDE_1, that is

$$PDE_1 \stackrel{def}{=} (pde\_ext\_cont\_req, r_{pde\_ext\_cont\_req}).PDE_2,$$

where "$\stackrel{def}{=}$" is another combinator which will be introduced below.

Constant: The constant combinator "$\stackrel{def}{=}$" assigns names to components (behaviours). In the above example, i.e., $PDE_1 \stackrel{def}{=} (pde\_ext\_cont\_req, r_{pde\_ext\_cont\_req}).PDE_2$, we assign a name "$PDE_1$" to the component $(pde\_ext\_cont\_req, r_{pde\_ext\_cont\_req}).PDE_2$. This can also be regarded as the constant PDE_1 being given the behaviour of the component $(pde\_ext\_cont\_req, r_{pde\_ext\_cont\_req}).PDE_2$. The constant combinator can allow infinite behaviour over finite states to be defined via mutually recursive definitions.

Cooperation: Interactions between components can be represented through the cooperation combinator " $\underset{L}{\bowtie}$ ". In fact, $P \underset{L}{\bowtie} Q$ denotes cooperation between P and Q over action types in the cooperation set L. The cooperands are forced to synchronise on action types in L while they can proceed independently and concurrently with other enabled activities. The rate of the synchronised activity is determined by the slower cooperation. We write $P \parallel Q$ as an abbreviation for $P \underset{L}{\bowtie} Q$ when $L = \emptyset$.

In the working cycle of the content adaptation system, after the generation of the request the next event is to pass the request to the AM. This event should be represented by a synchronous

activity because it must be completed cooperatively by both the PDE and the AM. We use $pde\_int\_cont\_req$ to denote the action type. In the context of the PDE and the AM, the event is respectively modelled by

$$PDE_2 \stackrel{def}{=} (pde\_int\_cont\_req, r_{pde\_int\_cont\_req}).PDE_3$$

and

$$AM_1 \stackrel{def}{=} (pde\_int\_cont\_req, \top).AM_2.$$

The cooperation between PDE_2 and AM_1 can be expressed by $PDE_2 \underset{\{pde\_int\_cont\_req\}}{\bowtie} AM_1$. Here the notation "$\top$" reflects that for the AM the activity $pde\_int\_cont\_req$ is passive, and the rate is determined by its cooperation partner—the PDE. If the rate for the AM is not passive and assigned as r, i.e., $AM_1 \stackrel{def}{=} (pde\_int\_cont\_req, r).AM_2$, then the rate of the shared activity $pde\_int\_cont\_req$ is determined by the smaller of the two rates, i.e. $\min\{r_{pde\_int\_cont\_req}, r\}$.

Moreover, suppose there are two PDEs in the system and there is no cooperation between these two PDE_2. This can be modelled by $PDE_2 \parallel PDE_2$, which is equivalent to $PDE_2 \underset{\emptyset}{\bowtie} PDE_2$. Their cooperation with the AM through the activity $pde\_int\_cont\_req$ can be represented by $(PDE_2 \parallel PDE_2) \underset{\{pde\_int\_cont\_req\}}{\bowtie} AM_1$. We sometimes use the notation $PDE_2[M]$ to represent $\underbrace{PDE_2 \parallel \cdots \parallel PDE_2}_{M \text{ times}}$.

Choice: The choice combinator "+" expresses competition between activities. The component $P + Q$ models a system which may behave either as P or as Q. The activities of both P and Q are enabled. Whichever activity completes first must belong to P or Q. This activity distinguishes one of the components, P or Q, and the component $P + Q$ will subsequently behave as this component. Because of the continuous nature of the probability distributions, the probability of P and Q both completing an activity at the same time is zero. The choice combinator represents uncertainty about the behaviour of a component.

For example, in our content adaptation system, after forwarding the request to the AM, the PDE waits for a response. There are two possible responses, which are represented by two possible activities: receiving the content from the C/S Provider directly (csp_to_pde) or receiving the adapted content from the CA (ca_to_pde). This event can be represented by

$$PDE_3 \stackrel{def}{=} (csp\_to\_pde, \top).PDE_4 + (ca\_to\_pde, \top).PDE_4.$$

The rates $\top$ here reflect that for the PDE both activities are passive, and their rates are determined by their cooperation partners—the C/S Provider and the CA respectively.

Hiding: The hiding combinator "/" provides type abstraction, without affecting the duration of the activity. In P/L all activities whose action types are in L appear as the "private" type τ but their rates are unaffected. For example, after receiving the content, the PDE will forward it to the user interface and then go back to its initial state, which is modelled by

$$PDE_4 \stackrel{def}{=} (pde\_user\_interface, r_{pde\_user\_interface}).PDE_1 .$$

The activity $pde\_user\_interface$ may be hidden from the outside, and this can be expressed by

$$PDE_4 \stackrel{def}{=} (pde\_user\_interface, r_{pde\_user\_interface}).PDE_1/\{pde\_user\_interface\},$$

which is equivalent to

$$PDE_4 \stackrel{def}{=} (\tau, r_{pde\_user\_interface}).PDE_1 .$$

The precedence of the above combinators is as follows:

$$\text{hiding} > \text{prefix} > \text{cooperation} > \text{choice},$$

that is, hiding enjoys the highest precedence, prefix comes next followed by cooperation, and choice has the lowest precedence. We can use brackets to clarify the grouping as in elementary algebra and to force a different precedence. The syntax may be formally introduced by means of the following grammar:

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

where S denotes a *sequential component* and P stands for a *model component* which executes in parallel. C represents a constant which denotes either a sequential component or a model component.

2.3.3 Execution strategies, apparent rate and operational semantics

The dynamic behaviour of a PEPA model whenever more than one activity is enabled, is governed by a strategy called the *race condition*. In this strategy, all the enabled activities compete

with each other but only the fastest succeeds to proceed. The probability of an activity winning the race is given by the ratio of the activity rate of that activity to the sum of the activity rates of all the activities engaged in the race. This gives rise to an implicit probabilistic choice between actions dependent of the relative values of their rates. Therefore, if a single action in a system has more than one possible outcome, we may represent this action by more than one activity in the corresponding PEPA model. For example, AM_4 can perform the action $am\_assimilation$ with the rate $r_{am\_assimilation}$, and then behave as AM_5 or AM_9, with the probabilities p and $1-p$ respectively. This can be modelled as:

$$\begin{aligned} AM_4 \stackrel{def}{=} & (am\_assimilation, p \times r_{am\_assimilation}).AM_5 \\ & + (am\_assimilation, (1-p) \times r_{am\_assimilation}).AM_9. \end{aligned}$$

Here the component AM_4 has two separate activities of the same action type. To an external observer, the sum of the rates of the type $am\_assimilation$ in this component will be the same, that is $r_{am\_assimilation} = p \times r_{am\_assimilation} + (1-p) \times r_{am\_assimilation}$. This is called the apparent rate of $am\_assimilation$.

In the PEPA language, the *apparent rate* of action type α in a process P, denoted by $r_\alpha(P)$, is the overall rate at which α can be performed by P. It is defined as follows:

1. $r_\alpha((\beta, r).P) = \begin{cases} r & \text{if } \beta = \alpha \\ 0 & \text{if } \beta \neq \alpha \end{cases}$
2. $r_\alpha(P + Q) = r_\alpha(P) + r_\alpha(Q)$
3. $r_\alpha(P/L) = \begin{cases} r_\alpha(P) & \text{if } \alpha \notin L \\ 0 & \text{if } \alpha \in L \end{cases}$
4. $r_\alpha\left(P \underset{L}{\bowtie} Q\right) = \begin{cases} r_\alpha(P) + r_\alpha(Q) & \text{if } \alpha \notin L \\ \min(r_\alpha(P), r_\alpha(Q)) & \text{if } \alpha \in L \end{cases}$

If more than one activity of a given passive type can be simultaneously enabled by a component, each unspecified activity rate must also be assigned a weight to reflect the relative probabilities of the possible outcomes of the activities of that action type [Hil96]. For example, the component

$$P \stackrel{def}{=} (\alpha, w_1\top).P_1 + (\alpha, w_2\top).P_2$$

will behave as P_1 or P_2 with the probabilities $\dfrac{w_1}{w_1 + w_2}$ and $\dfrac{w_2}{w_1 + w_2}$ respectively, after the

passive action α is completed. The comparison and manipulation of unspecified activity rates are defined as:

$$\begin{aligned} r < w\top & \qquad \text{for all } r \in \mathbb{R}^+ \text{ and for all } w \in \mathbb{N} \\ w_1\top < w_2\top & \qquad \text{if } w_1 < w_2 \text{ for all } w_1, w_2 \in \mathbb{N} \\ w_1\top + w_2\top = (w_1 + w_2)\top & \qquad \text{for all } w_1, w_2 \in \mathbb{N} \\ \frac{w_1\top}{w_2\top} = \frac{w_1}{w_2} & \qquad \text{for all } w_1, w_2 \in \mathbb{N} \end{aligned}$$

We use $(\alpha, \top)$ to represent $(\alpha, 1\top)$, and assume that multiple instances have equal probabilities of occurring if no weights are assigned.

Operational semantics of a process algebra defines the rules of how processes evolve and how states transition. The formal structured *operational semantics* of PEPA is presented in Figure 2.3. These rules are to be interpreted as follows: if the transition(s) above the inference line can be inferred, then we can deduce the transition below the line. All rules presented in Figure 2.3 are straightforward and it is not necessary to give an explanation, except for the third one, i.e. the rule of the cooperation combinator. In this rule, the apparent rate of a shared activity α in the component $E \underset{L}{\bowtie} F$, i.e. $r_\alpha(E \underset{L}{\bowtie} F)$, is set to be $\min\{r_\alpha(E), r_\alpha(F)\}$, i.e. the smaller of the apparent rates of that action type in the components E and F. The action type α may have multiple activities which may result in different outcomes. The probability that the activity (α, r_1) (respectively, (α, r_2)) in E (respectively, F) occurs is $\frac{r_1}{r_\alpha(E)}$ ($\frac{r_2}{r_\alpha(F)}$). After completing (α, r_1) ((α, r_2)), the component E (F) behaves as E' (F'). So, assuming independence of choice in E and F, the probability of the transition $E \underset{L}{\bowtie} F \xrightarrow{(\alpha, R)} E' \underset{L}{\bowtie} F'$ is $\frac{r_1}{r_\alpha(E)} \frac{r_2}{r_\alpha(F)}$, and thus the rate of the shared activity is $R = \frac{r_1}{r_\alpha(E)} \frac{r_2}{r_\alpha(F)} \min(r_\alpha(E), r_\alpha(F))$.

Based on the operational semantics of PEPA, a PEPA model can be viewed as a *labelled multi-transition* system. In general a labelled transition system $(S, T, \{\xrightarrow{t} \mid t \in T\})$ is composed of a set of states S, a set of transition labels T and a transition relation $\xrightarrow{t}$ defined on $S \times S$ for each $t \in T$. In PEPA models, an action may represent and result in multiple system transitions. Thus, as pointed out in [Hil96], PEPA may be regarded as a labelled multi-transition system $(\mathcal{C}, \mathcal{A}ct, \{\xrightarrow{(\alpha, r)} \mid (\alpha, r) \in \mathcal{A}ct\})$, where $\mathcal{C}$ is the set of components, $\mathcal{A}ct$ is the set of activities and the multi-relation $\xrightarrow{(\alpha, r)}$ is given by the rules in Figure 2.3.

Prefix:

$$\frac{}{(\alpha, r).E \xrightarrow{(\alpha,r)} E}$$

Choice:

$$\frac{E \xrightarrow{(\alpha,r)} E'}{E + F \xrightarrow{(\alpha,r)} E'}, \qquad \frac{F \xrightarrow{(\alpha,r)} F'}{E + F \xrightarrow{(\alpha,r)} F'}$$

Cooperation:

$$\frac{E \xrightarrow{(\alpha,r)} E'}{E \underset{L}{\bowtie} F \xrightarrow{(\alpha,r)} E' \underset{L}{\bowtie} F}(\alpha \notin L), \qquad \frac{F \xrightarrow{(\alpha,r)} F'}{E \underset{L}{\bowtie} F \xrightarrow{(\alpha,r)} E \underset{L}{\bowtie} F'}(\alpha \notin L)$$

$$\frac{E \xrightarrow{(\alpha,r_1)} E' \quad F \xrightarrow{(\alpha,r_2)} F'}{E \underset{L}{\bowtie} F \xrightarrow{(\alpha,R)} E' \underset{L}{\bowtie} F'}(\alpha \in L), \quad \text{where} \quad R = \frac{r_1}{r_\alpha(E)} \frac{r_2}{r_\alpha(F)} \min(r_\alpha(E), r_\alpha(F)),$$

where $r_\alpha(E), r_\alpha(F)$, are the apparent rates of action of type α in the component E and F respectively.

Hiding:

$$\frac{E \xrightarrow{(\alpha,r)} E'}{E/L \xrightarrow{(\alpha,r)} E'/L}(\alpha \notin L), \qquad \frac{E \xrightarrow{(\alpha,r)} E'}{E/L \xrightarrow{(\tau,r)} E'/L}(\alpha \in L)$$

Constant:

$$\frac{E \xrightarrow{(\alpha,r)} E'}{A \xrightarrow{(\alpha,r)} E'}(A \stackrel{def}{=} E)$$

Figure 2.3: *Operational semantics of PEPA*

2.3.4 CTMC underlying PEPA model

The memoryless property of the exponential distribution, which is satisfied by the durations of all activities, means that there is a CTMC underlying any given PEPA model [Hil96]. By solving the matrix equation characterising the global balance equations associated with this CTMC using linear algebra, the steady-state probability distribution can be obtained, from which performance measures such as throughput and utilisation can be derived. Similarly the matrix may be used as the basis for transient analysis, allowing measures such as response time distributions to be calculated. In the next section, we will use the content adaptation example to illustrate how to derive performance measures from a PEPA model.

2.3.5 Attractive features of PEPA

The most attractive and important features which the PEPA language has whilst other existing performance modelling paradigms may not, are *compositionality*, *formality* and *abstraction* [Abo]. Compositionality divides a system into its subsystems with the associated interactions amongst them. Formality gives a precise meaning and description to all terms in the language. Abstraction builds up complex models from detailed components while disregarding the internal behaviour when it is unnecessary. For a brief comparison with queueing networks, Petri nets and their stochastic extensions, please refer to the following Table 2.1. A more detailed comparison can be found in [DHR95, HRRS01].

	Compositionality	Formality	Abstraction
Queueing Networks	Yes	No	No
Petri Nets and Extentions	No	Yes	No
PEPA	Yes	Yes	Yes

Table 2.1: *Comparison between PEPA and other paradigms*

2.4 Performance Measures and Performance Evaluation for Small Scale Content Adaptation Systems

This section will present the use of PEPA to analyse the performance of the mechanisms used to adapt content and services to the users' needs. The system is based, primarily, on the use of a personal assistant agent to specify and control what the user needs and constraints are in order

to receive a particular service or content. This interacts with a content adaptation mechanism that resides in the network. We will discuss what kind of performance measures are of interest and how to derive these measures from the system. Performance of the system, as we will see, depends upon an efficient negotiation, content adaptation, and delivery mechanism.

2.4.1 PEPA model and parameter settings

Based on the architecture and working cycle presented in the previous subsection, this section defines the PEPA model of the content adaptation system. The system model is comprised of four components, corresponding to the four major entities of the the architecture, i.e., the PDE, the AM, the CA and the C/S Provider. Each of the components has a repetitive behaviour, reflecting its role within the working cycle. There is no need to represent all aspects of the components' behaviour in detail, since the level of abstraction is chosen to be sufficient to capture the time/resource consuming activities. Below, PEPA definitions for the components are shown.

In Section 2.3.2, we have given the PEPA definition for the PDE. For convenience, we present it again.

PDE: The behaviour of the PDE begins with the generation of a request for content adaptation, represented as action $pde\_ext\_cs\_req$. The rate here reflects the expected rate at which the user will submit requests for content adaptation. The next event is to pass the request to the AM, $pde\_int\_cs\_req$, which is a synchronous activity. After that, the PDE waits for a response. The model reflects that there are two possible responses, by having two possible activities: receiving the content from the C/S Provider directly or receiving the adapted content from the CA, which are represented by csp_to_pde and ca_to_pde respectively. The rates $\top$ here indicate that for the PDE both activities are passive, and their rates are determined by their cooperation partners—the C/S Provider and the CA respectively—reflecting their relative probabilities. In each case the final action of the PDE is to send appropriate information to the user interface, $pde\_user\_interface$. After completing this action, the PDE goes back to the initial state.

$$
\begin{aligned}
PDE_1 &\stackrel{def}{=} (pde\_ext\_cont\_req, r_{pde\_ext\_cont\_req}).PDE_2 \\
PDE_2 &\stackrel{def}{=} (pde\_int\_cont\_req, r_{pde\_int\_cont\_req}).PDE_3 \\
PDE_3 &\stackrel{def}{=} (csp\_to\_pde, \top).PDE_4 \\
&\quad + \quad (ca\_to\_pde, \top).PDE_4 \\
PDE_4 &\stackrel{def}{=} (pde\_user\_interface, r_{pde\_user\_interface}).PDE_1
\end{aligned}
$$

AM: After receipt of a request from the PDE, $pde\_int\_cont\_req$, the AM asks for and receives the content/service context from the C/S Provider, represented as csp_cc_req and csp_cc_res respectively. Depending on the received information, there are two choices for the subsequent action, $am\_assimilation$. The rates of these two competing activities reflect their relative probabilities. Here the probabilities are equal, thus are 0.5. If the C/S Provider can offer the desired content without further adaptation, the AM requests the C/S Provider to provide the content to the PDE directly (am_cont_req) and then goes back to its initial state. Otherwise, the AM will request the context from the CA, ca_states_req. Based on the response from the CA (ca_states_res), an adaptation decision and plan will be made ($am\_decision$) and then be forwarded to the C/S Provider (am_adapt_plan). After that, the AM goes back to its starting state.

$$
\begin{aligned}
AM_1 &\stackrel{def}{=} (pde\_int\_cont\_req, \top).AM_2 \\
AM_2 &\stackrel{def}{=} (csp\_cc\_req, r_{csp\_cc\_req}).AM_3 \\
AM_3 &\stackrel{def}{=} (csp\_cc\_res, \top).AM_4 \\
AM_4 &\stackrel{def}{=} (am\_assimilation, \tfrac{1}{2} r_{am\_assimilation}).AM_5 \\
&\quad + \quad (am\_assimilation, \tfrac{1}{2} r_{am\_assimilation}).AM_9 \\
AM_5 &\stackrel{def}{=} (ca\_states\_req, r_{ca\_states\_req}).AM_6 \\
AM_6 &\stackrel{def}{=} (ca\_states\_res, \top).AM_7 \\
AM_7 &\stackrel{def}{=} (am\_decision, r_{am\_decision}).AM_8 \\
AM_8 &\stackrel{def}{=} (am\_adapt\_plan, r_{am\_adapt\_plan}).AM_1 \\
AM_9 &\stackrel{def}{=} (am\_cont\_req, r_{am\_cont\_req}).AM_1
\end{aligned}
$$

Similarly we can define the PEPA models for the CA and the C/S Provider respectively.

CA:

$$\begin{aligned}
CA_1 &\stackrel{def}{=} (ca_states_req, \top).CA_2 \\
CA_2 &\stackrel{def}{=} (ca_states_res, r_{ca_states_res}).CA_3 \\
CA_3 &\stackrel{def}{=} (csp_call_ca_adapt, \top).CA_4 \\
CA_4 &\stackrel{def}{=} (ca_adaptation, r_{ca_adaptation}).CA_5 \\
CA_5 &\stackrel{def}{=} (ca_to_pde, r_{ca_to_pde}).CA_1
\end{aligned}$$

C/S Provider:

$$\begin{aligned}
CSP_1 &\stackrel{def}{=} (csp_cc_req, \top).CSP_2 \\
CSP_2 &\stackrel{def}{=} (csp_cc_res, r_{csp_cc_res}).CSP_3 \\
CSP_3 &\stackrel{def}{=} (am_cont_req, \top).CSP_4 \\
&\quad + \quad (am_adapt_plan, \top).CSP_5 \\
CSP_4 &\stackrel{def}{=} (csp_to_pde, r_{csp_to_pde}).CSP_1 \\
CSP_5 &\stackrel{def}{=} (csp_call_ca_adapt, r_{csp_call_ca_adapt}).CSP_1
\end{aligned}$$

System Equation: The final part of the definition of the model is the system equation which specifies how the complete model is constructed from the defined components. It specifies how many copies of each entity there are present in the system, and how the components interact, by forcing cooperation on some of the activity types. For our model the system equation is as shown below, where M represents the number of independent copies of the PDE in the system, which is a variable of some of our experiments. Similarly, N, P, and Q represent the number of copies of the AM, CA and C/S Provider respectively. Here N, P, and Q are set to one in this chapter, reflecting that there is only one AM, CA and C/S Provider in the model.

$$PDE_1[M] \underset{L_1}{\bowtie} \left(\left(AM_1[N] \underset{L_2}{\bowtie} CA_1[P] \right) \underset{L_3}{\bowtie} CSP_1[Q] \right),$$

where

$$\begin{aligned}
L_1 &= \{pde_int_cont_req, ca_to_pde, csp_to_pde\}, \\
L_2 &= \{ca_states_req, ca_states_res\}, \\
L_3 &= \{csp_cc_req, csp_cc_res, am_cont_req, am_adapt_plan, csp_call_ca_adapt\}.
\end{aligned}$$

As in all quantitative modelling it is important that the parameters used within the model are

Action	Description	Duration	Rate
$pde\_ext\_cont\_req$	user inputs an C/S request	1000	1
$pde\_int\_cont\_req$	PDE forwards the internal C/S request to AM	60	16.7
$pde\_user\_interface$	PDE forwards the adapted C/S to user's interface	83.3	12
$am\_assimilation$	AM assimilates the contexts	333.3	3
am_cont_req	AM forwards the content request to C/S Provider	60	16.7
$am\_decision$	AM makes an adaptation decision	333.3	3
am_adapt_plan	AM forwards the adaptation plan to C/S Provider	60	16.7
ca_states_req	AM asks for CA's states	60	16.7
ca_states_res	CA transmits information to AM	60	16.7
$ca\_adaptation$	CA's adaptation process	1000	1
ca_to_pde	CA transmits the adapted content to PDE	150	6.7
csp_cc_req	AM asks for C/S Provider's context	40	25
csp_cc_res	C/S Provider submits the context to AM	40	25
$csp\_call\_ca\_adapt$	C/S Provider forwards the content to CA for adaptation	150	6.7
csp_to_pde	C/S Provider forwards the content to PDE	150	6.7

Table 2.2: *Parameter settings (unit of duration: millisecond)*

as realistic as possible if the analysis is to generate useful results. In our model each of the activities in the model must be assigned an appropriate activity rate. In order to do this we set similar parameter values to the published measurement results in the literature [CCC05a, CCC05b, CCL05], which are based on the real implementation of some experimental system. The resulting parameter values are shown in Table 2.2, together with the intuitive explanation of each parameter. Note the rate represents how many activities can be completed in unit time, which in our case is one second. The final additional parameter is the number of independent PDE entities active within our system. In these initial experiments we assume that this parameter has value one, unless otherwise stated. Experiments in this chapter are conducted using the PEPA Eclipse Plug-in and associated tools. More details on these tools can be found at `http://www.dcs.ed.ac.uk/pepa`.

2.4.2 Performance measures and performance evaluation: throughput and utilisation

In the following, we will discuss the performance measures of interest in the content adaptation system: adaptation throughput, utilisation efficiency, and response time. Moreover, a performance evaluation of the system will be presented.

As we have mentioned, for any given PEPA model, there is an underlying CTMC. Assume the state space of this CTMC is S, and the infinitesimal generator is Q, then the steady-state probability distribution π can be found through the *global balance equation*

$$Q\pi = 0 \tag{2.1}$$

with the *normalisation condition*

$$\sum_{\mathbf{s}\in S} \pi(\mathbf{s}) = 1, \tag{2.2}$$

where $\pi(\mathbf{s})$ is the steady-state probability that the model is in the state $\mathbf{s} \in S$.

If the states of a Markov chain are assigned rewards, i.e. a reward structure is associated with this Markov chain, then this Markov chain is called a Markov reward model [How71]. The performance measures of interest can be represented by using this kind of Markov reward structure [CH96]. For example, for the CTMC underlying the given PEPA model, we define a function $\rho : S \to \mathbb{R}^+$, which associates a reward $\rho(\mathbf{s})$ to a state $\mathbf{s} \in S$. A performance measure such as throughput or utilisation can be then calculated as the average reward R:

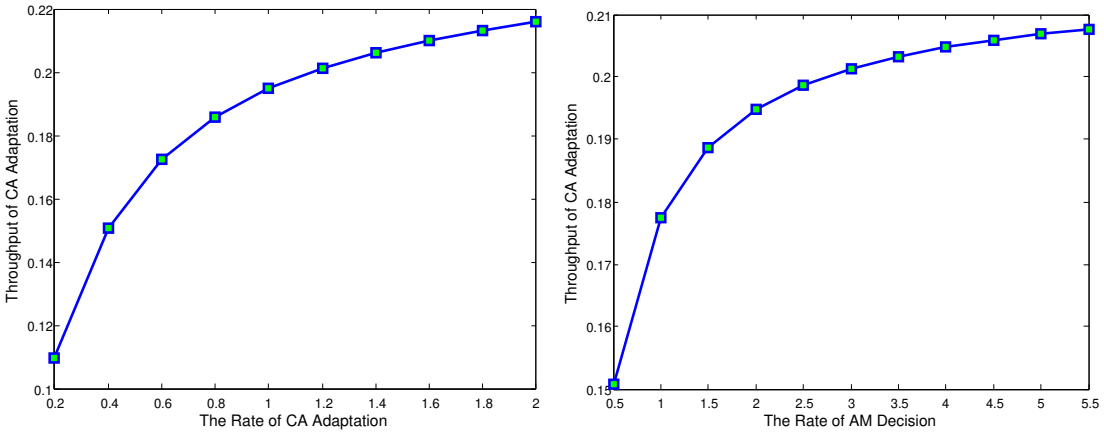
$$R = E[\rho] = \sum_{\mathbf{s}\in S} \rho(\mathbf{s})\pi(\mathbf{s}).$$

Following [CCE+03, Hil96], the definition of *throughput* of an activity is the average number of activities completed by the system during one unit time (one second). We are interested in the throughput of the activity "$ca\_adaptation$" in the content adaptation system, since it reflects how fast the system runs the adaptation. According to the PEPA model, only CA_4 can perform this activity. Therefore, the population of CA_4 and the rate of this activity (notice that the rate indicates the average number of the activity completed by the component in a unit time), i.e. $r_{ca\_adaptation}$, determines the reward function of the throughput of $ca\_adaptation$: $\rho(\mathbf{s}) = \mathbf{s}[CA_4]r_{ca\_adaptation}$, where $\mathbf{s}[CA_4]$ represents the number of CA_4 in the state $\mathbf{s}$. In particular, if $\mathbf{s}[CA_4] = 0$, then the reward $\rho(s)$ is zero. The average throughput of $ca\_adaptation$ is thus given as:

$$Thr(ca\_adaptation) = E[\rho] = \sum_{\mathbf{s}\in S} \pi(\mathbf{s})\mathbf{s}[CA_4]r_{ca\_adaptation}.$$

Obviously, this throughput is affected by the steady-state probability distribution π and the

activity rate $r_{ca\_adaptation}$ (notice that $r_{ca\_adaptation}$ can also affect π). Intuitively, increasing the rate of adaptation or decreasing its delay can improve its throughput. From Figure 2.4, it can be observed that the throughput of adaptation is sensitive not only to its own rate but also to the rate of AM's decision when the latter approaches lower rates.



(a) Impact of the Adaptation rate on its throughput

(b) Impact of the AM decision rate on the adaptation throughput

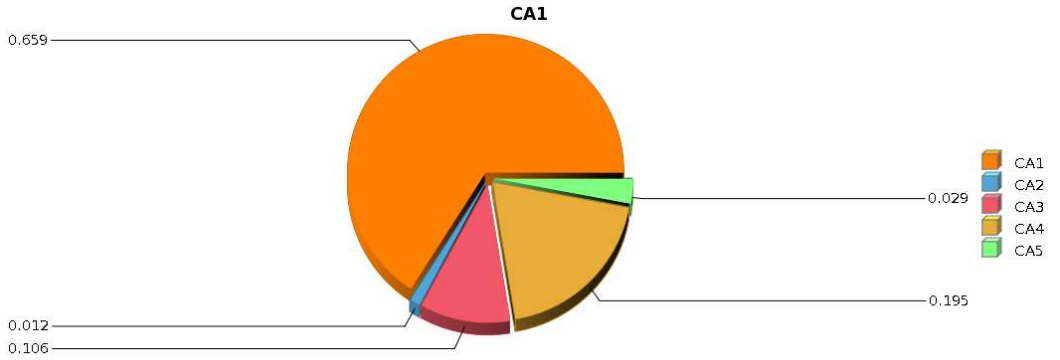
Figure 2.4: *Throughput of the CA adaptation* ($(M, N, P, Q) = 1$)

The system manager's interests include not only the speed of the system's operation but also the system's utilisation efficiency. Increasing the adaptation rate speeds up the running of the whole system. However this does not imply that the system is more efficiently utilised. To illustrate, we introduce the definition of *utilisation*, i.e. the probability that a component stays in a local state. For the CA, there are five local states, CA_i, $i = 1, \cdots, 5$. CA_1 is the state of waiting for and receiving the context requirement from the AM. The utilisation of the idle state CA_1 of the CA in a state $\mathbf{s} \in S$, is defined as the proportion of the population of CA_1 in the total population of the CA, N_{CA}, that is, the corresponding reward function is $\rho(\mathbf{s}) = \dfrac{\mathbf{s}[CA_1]}{N_{CA}}$. Thus the average utilisation of CA_1 is defined as

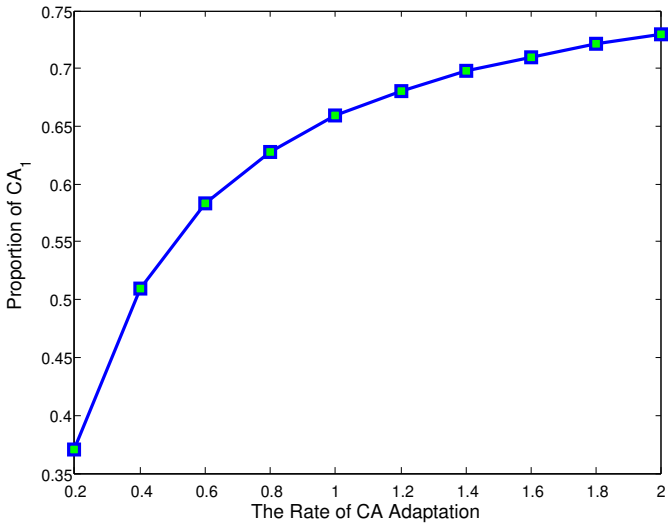
$$\text{Util}(CA_1) = E[\rho] = \sum_{\mathbf{s} \in S} \left(\frac{\mathbf{s}[CA_1]}{N_{CA}} \right) \pi(\mathbf{s}). \tag{2.3}$$

If there are no synchronisations, the probabilities of the CA in its five states are proportional to their average time for completing the respective activities. In this case we would expect CA_1's occupancy to be small. However, the CA has to synchronise with other events, which means that CA_1 corresponds to the longest time in this component with a proportion of about

65.9% (see Figure 2.5 (a)). Moreover, the smaller the adaptation's duration is, the bigger CA_1's proportion (see Figure 2.5 (b)). When the CA operates without any synchronisation delays, CA_1's proportion could be 4.23% (or $\frac{60}{60+60+150+150+1000}$), thus the CA has sufficient capacity to serve more requests, and be better utilised.



(a) State occupancy of the CA



(b) Proportion of CA idle state occupancy vs adaptation rate

Figure 2.5: *Utilisation of the CA*

We should point out that for different communication and computer systems, the performance measures of interest may be different. For example, in the papers [WLH09b, WLH09a] which present the performance evaluation of mobile networks by PEPA, the metrics of interest include handover rate and blocking probability. In [RV06, KV05], the authors are interested in using collision probability and channel utilisation to measure the performance of 802.11 Ad-Hoc networks. However, these metrics can nevertheless be derived through the Markov reward

approach. For our content adaptation system, adaptation throughput and system utilisation efficiency are of interest to the system manager while response time is important to the user. However, in general the measure of response time cannot be derived through the reward approach. In the next subsection, we will show how to derive the response time from the content adaptation model.

2.4.3 Performance measures and performance evaluation: response time

The response time considered here is the duration between the input of a request to the system and the return of the (adapted) content. This performance metric has a major impact on users' satisfaction, since it measures the users' waiting time for the desired content while reflecting the operation speed of the system.

The service corresponding to the input request can be classified into two cases, according to whether the CA's adaptation is needed. See the following two "service flows", which reflect the working cycle of the system. In service flow 1, the AM asks the C/S Provider to send the content to the PDE directly, without the CA's participation. In service flow 2, the CA's adaptation and the interactions between the CA and the other entities are needed. Of course, service flow 2 costs more time. As we mentioned in the PEPA definition of the AM, the probabilities of these two flows being chosen are set to be equal.

Service flow 1

start activity $=$
$pde\_int\_cont\_req \rightarrow csp\_cc\_req \rightarrow csp\_cc\_res \rightarrow am\_assimilation \rightarrow am\_cont\_req \rightarrow csp\_to\_pde$
$=$ stop activity

Service flow 2

start activity $=$
$pde\_int\_cont\_req \rightarrow csp\_cc\_req \rightarrow csp\_cc\_res \rightarrow am\_assimilation \rightarrow ca\_states\_req \rightarrow ca\_states\_res \rightarrow am\_decision \rightarrow am\_adapt\_plan \rightarrow csp\_call\_ca\_adapt \rightarrow ca\_adaptation \rightarrow ca\_to\_pde$
$=$ stop activity

Clearly, we cannot derive the response time from a PEPA model through the reward approach,

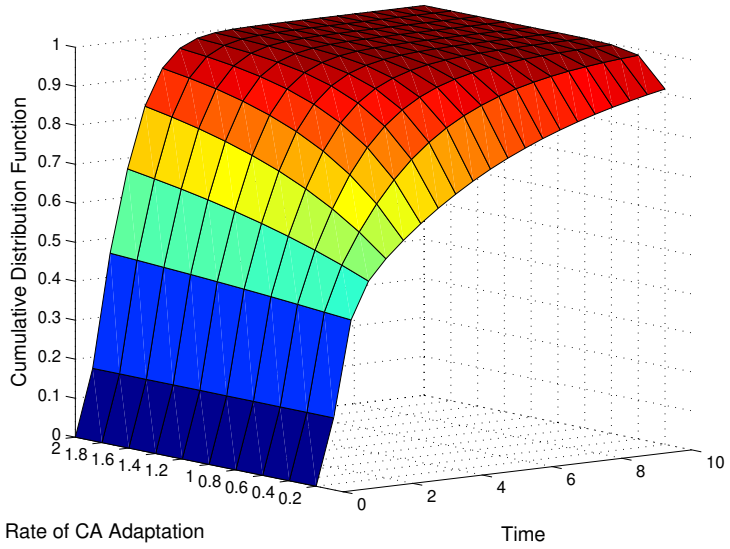
since the response time is a random variable, which is specified by the duration of the service flow and thus is related to multiple states rather than a single one. However, an associated tool for PEPA, ipc/Hydra [BDGK, BK04], can help to obtain the response time based on the state space and the equilibrium probability distributions. In this chapter, all experiments related to the response time are carried out using this software.

The cumulative distribution functions of the system's response time under our previous parameter setting are demonstrated in Figure 2.6. Figure 2.6 (a) shows that the response time has a strong dependence on the content adaptation rate, when the adaptation rate is less than one, corresponding to an average adaptation time of one second. Conversely, Figure 2.6 (b) shows that the AM's rate of decision making has little effect on the response time of the system, unless the rate is less than 1.5. From a system perspective, if complexity in the AM can be traded off with complexity in the CA, perhaps by a more involved process of selecting adaptation parameters, the response time could be lowered.

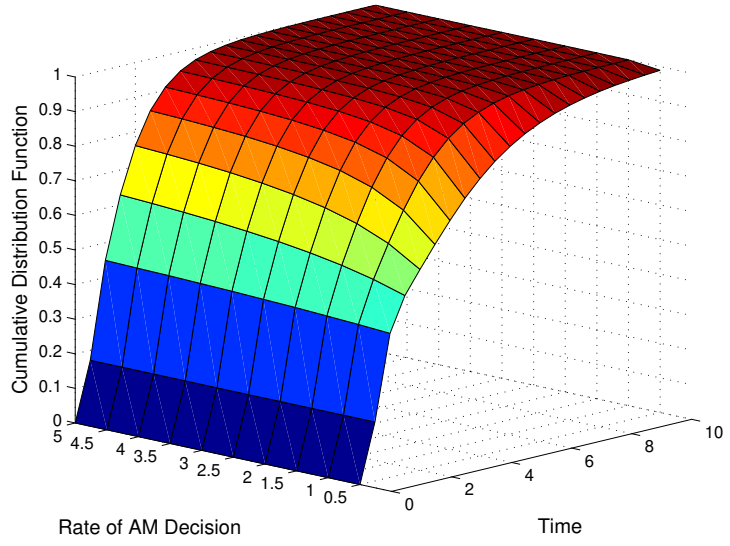
The effect of increasing the adaptation rate on system performance has been illustrated in Figure 2.9 (a). As adaptation rate increases, the adaptation throughput increases, while the response time and the utilisation efficiency decrease. Thus, an improved user experience, as measured by response time, can be obtained through improving the adaptation rate, at the expense of increased redundancy in the CA.

A full network would comprise many PDEs that co-exist and share system resources. This has the effect of changing the load on system components, altering throughput and waiting times. Figure 2.8 (a) shows that the CA's waiting time decreases as the number of PDEs increases, due to more frequent requests being received, while Figure 2.8 (b) illustrates that the throughput of adaptation is increasing, due to the number of requests that are being served. For example, four PDEs result in more than 0.45 adaptations per second being completed compared with 0.2 adaptations in the case of one PDE.

On the other hand, more PDEs, which make other components more busy, results in longer user waiting times or the system's response time in general (see Figure 2.7). Figure 2.9 (b) illustrates the effect of increasing the number of PDEs being supported by a system on adaptation throughput, response time and utilisation of the CA. It shows that there is a trade-off between the response time that can be achieved and the load placed on the adaptation process in terms of achieved throughput and utilisation. This information can be used in the planning process to



(a) Response time vs adaptation rate



(b) Response time vs AM decision rate

Figure 2.6: *Response time as a function of adaptation rate and AM decision rate*

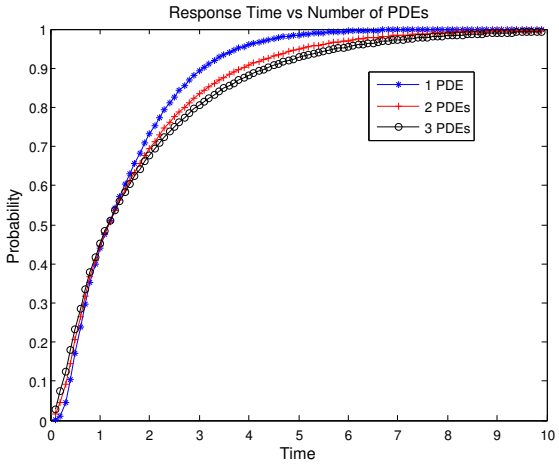
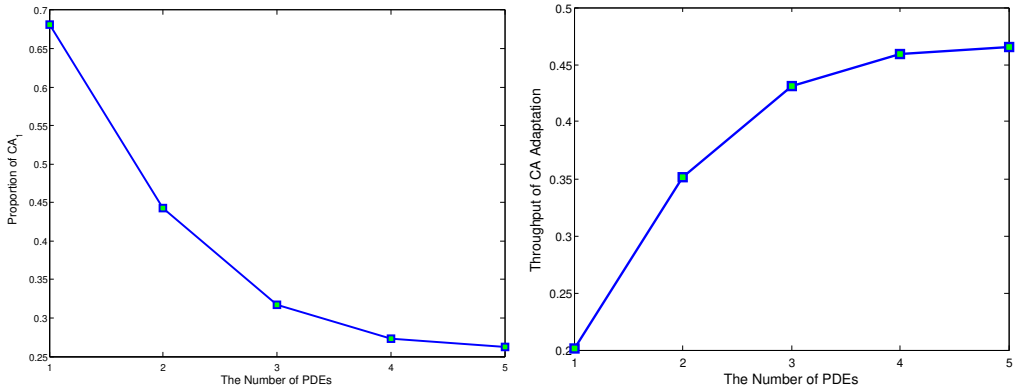
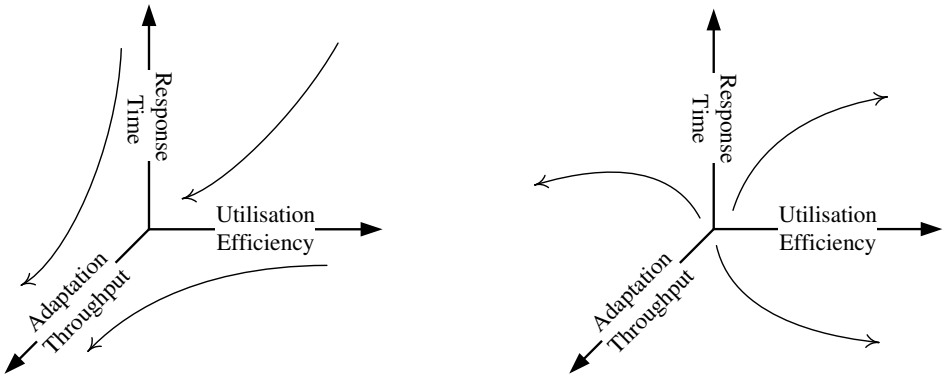


Figure 2.7: *Response time changes with the number of PDEs*



(a) Utilisation efficiency vs the number of PDEs (b) Adaptation throughput vs the number of PDEs

Figure 2.8: *Throughput and utilisation changes with the number of PDEs*



(a) Impact of increasing the adaptation rate (b) Impact of increasing the number of PDEs

Figure 2.9: *Adaptation rate and the number of PDEs' impact on the system performance*

appropriately dimension a system to achieve its potential.

As illustrated by Figure 2.9, as well as the reward equation, $R = \sum_{\mathbf{s} \in S} \rho(\mathbf{s})\pi(\mathbf{s})$, the system performance is affected by the activity rates and the populations of components (in the reward equation, these two factors determine the steady-state probability distribution π). But we do not know how these factors explicitly and analytically impact on the performance. As we will see in Chapter 7, the performance is governed by a set of nonlinear algebra equations in the sense of approximation. Based on those nonlinear equations, we clearly know how the system will perform when the rates or populations change. Since these equations can be easily derived according to the corresponding PEPA model, the performance optimisation based on them will be much simpler and more convenient.

2.4.4 Enhancing PEPA to evaluate large scale content adaptation systems

The previous subsections have presented the performance analysis for the content adaptation framework proposed by the Mobile VCE. Furthermore, for the architecture design of the component adaptation manager, we have shown in [ADLM08] that the adaptation gateway of this entity is a potential bottleneck for the system, suggesting that concurrent handling of adaptation requests should be adopted. In the aforementioned papers [Dey00, CCC05a, CCC05b, CCL05], some performance analyses, especially the distribution of the response time have also been presented.

However, these analyses are based on small scale systems, i.e., not many users and servers are taken into consideration. A realistic system may comprise a large number of users and entities. Modelling systems at large scale can provide some insights about the system performance which cannot be obtained via small scale modelling. For example, as we will see from Figure 7.7(a) and Figure 7.7(b) in Chapter 7, the throughput of activities will remain flat after some critical points in terms of the numbers of users, reflecting no improvement of performance after the system resources are fully utilised. This important fact cannot be illustrated by results such as those shown in Figure2.8 (b), which are based on small scale modelling.

Performance evaluation for large scale content adaptation systems has been considered in [DHL]. In this paper, the Monte Carlo simulation method is adopted to derive the response time from the system model. But the computational complexity, mainly in terms of the convergence speed of simulation, is very high and thus it is not possible to make real-time performance monitoring

or prediction available.

In order to efficiently carry out performance evaluation as well as qualitative validation for large scale content adaptation systems, we have to find appropriate approaches. In the following chapters we will present our enhancement and investigations for PEPA. The new techniques will be utilised to assess large scale content adaptation systems, which will be presented in Chapter 7.

2.5 Related work

In the past fifteen years, many techniques dealing with the state-space explosion problem have been developed for performance modelling paradigms including the stochastic process algebra PEPA. This section presents a brief overview on these techniques, with a focus on PEPA.

2.5.1 Decomposition technique

For a Markov process with large state space, it is often not possible to get the exact solution or the equilibrium distribution because there are not enough time and storage resources to generate the states and the transition rates, or to solve the associated balanced equation. One approach proposed to deal with this problem is that the solution of the single system can be formed by a set of solutions which correspond to a set of subsystems. This approach is introduced by Simon and Ando in [SA61] and called the *decomposition / aggregation* approach to the solution of CTMC. A variety of decompositional techniques have been proposed to aid in the solution of large Markov processes. According to the paper [Hil98] written by Hillston, these decompositional techniques can be classed into two categories: product form solution and aggregated decomposed solution. The following introduction to these techniques is mainly based on the paper [Hil98].

2.5.1.1 Product form solutions

If a Markov process can be decomposed into subsystems which behave as if they are statistically independent, and the equilibrium distribution of this Markov process can be written as a product of the equilibrium distributions of those subsystems, then this Markov process is said to have *product form solution* or *product form distribution*. Clearly product form solutions are

an efficient mechanism in deriving the performance from a Markovian model since there is no need to generate the entire state space of the model.

Product form distributions have been widely used in the analysis of queueing networks [BCMP75] and Petri nets. Based on earlier work on product form criteria for stochastic Petri nets [BS94, HLT89, HT91], preliminary work on deriving product form criteria for stochastic process algebras such as PEPA has been reported by Sereno in [Ser95]. In this paper, the method to characterise the class of models which have a product form is based on the routing processes, and relies on vector representations of the state and the action of a model. In this approach, the routing process is a defined Markov chain in which the states correspond to the actions of the PEPA model, and its balance equations correspond to the traffic equations of a queueing network. As pointed out in [Ser95], if the state space of this process can be partitioned into equivalence classes of enabling actions, then a product form solution exists.

A product form in the context of stochastic Petri nets, where the decomposition is carried out over subnets, has been investigated by Lazar and Robertazzi in [LR91]. The results in [LR91] are generalised by Boucherie in [Bou94] to characterise a class of Markov processes. The processes belonging to this class can be formed as the product of a set of Markov processes which compete over a set of resources. In [HT99], Hillston and Thomas characterise this class of Markov processes in the PEPA language: these models consist of independent components, which give rise to the constituent Markov processes of the underlying Markov processes, and these components are connected indirectly through synchronisation with resource components. In this paper, the cases in which these models exhibit a product form solution have been identified, so that the components of the model are solved in isolation, these partial solutions subsequently being combined to give a solution of the complete model.

A new derived combinator for PEPA has been presented and used to construct models which have an insensitive structure by Clark and Hillston in the paper [CH02]. This structure is characterised by an underlying CTMC that is *insensitive*, that is, its steady-state distribution depends on the distribution of one or more of the random variables representing residence in a state only through their mean. In this structure, a particular set of activities are therefore not necessarily assumed to be exponentially distributed. The identified model structure has a product form solution, which does not match the previous criteria for the currently known stochastic process algebra product form classes [CH02].

Not all PEPA models have product form solution, because the necessary structural conditions are stringent. Some classes of models have been transformed into new models which are product form, based on modifications of the PEPA expressions representing them [TH00].

Product-form solutions in Markovian process algebras such as PEPA can be constructed using the Reversed Compound Agent Theorem (RCAT), as shown by Harrison in [Har03]. The RCAT is a compositional result used to determine the reversed processes. From a reversed process, a product form solution for the joint state probabilities follows directly. Therefore, the RCAT provides an alternative methodology for finding product-form solutions in PEPA. The RCAT has been generalised by the same author in two ways in [Har04]. The first generalisation was, by relaxing two conditions of the original RCAT, to yield a more general result that applies to a wider class of concurrent systems. Another generalisation was "obtained and used to derive the equilibrium state probabilities in a similar staged queue with processor sharing queueing discipline", which lead to a *non-product form* solution for a class of queueing networks with global state-dependence. Some further investigation and application of non-product forms can be found in [Har06] and [Tho06]. More recently, a type of product-form was obtained by Fourneau *et al.* in [FPS07] for a pair of CTMCs, expressed as stochastic automata networks, which did not have synchronised transitions but in which the rate of a transition in either chain depended on the current state of the other chain. The conditions derived in [FPS07] for a product-form to exist at equilibrium, have been obtained alternatively in [Har09] using a special case of RCAT.

2.5.1.2 Aggregated decomposed solutions

In addition to the product form solution, there is another decomposed solution technique that is suggested by the stochastic process algebra model structure, called the *aggregated decomposed solution*. This approach involves a stochastic representation of the interactions between components, and the solution of the single model is obtained by a set of solutions of submodels using an aggregated version of the original model.

Work on time scale decomposition in PEPA, based on Courtois's *near complete decomposability* [Cou77], has been presented by Hillston and Mertsiotakis in [HM95]. Time scale decomposition is a popular aggregated decomposition technique, which decomposes a CTMC so that short term equilibrium is reached within single partitions, and partition changes occur only rarely as the process approaches its long term equilibrium [SA61, HM95]. The work re-

ported in [HM95] is inspired by related work on time scale decomposition of stochastic Petri nets [ARI89, BT93], and relies on a classification of all actions relative to some threshold rate. Later work to tackle the problem of hybrid components which enable both fast and slow actions is presented in [Mer98, Mer97]. The slow behaviour of the hybrid was extracted into a separate shadow component, making the original component passive with respect to these actions.

Decision free processes is another approach to the decomposition of a class of stochastic process algebra models. In this approach, the model is partitioned into components, and they are then recombined so that one component is fully represented while the other is reduced to a minimal form, usually consisting of a single transition. For the work about the application of this approach to throughput approximation, see [Mer98, MS96, MS97].

There are several other kinds of decomposition techniques including the technique based on the notion of *near-independence* and the technique of *decomposition via synchronisation points*. Components are considered by Ciardo and Trivedi to be near-independent if they operate in parallel and rarely interact [CT91]. Following the basic idea that near-independent components can be solved independently in conjunction with a graph model which represents the dependencies, a near-independence based decomposition technique was proposed for stochastic reward nets [CT91]. It has been suggested by Bohnenkamp and Haverkort in [BH97] that this technique could be adapted for stochastic process algebra models. In the proposed approach the dependence between parallel components was recognised as the actions on which they cooperate. In [BH99] Bohnenkamp and Haverkort considered, within a stochastic process algebra framework, a class of models in which there was a fixed number of sequential processes running in parallel and synchronising on the same global set of actions. Within this class of models their solution technique is exact with respect to throughput and local steady-state probabilities.

2.5.2 Tensor representation technique

Another technique which has been taken to exploit the model compositionality is the use of Kronecker algebra. Kronecker algebra representations were first developed by Plateau in the context of stochastic automata networks [Pla84, Pla85], to analytically represent the generator matrix of the Markov process underlying a stochastic automata network model. This technique can relieve the state-space explosion problem arising in the numerical solution of Markov chains since the solution can be achieved via this tensor expression of submatrices and the complete matrix does not need to be generated. More recently, Kronecker-based solution techniques

have been developed for various Petri net based formalisms, see [Don94,Kem96,CM99,DK00].

Analogous to the representation of a stochastic automata network, it has been demonstrated by Hillston and Kloul in [HK01] that PEPA models can be represented analytically using Kronecker algebra and solved without constructing the complete generator matrix. Furthermore, this Kronecker representation has been combined with the aggregation technique to deal with larger models [HK07]. However, as pointed out in [HK07], these techniques "can be viewed as shifting the problem rather than avoiding it, in the sense that there are still fundamental limits on the size of model which can be analysed".

In addition, this tensor representation technique has also been applied to an extension of PEPA (see [ER00]) and other stochastic process algebras such as Markovian process algebra [Buc94] and TIPP [RS94].

2.5.3 Abstraction and stochastic bound techniques

Smith discussed another way to analyse large scale PEPA models in his series of papers [Smi09a, Smi09b,Smi09c], i.e. using abstraction — constructing a smaller model that bounds the properties of the original. Abstract Markov chains [FLW06,KKLW07] and stochastic bounds [FLQ04, SD83] are two techniques used for producing bounded abstractions of a CTMC: the former can be used to bound transient properties, and the latter for various monotone properties such as the steady-state distribution [Smi09a]. These two techniques are originally specified and used in the context of Markov chains, but have been extended, based on a Kronecker representation for the generator matrix of a PEPA model [HK01], so that they can be applied compositionally to PEPA models [Smi09b]. An algorithm for constructing a compositional upper bound of a PEPA component has also been presented in [Smi09b].

2.5.4 Fluid approximation technique

The techniques reported above are based on the discrete state space. However, as the size of the state space is extremely large, these techniques are not always strong enough to handle the state-space explosion problem. To avoid this problem Hillston proposed a radically different approach in [Hil05a] from the following two perspectives: choosing a more abstract state representation in terms of state variables, quantifying the types of behaviour evident in the model; and assuming that these state variables are subject to continuous rather than discrete change.

This approach results in a set of ODEs, leading to the evaluation of transient, and in the limit, steady state measures.

An interpretation as well as a justification of this approximation approach has been demonstrated by Hayden in his dissertation [Hay07a]. In [Hay07a, HB08], generation of similar systems of coupled ODEs for higher-order moments such as variance has been addressed. Additionally, the dissertation [Hay07a] discusses how to derive stochastic differential equations from PEPA models.

More recently, some extensions of the previous mapping from PEPA to ODEs have been presented by Bradley *et al.* in [BGH07]. In particular, passive rates are introduced into the fluid approximation. In the recent paper [HB10], different existing styles of passive cooperation in fluid models are compared and intensively discussed. Moreover, a new passive fluid semantics for passive cooperation, which can be viewed as approximating the first moments of the component counting processes, has been provided, with a theoretical justification. The paper [BHM$^+$09] considers the application of this fluid approximation approach with modifications in the context of epidemiology. In this paper, the notions of side and self-loops are added to the activity matrix, and the rates are calculated differently, for the purpose of deriving from PEPA models the most commonly used ODEs in the context of epidemiology. In [Tri09] by Tribastone, a new operational semantics is proposed to give a compact symbolic representation of PEPA models. This semantics extends the application scope of the fluid approximation of PEPA by incorporating all the operators of the language and removing earlier assumptions on the syntactical structure of the models amenable to this analysis.

The fluid approximation approach has also been applied to timed Petri nets to deal with the state-space explosion problem [SR05, MRS06]. The comparison between the fluid approximation of PEPA models and timed continuous Petri nets has been demonstrated by Galpin in [Gal08]. This paper has established links between two continuous approaches to modelling the performance of systems. In the paper, a translation from PEPA models to continuous Petri nets and *vice versa* has been presented. In addition, it has been shown that the continuous approximation using PEPA has infinite server semantics. The fluid approximation approach has also been used by Thomas to derive asymptotic solutions for a class of closed queueing networks [Tho09]. In this paper, an analytical solution to a class of models, specified using PEPA, is derived through the ODE approach. It is shown that "this solution is identical to that used for many years as an asymptotic solution to the mean value analysis of closed queueing networks".

Moreover, the relationship between the fluid approximation and the underlying CTMCs for a special PEPA model has been revealed by Geisweiller *et al.* in [GHS08]: the ODEs derived from the PEPA description are the limits of the sequence of underlying CTMCs. It has been shown in [Gil05] by Gilmore that for some special examples the equilibrium points of the ODEs derived from PEPA models coincide the steady-state probability distributions of the CTMCs underlying the nonsynchronised PEPA models.

In addition, there are several papers which discuss how to derive response time from the fluid approximation of PEPA models. In [BHKS08], by constructing an absorption operator for the PEPA language, Bradley *et al.* allow general PEPA models to be analysed for fluid-generated response times. Clark *et al.* demonstrate in [CDGH08] how to derive expected passage response times using Little's law based on averaged populations of entities in an equilibrium state. This technique has been generalised into one for obtaining a full response-time profile computing the probability of a component observing the completion of a response at a given time after the initiation of the request, see [Cla09]. Moreover, an error in the passage specification in the approach taken in [BHKS08] has been uncovered and rectified in [Cla09] by Clark.

The ODE method associated with the PEPA language has demonstrated successful application in the performance analysis of large scale systems. In Harrison and Massink's paper [HM09], quantitative models of a class of ubiquitous systems, including a guidance system to assist out-patients in a hospital, are considered and analysed using PEPA and the associated ODE approximation approach. The analyses provide insight into the impact of a ubiquitous system design on the congestion experienced by users in different traffic situations. This paper shows that "the ODE approach is an attractive alternative to simulation to explore the effect on visitor flows for different design options during early design phases".

Zhao and Thomas's paper [ZT08] considers a PEPA model of a key distribution centre. By combining successive internal actions into a single action with a modified rate, the system behaviour is approximated by a simpler model, a queueing network model which gives explicit performance measures. The fluid approximation was derived from the simplified models and compared to the queueing approximation in [TZ08, ZT09]. A limitation of the fluid approximation approach has been pointed out in [ZT09]: not all desired metrics can be obtained.

The work on the fluid approximation of PEPA reported above mainly deals with some extensions to make this approach more applicable or demonstrates its applications in some specific

areas. However, there are not many discussions on the fundamental problems, such as the existence, uniqueness, boundedness and nonnegativeness of the solution, as well as the its asymptotic behaviour as time tends to infinity, and the relationship between the derived ODEs and the underlying CTMCs for general PEPA models. As for applications, the basic problem of what kind of performance metrics can and cannot be derived through this approach, has also not been discussed generally and in detail. This thesis will focus on these topics and give answers to these problems.

2.6 Summary

In this chapter, we have introduced the content adaptation framework proposed by the Mobile VCE as well as the PEPA language. For small scale content adaptation systems based on this framework, the PEPA modelling and evaluation have been presented. Some parts of this work have been published as a joint work in the Journal of Wireless Personal Communications, see [DHL09]. This chapter has also presented a review of the related work on dealing with the state-space explosion problem in the current performance modelling paradigms, with an emphasis on the PEPA language.

Chapter 3

New Representation for PEPA: from Syntactical to Numerical

3.1 Introduction

This chapter defines a numerical presentation scheme for PEPA, for the purpose of applying powerful mathematical tools and methods for both qualitative and quantitative analysis of large scale PEPA models with repeated components. In this presentation scheme, a state of a system is represented by a numerical vector, with each entry being a non-negative integer recording the population of the components in the corresponding local derivative. Any transition between states is indicated by a labelled activity, or equivalently, a transition vector. All the transition vectors form an activity matrix, which captures the structural information of the given model. The average duration of labelled activities or transitions are specified by transition rate functions that are defined to capture the timing information of the system.

The fact that the structural and timing information is captured means that the representation can provide a platform to directly employ mathematical methods such as linear programming and differential equations to analyse a PEPA model. This chapter presents a technical preparation including the definitions and some fundamental investigations of the new representation, for the further study of PEPA appearing in the following chapters.

The remainder of this chapter is structured as follows. Section 3.2 gives the definition of the numerical vector form which is used to represent the states of PEPA models. The efficiency of this form is demonstrated. Labelled activities and activity matrices are defined in Section 3.3 while Section 3.4 defines transition rate functions. The consistency between these definitions and PEPA language is formulated as propositions. An algorithm is given for automatically deriving the labelled activities, activity matrix and transition rate functions from any PEPA model. Section 3.5 presents an initial discussion of how to utilise some efficient approaches to investigate PEPA models. Finally, a summary is presented in Section 3.6.

3.2 Numerical Vector Form

The numerical vector form, proposed by Hillston in [Hil96] as a model aggregation technique to represent the states of PEPA models with repeated components, can efficiently decrease the size the underlying state space from exponential to at most polynomial in the number of components. This section discusses the definition and efficiency of the numerical vector form.

3.2.1 State-space explosion problem: an illustration by a tiny example

Let us first consider the following tiny example. A User-Provider system is composed of two types of entities: $User$ and $Provider$. The communication between them is through a shared activity $task_1$. After $task_1$ is fired, $User_1$ becomes $User_2$ while $Provider_1$ becomes $Provider_2$ simultaneously. $User_2$ can fire $task_2$ and then go back to $User_1$. $Provider_2$ will become $Provider_1$ when $reset$ is fired. The PEPA model for this User-Provider system is illustrated below:

Model 1. ***PEPA Model of User-Provider System***

PEPA Definition for $User$:

$$User_1 \stackrel{\text{def}}{=} (task_1, a).User_2$$
$$User_2 \stackrel{\text{def}}{=} (task_2, b).User_1$$

PEPA Definition for $Provider$:

$$Provider_1 \stackrel{\text{def}}{=} (task_1, a).Provider_2$$
$$Provider_2 \stackrel{\text{def}}{=} (reset, d).Provider_1$$

System Equation:

$$\underbrace{User_1 || \cdots || User_1}_{M \text{ copies}} \underset{\{task1\}}{\bowtie} \underbrace{Provider_1 || \cdots || Provider_1}_{N \text{ copies}}$$

The system equation in a PEPA model specifies how many copies of each entity are presented in the system, and how the components interact, by forcing cooperation on some of the activity types. In Model 1, the exact numbers of independent copies of the $User$ and $Provider$, i.e. M and N respectively, are both considered as variables. According to the semantics of PEPA

originally defined in [Hil96], the size of the state space of the CTMC underlying Model 1 is 2^{M+N}. That is, the size of the state space increases exponentially with the numbers of the users and providers in the system. Consequently, the dimension of the infinitesimal generator of the CTMC is $2^{M+N} \times 2^{M+N}$. The computational complexity of solving the global balance equation to get the steady-state probability distribution and thus derive the system performance, is therefore exponentially increasing with the numbers of the components. When M and/or N are large, the calculation of the stationary probability distribution will be infeasible due to limited resources of memory and time. The problem encountered here is the so-called *state-space explosion* problem.

Not only quantitative but also qualitative analysis suffers from the state-space explosion problem. For example, it is impossible to explore the entire state space with limited resources to check whether there is a deadlock. In fact, even the derivation and storage of the state space can become a problem since it is very large.

Fortunately, a model aggregation technique introduced by Gilmore *et al.* in [GHR01] and by Hillston in [Hil05a] can help to relieve the state-space explosion problem.

3.2.2 Definition of numerical vector form

We have introduced in 2.3.3 that in PEPA the state representation is in fact a labelled multi-transition system. This is also termed the *derivation graph*. The usual state representation in PEPA models is in terms of the syntactic forms of the model expression. Thus, as pointed out in [Hil05a], each node in the derivation graph is a distinct syntactic form and each arc represents a possible activity causing the state change. Clearly, if the action types are ignored, the derivation graph can be considered to be the state transition diagram of a CTMC.

When a large number of repeated components are involved in a system, the state space of the CTMC can be large, as Model 1 shows. This is mainly because each copy of the same type of component is considered to be distinct, resulting in distinct Markovian states. The multiple states within the model that exhibit the same behaviour can be aggregated to reduce the size of the state space as shown by Gilmore *et al.* [GHR01] using the technique based on a vector form. The derivation graph is therefore constructed in terms of equivalence classes of syntactic terms. "At the heart of this technique is the use of a canonical state vector to capture the syntactic form of a model expression", as indicated in [Hil05a], "if two states have the same canonical state

vector they are equivalent and need not be distinguished in the aggregated derivation graph".

Rather than the canonical representation style, an alternative numerical vector form was proposed by Hillston in [Hil05a] for capturing the state information of models with repeated components. In the numerical vector form, there is one entry for each local derivative of each type of component in the model. The entries in the vector are the number of components currently exhibiting this local derivative, no longer syntactic terms representing the local derivative of the sequential component. Following [Hil05a], hereafter the term *local derivative* refers to the local state of a single sequential component, whereas *derivative* is used for a global state represented in its syntactic form. The definition of numerical vector form is given below.

Definition 3.2.1. *(**Numerical Vector Form [Hil05a]**). For an arbitrary PEPA model $\mathcal{M}$ with n component types $C_i, i = 1, 2, \cdots, n$, each with d_i distinct local derivatives, the numerical vector form of $\mathcal{M}$, $\mathbf{m}(\mathcal{M})$, is a vector with $d = \sum_{i=1}^{n} d_i$ entries. The entry $\mathbf{m}[C_{i_j}]$ records how many instances of the jth local derivative of component type C_i are exhibited in the current state.*

According to Definition 3.2.1, the discrimination between any two system states is characterised by the two system vectors. That is, if the two vectors are different then the two states are considered different, otherwise they are the same.

For a sequential component C_i with the local derivatives $C_{i_1}, C_{i_2}, \cdots, C_{i_{d_i}}$, define

$$\mathbf{m}(C_i) := \left(\mathbf{m}[C_{i_1}], \mathbf{m}[C_{i_2}], \cdots, \mathbf{m}[C_{i_{d_i}}]\right)^T,$$

where $\mathbf{m}$ is the system vector. So $\mathbf{m}(C_i)$ is a subvector of $\mathbf{m}$, which can be considered the restriction of the system vector in the context of the sequential component C_i. For convenience, $\mathbf{m}(C_i)$ is called the *state vector of the sequential component C_i* or the *state vector of component type C_i*, or just *C_i's vector* for short.

Remark 3.2.1. *Obviously $\mathbf{m}(C_{i_j}) \geq 0$ for each C_{i_j}. At any time, each sequential component stays in one and only one local derivative. So the sum of $\mathbf{m}(C_i)$, i.e. $\sum_{j=1}^{d_i} \mathbf{m}[C_{i_j}]$, specifies the population of C_i in the system. In other words, if there are M_i copies of the sequential component C_i in the system, then $\sum_{j=1}^{d_i} \mathbf{m}[C_{i_j}] = M_i$ for any system state.*

The entries in the system vector or a sequential component's vector are no longer syntactic terms representing the local derivative, but the number of components currently exhibiting this

local derivative. This model-aggregation technique can significantly reduce the size of the state space, i.e. the number of the system states.

3.2.3 Efficiency of numerical vector form

By adopting the new representation technique, the number of the states of the system can be reduced to only increase (at most) polynomially with the number of instances of the components. This fact is stated in the following Proposition 3.2.1. Before turning to this conclusion a lemma is introduced, which will be used in the proof of the proposition.

Lemma 3.2.1.

$$\#\left\{(a_1, a_2, \cdots, a_d) : \sum_{j=1}^{d} a_i = m, a_j \in \mathbb{Z}^+ (j = 1, 2, \cdots, d)\right\} = \begin{pmatrix} m+d-1 \\ d-1 \end{pmatrix}.$$

where $\#A$ is defined as the cardinality of the set A, i.e. the number of elements of A; $\mathbb{Z}^+$ is the set of nonnegative integers. This a well-known combinatorial formula. Readers are referred to Theorem 3.5.1 in [Bru98] for reference. Lemma 3.2.1 specifies how many solutions satisfy the condition

$$\begin{cases} a_1, a_2, \cdots, a_d \geq 0,\ a_j \in \mathbb{Z}\ (j = 1, 2, \cdots, d), \\ \sum_{j=1}^{d} a_j = m. \end{cases}$$

Apply Lemma 3.2.1 to the vector of C_i defined in the last subsection,

$$\mathbf{m}(C_i) = \left(\mathbf{m}[C_{i_1}], \mathbf{m}[C_{i_2}], \cdots, \mathbf{m}[C_{i_{d_i}}]\right)^T,$$

where, by Remark 3.2.1, provided there are M_i copies of C_i in the system, $\mathbf{m}(C_i)$ satisfies

$$\begin{cases} \mathbf{m}[C_{i_1}], \mathbf{m}[C_{i_2}], \cdots, \mathbf{m}[C_{i_{d_i}}] \in \mathbb{Z}^+, \\ \sum_{j=1}^{d} \mathbf{m}[C_{i_j}] = M_i. \end{cases} \tag{3.1}$$

Then there are at most $\begin{pmatrix} M_i + d_i - 1 \\ d_i - 1 \end{pmatrix}$ solutions, i.e. $\begin{pmatrix} M_i + d_i - 1 \\ d_i - 1 \end{pmatrix}$ states in terms of C_i in the system. The phrase "at most" used here is to reflect that $\begin{pmatrix} M_i + d_i - 1 \\ d_i - 1 \end{pmatrix}$ is an upper bound of the exact number of the states in terms of C_i, since the possible synchronisations in

the PEPA model have not been taken into account in the restrictions (3.1) and thus the current restrictions may allow extra freedom for the solutions. This argument leads to the following

Proposition 3.2.1. *Consider a system comprising n types of component, namely $C_1, \mathcal{C}_2, \cdots, C_n$, with M_i copies of the component of type C_i in the system, where C_i has d_i local derivatives, for $i = 1, 2, \cdots, n$. Then the size of the state space of the system is at most*

$$\prod_{i=1}^{n} \left(\begin{array}{c} M_i + d_i - 1 \\ d_i - 1 \end{array} \right) \leq \prod_{i=1}^{n} (m_i + d_i - 1)^{d_i - 1}.$$

Proof. For each component type C_i, there are at most $\left(\begin{array}{c} M_i + d_i - 1 \\ d_i - 1 \end{array} \right)$ solutions for the system vector. So the total number of states of the system is at most $\prod_{i=1}^{n} \left(\begin{array}{c} M_i + d_i - 1 \\ d_i - 1 \end{array} \right)$. Notice

$$\left(\begin{array}{c} M_i + d_i - 1 \\ d_i - 1 \end{array} \right) = \frac{(M_i + d_i - 1)(M_i + d_i - 2) \cdots (M_i + 1)}{(d_i - 1)!} \leq (M_i + d_i - 1)^{d_i - 1},$$

so

$$\prod_{i=1}^{n} \left(\begin{array}{c} M_i + d_i - 1 \\ d_i - 1 \end{array} \right) \leq \prod_{i=1}^{n} (M_i + d_i - 1)^{d_i - 1}.$$

This completes the proof. □

Proposition 3.2.1 gives an upper bound, e.g. $\prod_{i=1}^{n} (M_i + d_i - 1)^{d_i - 1}$, for the size of the state space of the given system. In this term, n and d_i are fixed in the PEPA definition for the system, while M_i can be considered as variables which refer to the numbers of the repeated components. This bound guarantees that the size of the state space increases at most *polynomially* with the number of instances of the components. Since the state space of a model is the foundation for both qualitative and quantitative analysis, the size of the state space mainly determines the computational complexity of the state-space-based algorithms for analysing the system. For example, since the current deadlock-checking algorithm needs to explore the entire space to find whether there is a deadlock, so the efficiency, of course, depends on the size of the state space. As for quantitative analysis, the performance measures such as throughput and utilisation, obviously suffer the size of the state space because they derive from the equilibrium probabilities distributed on each state in the space. Compared to the exponential increase of

state space size, the benefit brought by the new representation is very significant. An illustration of the state space represented in the numerical vector form for the previous example, is given in the next subsection.

3.2.4 Model 1 continued

This subsection presents the state space for Model 1 in Section 3.2.1:

$$\begin{aligned} User_1 &\overset{def}{=} (task_1, a).User_2 \\ User_2 &\overset{def}{=} (task_2, b).User_1 \\ Provider_1 &\overset{def}{=} (task_1, a).Provider_2 \\ Provider_2 &\overset{def}{=} (reset, d).Provider_1 \\ User_1[M] &\underset{\{task_1\}}{\bowtie} Provider_1[N]. \end{aligned}$$

In the model, there are two component types, $User$ and $Provider$; each has two local derivatives, $User_1$, $User_2$ and $Provider_1$, $Provider_2$ respectively. According to Definition 3.2.1, the system vector $\mathbf{m}$ has four entries representing the instances of components in the total four local derivatives, that is

$$\mathbf{m} = (\mathbf{m}[User_1], \mathbf{m}[User_2], \mathbf{m}[Provider_1], \mathbf{m}[Provider_2])^T .$$

Let $M = N = 2$, then the system equation of Model 1 determines the starting state:

$$\mathbf{m} = (M, 0, N, 0)^T = (2, 0, 2, 0)^T := \mathbf{s}_1.$$

After firing the synchronised activity $task_1$, then one instance of $User_1$ and one copy of $Provider_1$ become $User_2$ and $Provider_2$ simultaneously and respectively. Then the system vector becomes $\mathbf{s}_2 = (1, 1, 1, 1)^T$, reflecting each local derivative being occupied by one component. By enabling activities or transitions, all reachable system states can be manifested as follows:

$$\begin{aligned} &\mathbf{s}_1 = (2,0,2,0)^T, \quad \mathbf{s}_2 = (1,1,1,1)^T, \quad \mathbf{s}_3 = (1,1,2,0)^T, \\ &\mathbf{s}_4 = (1,1,0,2)^T, \quad \mathbf{s}_5 = (0,2,1,1)^T, \quad \mathbf{s}_6 = (2,0,1,1)^T, \\ &\mathbf{s}_7 = (0,2,0,2)^T, \quad \mathbf{s}_8 = (0,2,2,0)^T, \quad \mathbf{s}_9 = (2,0,0,2)^T. \end{aligned} \tag{3.2}$$

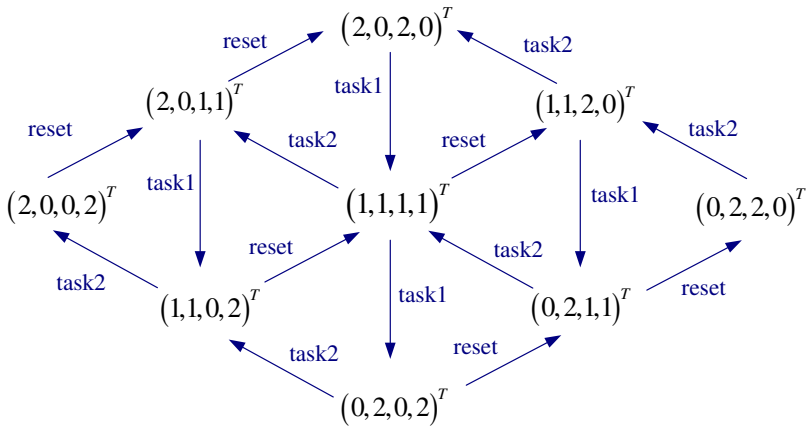


Figure 3.1: *Transition between States (a revised version of the one in [Hil05a])*

The transition relationship between these states is illustrated by Figure 3.1. As shown in (3.2) and Figure 3.1, there are nine states in the state space of Model 1, i.e. the size of the state space is 9. An upper bound of the size given by Proposition 3.2.1 is

$$\begin{pmatrix} M+2-1 \\ 2-1 \end{pmatrix} \begin{pmatrix} N+2-1 \\ 2-1 \end{pmatrix} = (M+1) \times (N+1).$$

Since $M = N = 2$ considered here, the upper bound is $(2+1) \times (2+1) = 9$, coinciding with the size of the state space. Therefore, the bound given in Proposition 3.2.1 is sharp and can be hit in some situations.

It is a significant improvement to reduce the size of the state space from $2^M \times 2^N$ to $(M+1) \times (N+1)$, without relevant information and accuracy loss. However, this does not imply there is no complexity problem. In practice, when hundreds of components exist in the system, for example $M = N = 999$, then $(M+1) \times (N+1) = 10^6$. This is still a large number, so that even the storage may become a problem with limited memory, let alone the analysis of the state space. The following table, Table 4.5, gives the runtimes of deriving the state space in several different scenarios. All experiments were carried out using the PEPA Plug-in (v0.0.19) for Eclipse Platform (v3.4.2), on a 2.66GHz Xeon CPU with 4Gb RAM running Scientific Linux 5. The runtimes here are elapsed times reported by the Eclipse platform.

If there are 400 users and 300 providers in the system, the Eclipse platform reports the error message of "Java heap space", while 400 users and 400 providers result in the error information

(M, N)	(300,300)	(350,300)	(400,300)	(400,400)
time	2879 ms	4236 ms	"Java heap space"	"GC overhead limit exceeded"

Table 3.1: *Elapsed time of state pace derivation*

of "GC overhead limit exceeded". These experiments show that the state-space explosion problem cannot be completely solved by just using the technique of numerical vector form, even for a tiny PEPA model. That is, in order to do practical analysis for large scale PEPA models in terms of both qualitative and quantitative aspects, we need to go further to investigate PEPA and develop associated efficient computational methods and tools. The study of these topics constitutes the content of the next three chapters, whilst some basic technical preparation for the further research is given in this chapter. In particular, in the following sections the *activity matrices* and *transition rate functions* are defined to capture, especially in numerical forms, the structural and timing information of PEPA models respectively.

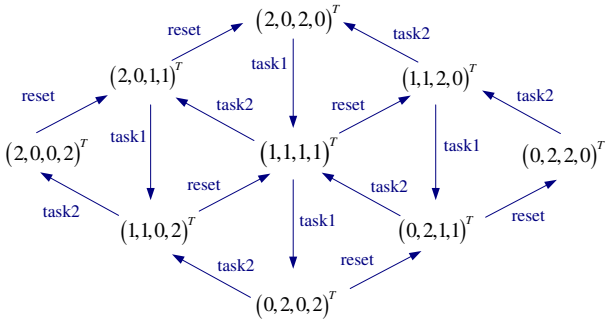
3.3 Labelled Activity and Activity Matrix

The numerical representation of system states can not only decrease the size of the state space and thus the associated computational complexity, but more importantly, it provides a numerical foundation for further qualitative and quantitative analysis for PEPA models. In the PEPA language, the transition is embodied in the syntactical definition of activities, in the context of sequential components. Since the consideration is in terms of the whole system rather than sequential components, the transition between these system states should be defined and represented. This section presents a numerical representation for the transitions between system states and demonstrates how to derive this representation from a general PEPA model.

3.3.1 Original definition of activity matrix

If a system vector changes into another vector after firing an activity, then the difference between these two vectors manifests the transition corresponding to this activity. Obviously, the difference is in numerical forms since all states are numerical vectors.

Consider Model 1 and its transition diagram in Figure 3.2. Each activity in the model corresponds to a vector, called the *transition vector*. For example, $task_1$ corresponds to the tran-



	l^{task_1}	l^{task_2}	l^{reset}
$User_1$	-1	1	0
$User_2$	1	-1	0
$Provider_1$	-1	0	1
$Provider_2$	1	0	-1

Figure 3.2: *Transition vectors form an activity matrix*

sition vector $l^{task_1} = (-1, 1, -1, 1)^T$. That is, the derived state vector by firing $task_1$ from a state, can be represented by the sum of l^{task_1} and the state enabling $task_1$. For instance, $(2, 0, 2, 0)^T + l^{task_1} = (1, 1, 1, 1)^T$ illustrates that $\mathbf{s}_1 = (2, 0, 2, 0)^T$ transitions into $\mathbf{s}_2 = (1, 1, 1, 1)^T$ after enabling $task_1$. Similarly, $\mathbf{s}_3 = (1, 1, 2, 0)^T$ changing into $\mathbf{s}_5 = (0, 2, 1, 1)$ after firing $task_1$ can be manifested as $(1, 1, 2, 0)^T + l^{task_1} = (0, 2, 1, 1)^T$.

Similarly, $task_2$ corresponds to $l^{task_2} = (1, -1, 0, 0)^T$ while $reset$ corresponds to $l^{reset} = (0, 0, -1, 1)$. The three transition vectors form a matrix, called an *activity matrix*, see the table on the right side of Figure 3.2. Each activity in the model is represented by a transition vector — a column of the activity matrix, and each column expresses an activity. So the activity matrix is essentially indicating both an injection and a surjection from syntactic to numerical representation of the transition between system states. The concept of the activity matrix for PEPA was first proposed by Hillston in [CGH05, Hil05a]. However, the original definition cannot fully reflect the representation mapping considered here. This is due to the fact of that the definition is local-derivative-centric rather than transition centric.

In order to present the formal definition of activity matrix in [Hil05a], some terminology is first introduced. Consider a local derivative d_i. An activity l_j is an *exit activity* of d_i if d_i enables l_j. Similarly, an activity l_j is an *entry activity* of d_i if there is a local derivative which enables l_j such that d_i is the resulting derivative after firing l_j. The impact of activities on derivatives, easily derived from the syntactic presentation of the model, can be recorded in a matrix form, as defined below.

Definition 3.3.1. *(**Activity Matrix** [Hil05a]). For a model with $N_{\mathcal{A}}$ activities and $N_{\mathcal{D}}$ distinct local derivatives, the activity matrix M_a is an $N_{\mathcal{D}} \times N_{\mathcal{A}}$ matrix, and the entries are defined as*

follows.

$$(U_i, l_j) = \begin{cases} +1 & \textit{if } l_j \textit{ is an entry activity of } U_i, \\ -1 & \textit{if } l_j \textit{ is an exit activityof } U_i\ , \\ 0 & \textit{otherwise}. \end{cases}$$

This definition will lead to the matrix in Figure 3.2. An algorithm to automatically derive the activity matrix from a given PEPA model is given in [Hil05a]. However, the definition and algorithm are local-derivative-centric, which result in some limitations for more general applications. For example, for some PEPA models (e.g. Model 2 in next subsection), some columns of the defined matrix cannot be taken as transition vectors so that this definition cannot fully reflect the PEPA semantics in some circumstances. In the following subsection, a modified definition of the activity matrix is given. The new definition is activity- or transition-centric, which brings the benefit that each transition is represented by a column of the matrix and vice versa.

3.3.2 Labelled activity and modified activity matrix

It is very common that in a PEPA model, there may be a choice of derivatives after firing an activity. For example, in the following Model 2, firing α in the component P may lead to two possible local derivatives: P_2 and P_3, while firing β may lead to P_1 and P_3. In addition, firing γ may lead to P_1, Q_1. See Figure 3.3. However, only one derivative can be chosen after each firing of an activity, according to the semantics of PEPA. But the current definition of activity matrix cannot clearly reflect this point. See the activity matrix of Model 2 given in Table 3.2. In addition, the individual activity γ in this table, which can be enabled by both P_3 and Q_2, may be confused as a shared activity.

Model 2.

$$\begin{aligned} P_1 &\stackrel{\text{def}}{=} (\alpha, r'_\alpha).P_2 + (\alpha, r''_\alpha).P_3 \\ P_2 &\stackrel{\text{def}}{=} (\beta, r_\beta).P_1 + (\beta, r'_\beta).P_3 \\ P_3 &\stackrel{\text{def}}{=} (\gamma, r_\gamma).P_1 \\ Q_1 &\stackrel{\text{def}}{=} (\alpha, r_\alpha).Q_2 \\ Q_2 &\stackrel{\text{def}}{=} (\gamma, r'_\gamma).Q_1 \\ P_1[A] &\underset{\{\alpha\}}{\bowtie} Q_1[B]. \end{aligned}$$

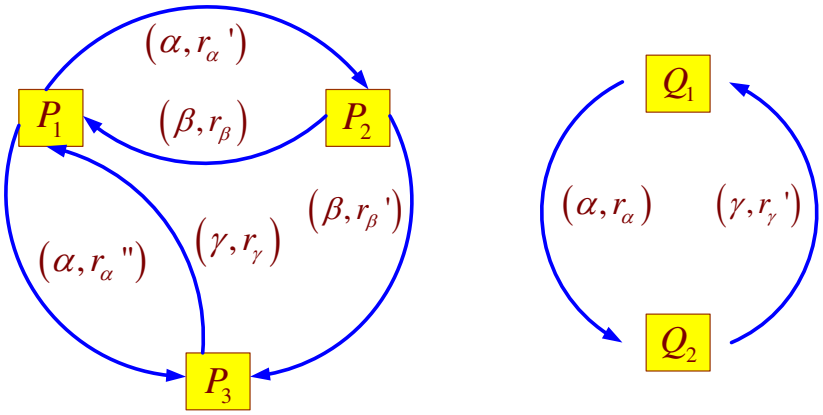


Figure 3.3: *Transition diagram of Model 2*

As a result, for Model 2 each column of its activity matrix cannot be considered as a transition vector, since for example, the transition to the next state vector from the starting state cannot be described using any column vector of the activity matrix.

	α	β	γ
P_1	-1	1	1
P_2	1	-1	0
P_3	1	1	-1
Q_1	-1	0	1
Q_2	1	0	-1

Table 3.2: *Originally defined activity matrix of Model 2*

In order to better reflect the semantics of PEPA, we modify the definition of the activity matrix in this way: if there are m possible outputs, namely $\{R_1, R_2, \cdots, R_m\}$, after firing either an individual or a shared activity l, then l is "split" into m labelled ls: $l^{w_1}, l^{w_2}, \cdots, l^{w_m}$. Here $\{w_i\}_{i=1}^m$ are m distinct labels, corresponding to $\{R_i\}_{i=1}^m$ respectively. Each l^{w_i} can only lead to a unique output R_i. Here there are no new activities created, since we just attach labels to the activity to distinguish the outputs of firing this activity. The modified activity matrix clearly reflects that only one, not two or more, result can be obtained from firing l. And thus, each l^{w_i} can represent a transition vector.

For example, see the modified activity matrix of Model 2 in Table 3.3. In this activity matrix, the individual activity γ has different "names" for different component types, so that it is not confused with a shared activity. Another activity β, is labelled as $\beta^{P_2 \to P_1}$ and $\beta^{P_2 \to P_3}$, to respectively reflect the corresponding two choices. In this table, the activity α is also split and

attached with labels.

l	$\alpha^{(P_1\to P_2,Q_1\to Q_2)}$	$\alpha^{(P_1\to P_3,Q_1\to Q_2)}$	$\beta^{P_2\to P_1}$	$\beta^{P_2\to P_3}$	$\gamma^{P_3\to P_1}$	$\gamma^{Q_2\to Q_1}$
P_1	-1	-1	1	0	1	0
P_2	1	0	-1	-1	0	0
P_3	0	1	0	1	-1	0
Q_1	-1	-1	0	0	0	1
Q_2	1	1	0	0	0	-1

Table 3.3: *Modified activity matrix of Model 2*

Before giving the modified definition of activity matrix for any general PEPA model, the pre and post sets for an activity are first defined. For convenience, throughout the thesis any transition $U \xrightarrow{(l,r)} V$ defined in the PEPA models may be rewritten as $U \xrightarrow{(l,r_l^{U\to V})} V$, or just $U \xrightarrow{l} V$ if the rate is not being considered, where U and V are two local derivatives.

Definition 3.3.2. ***(Pre and post local derivative)***

1. *If a local derivative U can enable an activity l, that is $U \xrightarrow{l} \cdot$, then U is called a* pre local derivative *of l. The set of all pre local derivatives of l is denoted by* $\mathrm{pre}(l)$*, called the* pre set *of l.*

2. *If V is a local derivative obtained by firing an activity l, i.e. $\cdot \xrightarrow{l} V$, then V is called a* post local derivative *of l. The set of all post local derivatives is denoted by* $\mathrm{post}(l)$*, called the* post set *of l.*

3. *The set of all the local derivatives derived from U by firing l, i.e.*

$$\mathrm{post}(U, l) = \{V \mid U \xrightarrow{l} V\},$$

is called the post set of l from U.

According to Definition 3.3.2, the pre and post sets of all activities in Model 2 are as listed below.

α:

$$\mathrm{pre}(\alpha) = \{P_1, Q_1\}, \qquad \mathrm{post}(\alpha) = \{P_2, P_3, Q_2\},$$
$$\mathrm{pre}(P_1, \alpha) = \{P_2, P_3\}, \quad \mathrm{pre}(Q_1, \alpha) = \{Q_2\}.$$

β:

$$\mathrm{pre}(\beta) = \{P_2\}, \quad \mathrm{post}(\beta) = \mathrm{post}(P_2, \beta) = \{P_1, P_3\}.$$

γ:

$$\begin{aligned} \mathrm{pre}(\gamma) &= \{P_3, Q_2\}, \quad \mathrm{post}(\gamma) = \{P_1, Q_1\}, \\ \mathrm{post}(P_3, \gamma) &= \{P_1\}, \quad \mathrm{post}(Q_2, \gamma) = \{Q_1\}. \end{aligned}$$

Obviously, if l has only one pre local derivative, i.e. $\#\mathrm{pre}(l) = 1$, then l is an individual activity, i.e. an activity that is not synchronised with other activities, like β in Model 2. But l being individual does not imply $\#\mathrm{pre}(l) = 1$, see γ for instance. If l is shared, then $\#\mathrm{pre}(l) > 1$, for example, see $\#\mathrm{pre}(\alpha) = \#\{P_1, Q_1\} = 2$. For a shared activity l with $\mathrm{pre}(l) = k$, there are k local derivatives that can enable this activity, each of them belonging to a distinct component type. The obtained local derivatives are in the set $\mathrm{post}(\mathrm{pre}(l)[i], l)$, where $\mathrm{pre}(l)[i]$ is the i-th pre local derivative of l. But only one of them can be chosen after l is fired from $\mathrm{pre}(l)[i]$. Since for the component type, namely i or $\mathcal{C}_i$, there are $\#\mathrm{post}(\mathrm{pre}(l)[i], l)$ outputs, so the total number of the distinct transitions for the whole system is

$$\prod_{i=1}^{k} \#\mathrm{post}(\mathrm{pre}(l)[i], l).$$

That is, there are $\prod_{i=1}^{k} \#\mathrm{post}(\mathrm{pre}(l)[i], l)$ possible results but only one of them can be chosen by the system after the shared activity l is fired. In other words, to distinguish these possible transitions, we need $\prod_{i=1}^{k} \#\mathrm{post}(\mathrm{pre}(l)[i], l)$ different labels. Here are the readily accessible labels:

$$(\mathrm{pre}(l)[1] \to V_1, \mathrm{pre}(l)[2] \to V_2, \cdots, \mathrm{pre}(l)[k] \to V_k),$$

where $V_i \in \mathrm{post}(\mathrm{pre}(l)[i], l)$. Obviously, for each vector

$$(V_1, V_2, \cdots, V_k) \in \mathrm{post}(\mathrm{pre}(l)[1], l) \times \mathrm{post}(\mathrm{pre}(l)[2], l) \times \cdots \times \mathrm{post}(\mathrm{pre}(l)[k], l),$$

the labelled activity $l^{(\mathrm{pre}(l)[1] \to V_1, \mathrm{pre}(l)[2] \to V_2, \cdots, \mathrm{pre}(l)[k] \to V_k)}$ represents a distinct transition. For example, α in Model 2 can be labelled as $\alpha^{(P_1 \to P_2, Q_1 \to Q_2)}$ and $\alpha^{(P_1 \to P_3, Q_1 \to Q_2)}$.

For an individual activity l, things are rather simple and easy: for $U \in \mathrm{pre}(l)$, l can be labelled as $l^{U \to \mathrm{post}(U,l)[1]}, l^{U \to \mathrm{post}(U,l)[2]}, l^{U \to \mathrm{post}(U,l)[k_U]}$, where $k_U = \#\mathrm{post}(U, l)$. Varying $U \in \mathrm{pre}(l)$, there are $\sum_{U \in \mathrm{pre}(l)} \#\mathrm{post}(U, l)$ labels needed to distinguish the possible transi-

tions. See $\beta^{P_2 \to P_1}, \beta^{P_2 \to P_3}, \gamma^{P_3 \to P_1}, \gamma^{Q_2 \to Q_1}$ in Model 2 for instance. Now we give the formal definition.

Definition 3.3.3. *(**Labelled Activity**).*

1. *For any individual activity l, for each $U \in \mathrm{pre}(l), V \in \mathrm{post}(U, l)$, label l as $l^{U \to V}$.*

2. *For a shared activity l, for each*

$$(V_1, V_2, \cdots, V_k) \in \mathrm{post}(\mathrm{pre}(l)[1], l) \times \mathrm{post}(\mathrm{pre}(l)[2], l) \times \cdots \times \mathrm{post}(\mathrm{pre}(l)[k], l),$$

label l as l^w, where

$$w = (\mathrm{pre}(l)[1] \to V_1, \mathrm{pre}(l)[2] \to V_2, \cdots, \mathrm{pre}(l)[k] \to V_k).$$

Each $l^{U \to V}$ or l^w is called a labelled activity. *The set of all labelled activities is denoted by $\mathcal{A}_{\mathrm{label}}$. For the above labelled activities $l^{U \to V}$ and l^w, their respective pre and post sets are defined as*

$$\mathrm{pre}(l^{U \to V}) = \{U\},\ \mathrm{post}(l^{U \to V}) = \{V\},$$

$$\mathrm{pre}(l^w) = \mathrm{pre}(l),\ \mathrm{post}(l^w) = \{V_1, V_2, \cdots, V_k\}.$$

According to Definition 3.3.3, each $l^{U \to V}$ or l^w can only lead to a unique output. No new activities are created, since labels are only attached to the activity to distinguish the results after this activity is fired.

The impact of labelled activities on local derivatives can be recorded in a matrix form, as defined below.

Definition 3.3.4. *(**Activity Matrix**). For a model with $N_{\mathcal{A}_{\mathrm{label}}}$ labelled activities and $N_{\mathcal{D}}$ distinct local derivatives, the activity matrix $\mathbf{C}$ is an $N_{\mathcal{D}} \times N_{\mathcal{A}_{\mathrm{label}}}$ matrix, and the entries are defined as follows*

$$\mathbf{C}(U_i, l_j) = \begin{cases} +1 & \text{if } U_i \in \mathrm{post}(l_j) \\ -1 & \text{if } U_i \in \mathrm{pre}(l_j) \\ 0 & \text{otherwise} \end{cases}$$

where l_j is a labelled activity.

The modified activity matrix captures all the structural information, including the information about choices and synchronisations, of a given PEPA model. From each row of the matrix, which corresponds to each local derivative, we can know which activities this local derivative can enable and after which activities are fired this local derivative can be derived. From the perspective of the columns, the number of "-1"s in a column tells whether the corresponding activity is synchronised or not. Only one "-1" means that this transition corresponds to an individual activity. The locations of "-1" and "1" indicate which local derivatives can enable the activity and what the derived local derivatives are, i.e. the pre and post local derivatives. In addition, the numbers of "-1"s and "1"s in each column are the same, because any transition in any component type corresponds to a unique pair of pre and post local derivatives. In fact, all this information is also stored in the labels of the activities. Therefore, with the transition rate functions defined in the next section to capture the timing information, a given PEPA model can be recovered from its activity matrix.

Hereafter the terminology of *activity matrix* refers to the one in Definition 3.3.4. This definition embodies the transition or operation rule of a given PEPA model, with the exception of timing information. For a given PEPA model, each transition of the system results from the firing of an activity. Each optional result after enabling this activity corresponds to a relevant labelled activity, that is, corresponds to a column of the activity matrix. Conversely, each column of the activity matrix corresponding to a labelled activity, represents an activity and the chosen derived result after this activity is fired. So each column corresponds to a system transition. Therefore, we have the following proposition, which specifies the correspondence between system transitions and the columns of the activity matrix.

Proposition 3.3.1. *Each column of the activity matrix corresponds to a system transition and each transition can be represented by a column of the activity matrix.*

As Proposition 3.3.1 reveals, the syntactically defined activity matrix provides the numerical representation for the transitions between system states. Moreover, it is convenient for utilising mathematical techniques such as linear algebra and linear programming to investigate the structural properties of PEPA models (see the next chapter for details).

An algorithm for automatically deriving the activity matrix from any PEPA model will be given in the next section. For convenience, in the following we define the pre and post activity matrices for PEPA, which can be directly obtained from the activity matrix.

Definition 3.3.5. *(**Pre Activity Matrix**, **Post Activity Matrix**) Let* $\mathbf{C}$ *be the activity matrix of a PEPA model. The* pre activity matrix $\mathbf{C^{Pre}}$ *and* post activity matrix $\mathbf{C^{Post}}$ *of the model are defined as follows:*

$$\mathbf{C^{Pre}}(U_i, l_j) = \begin{cases} +1 & \mathbf{C}(U_i, l_j) = -1 \\ 0 & \textit{otherwise}. \end{cases},$$

$$\mathbf{C^{Post}}(U_i, l_j) = \begin{cases} +1 & \mathbf{C}(U_i, l_j) = +1 \\ 0 & \textit{otherwise}. \end{cases}$$

Clearly, the pre activity matrix indicates the pre local derivatives for each labelled activity, i.e. the local derivatives which can fire this activity. The post activity matrix indicates the post local derivatives, i.e. the derived local derivatives after firing an activity. The activity matrix equals the difference between the pre and post activity matrices, i.e. $\mathbf{C} = \mathbf{C^{Pre}} - \mathbf{C^{Post}}$.

3.4 Transition Rate Function

As illustrated in previous sections, the system states and the transitions between them can be described using numerical vector forms. The structural information of any general PEPA model is captured in the activity matrix, which is constituted by all transition vectors. However, the duration of each transition has not yet been specified. This section defines transition rate functions for transition vectors or labelled activities to capture the timing information of PEPA models.

3.4.1 Model 2 continued

Let us start from Model 2 again:

$$P_1 \stackrel{def}{=} (\alpha, r'_\alpha).P_2 + (\alpha, r''_\alpha).P_3$$
$$P_2 \stackrel{def}{=} (\beta, r_\beta).P_1 + (\beta, r'_\beta).P_3$$
$$P_3 \stackrel{def}{=} (\gamma, r_\gamma).P_1$$
$$Q_1 \stackrel{def}{=} (\alpha, r_\alpha).Q_2$$
$$Q_2 \stackrel{def}{=} (\gamma, r'_\gamma).Q_1$$
$$P_1[A] \underset{\{\alpha\}}{\bowtie} Q_1[B].$$

As Table 3.3 shows, activity γ in Model 2 is labelled as $\gamma^{P_3 \to P_1}$ and $\gamma^{Q_2 \to Q_1}$. For $\gamma^{P_3 \to P_1}$, there are $\mathbf{x}[P_3]$ instances of the component type P in the local derivative P_3 in state $\mathbf{x}$, each enabling the individual activity concurrently with the rate r_γ. So the rate of $\gamma^{P_3 \to P_1}$ in state $\mathbf{x}$ is $f(\mathbf{x}, \gamma^{P_3 \to P_1}) = r_\gamma \mathbf{x}[P_3]$. Similarly, the rate for $\gamma^{Q_2 \to Q_1}$ in state $\mathbf{x}$ is $r_{\gamma'} \mathbf{x}[Q_2]$. This is consistent with the definition of apparent rate in PEPA, that states that if there are N replicated instances of a component enabling a transition (l, r), the apparent rate of the activity will be $r \times N$.

In Model 2 activity β is labelled as $\beta^{P_2 \to P_1}$ and $\beta^{P_2 \to P_3}$, to respectively reflect the corresponding two choices. According to the model definition, there is a flux of $r_\beta \mathbf{x}(P_2)$ into P_1 from P_2 after firing β in state $\mathbf{x}$. So the transition rate function is defined as $f(\mathbf{x}, \beta^{P_2 \to P_1}) = r_\beta \mathbf{x}[P_2]$. Similarly, we can define $f(\mathbf{x}, \beta^{P_2 \to P_3}) = r'_\beta \mathbf{x}[P_2]$. These rate functions can be defined or interpreted in an alternative way. In state $\mathbf{x}$, there are $\mathbf{x}[P_2]$ instances that can fire β. So the apparent rate of β is $(r_\beta + r'_\beta)\mathbf{x}[P_2]$. By the semantics of PEPA, the probabilities of choosing the outputs are $\frac{r_\beta}{r_\beta + r'_\beta}$ and $\frac{r'_\beta}{r_\beta + r'_\beta}$ respectively. So the rate of the transition $\beta^{P_2 \to P_1}$ is

$$f(\mathbf{x}, \beta^{P_2 \to P_1}) = \frac{r_\beta}{r_\beta + r'_\beta}(r_\beta + r'_\beta)\mathbf{x}[P_2] = r_\beta \mathbf{x}[P_2], \tag{3.3}$$

while the rate of the transition $\beta^{P_2 \to P_3}$ is

$$f(\mathbf{x}, \beta^{P_2 \to P_3}) = \frac{r'_\beta}{r_\beta + r'_\beta}(r_\beta + r'_\beta)\mathbf{x}[P_2] = r'_\beta \mathbf{x}[P_2]. \tag{3.4}$$

In Model 2, α is a shared activity with three local rates: r_α, r'_α and r''_α. The apparent rate of α in P_1 is $(r'_\alpha + r''_\alpha)\mathbf{x}[P_1]$, while in Q_1 it is $r_\alpha \mathbf{x}[Q_1]$. According to the PEPA semantics, the apparent rate of a synchronised activity is the minimum of the apparent rates of the cooperating components. So the apparent rate of α as a synchronisation activity is $\min\{(r'_\alpha + r''_\alpha)\mathbf{x}[P_1], r_\alpha \mathbf{x}[Q_1]\}$. After firing α, P_1 becomes either P_2 or P_3, with the probabilities $\frac{r'_\alpha}{r'_\alpha + r''_\alpha}$ and $\frac{r''_\alpha}{r'_\alpha + r''_\alpha}$ respectively. Simultaneously, Q_1 becomes Q_2 with the probability 1. So the rate function of transition $(P_1 \to P_2, Q_1 \to Q_2)$, represented by $f(\mathbf{x}, \alpha^{(P_1 \to P_2, Q_1 \to Q_2)})$, is

$$f(\mathbf{x}, \alpha^{(P_1 \to P_2, Q_1 \to Q_2)}) = \frac{r'_\alpha}{r'_\alpha + r''_\alpha} \min\{(r'_\alpha + r''_\alpha)\mathbf{x}[P_1], r_\alpha \mathbf{x}[Q_1]\}. \tag{3.5}$$

Similarly,

$$f(\mathbf{x}, \alpha^{(P_1 \to P_3, Q_1 \to Q_2)}) = \frac{r''_\alpha}{r'_\alpha + r''_\alpha} \min\{(r'_\alpha + r''_\alpha)\mathbf{x}[P_1], r_\alpha \mathbf{x}[Q_1]\}. \tag{3.6}$$

The above discussion about the simple example should help the reader to understand the definition of transition rate function for general PEPA models, which is presented in the next subsection.

3.4.2 Definitions of transition rate function

In a PEPA model, for convenience, we may rewrite any $U \xrightarrow{(l,r)} V$ as $U \xrightarrow{(l,r_l^{U\to V})} V$, where r is denoted by $r_l^{U\to V}$. The transition rate functions of general PEPA models are defined below. We first give the definition of the apparent rate of an activity in a local derivative.

Definition 3.4.1. *(**Apparent Rate of** l **in** U) Suppose l is an activity of a PEPA model and U is a local derivative enabling l (i.e. $U \in \mathrm{pre}(l)$). Let $\mathrm{post}(U, l)$ be the set of all the local derivatives derived from U by firing l, i.e. $\mathrm{post}(U, l) = \{V \mid U \xrightarrow{(l,r_l^{U\to V})} V\}$. Let*

$$r_l(U) = \sum_{V \in \mathrm{post}(U,l)} r_l^{U\to V}. \tag{3.7}$$

The apparent rate *of l in U in state $\mathbf{x}$, denoted by $r_l(\mathbf{x}, U)$, is defined as*

$$r_l(\mathbf{x}, U) = \mathbf{x}[U] r_l(U). \tag{3.8}$$

The above definition is used to define the following transition rate function.

Definition 3.4.2. *(**Transition Rate Function**) Suppose l is an activity of a PEPA model and $\mathbf{x}$ denotes a state vector.*

1. *If l is individual, then for each $U \xrightarrow{(l,r^{U\to V})} V$, the transition rate function of labelled activity $l^{U\to V}$ in state $\mathbf{x}$ is defined as*

$$f(\mathbf{x}, l^{U\to V}) = \mathbf{x}[U] r_l^{U\to V}. \tag{3.9}$$

2. *If l is synchronised, with* $\mathrm{pre}(l) = \{U_1, U_2, \cdots, U_k\}$*, then for each*

$$(V_1, V_2, \cdots, V_k) \in \mathrm{post}(U_1, l) \times \mathrm{post}(U_2, l) \times \cdots \times \mathrm{post}(U_k, l),$$

let $w = (U_1 \to V_1, U_2 \to V_2, \cdots, U_k \to V_k)$*. Then the transition rate function of labelled activity* l^w *in state* $\mathbf{x}$ *is defined as*

$$f(\mathbf{x}, l^w) = \left(\prod_{i=1}^{k} \frac{r_l^{U_i \to V_i}}{r_l(U_i)} \right) \min_{i \in \{1, \cdots, k\}} \{r_l(\mathbf{x}, U_i)\},$$

where $r_l(\mathbf{x}, U_i) = \mathbf{x}[U_i] r_l(U_i)$ *is the apparent rate of* l *in* U_i *in state* $\mathbf{x}$*. So*

$$f(\mathbf{x}, l^w) = \left(\prod_{i=1}^{k} \frac{r_l^{U_i \to V_i}}{r_l(U_i)} \right) \min_{i \in \{1, \cdots, k\}} \{\mathbf{x}[U_i] r_l(U_i)\}. \tag{3.10}$$

Remark 3.4.1. *Definition 3.4.2 accommodates the passive or unspecified rate* $\top$*. If there are some* $r_l^{U \to V}$ *which are* $\top$*, then the relevant calculation in the rate functions (3.9) and (3.10) can be made according to the following inequalities and equations that define the comparison and manipulation of unspecified activity rates (see Section 2.3.3 in Chapter 2):*

$$\begin{array}{rl} r < w\top & \text{for all } r \in \mathbb{R}^+ \text{ and for all } w \in \mathbb{N} \\ w_1\top < w_2\top & \text{if } w_1 < w_2 \text{ for all } w_1, w_2 \in \mathbb{N} \\ w_1\top + w_2\top = (w_1 + w_2)\top & \text{for all } w_1, w_2 \in \mathbb{N} \\ \frac{w_1\top}{w_2\top} = \frac{w_1}{w_2} & \text{for all } w_1, w_2 \in \mathbb{N} \end{array}$$

Moreover, we assume that $0 \cdot \top = 0$*. So the terms such as "*$\min\{A\top, rB\}$*" are interpreted as [BGH07]:*

$$\min\{A\top, rB\} = \begin{cases} rB, & A > 0, \\ 0, & A = 0. \end{cases}$$

The definition of the transition rate function is consistent with the semantics of PEPA. We state this result in a proposition.

Proposition 3.4.1. *The transition rate function in Definition 3.4.2 is consistent with the operational semantics of PEPA.*

The proof is given in Appendix B.1. Moreover, this kind of transition rate function has the following properties.

Proposition 3.4.2. *The transition rate function is nonnegative. If U is a pre local derivative of l, i.e. $U \in \mathrm{pre}(l)$, then the transition rate function of l in a state $\mathbf{x}$ is less than the apparent rate of l in U in this state, that is*

$$0 \leq f(\mathbf{x}, l) \leq r_l(\mathbf{x}, U) = \mathbf{x}[U] r_l(U),$$

where $r_l(U)$ is the apparent rate of l in U for a single instance of U.

The proof is trivial and omitted.

Proposition 3.4.3. *Let l be an labelled activity, and $\mathbf{x}, \mathbf{y}$ be two states. The transition rate function $f(\mathbf{x}, l)$ defined in Definition 3.4.2 satisfies:*

1. *For any $H > 0$, $Hf(\mathbf{x}/H, l) = f(\mathbf{x}, l)$.*
2. *There exists $M > 0$ such that $|f(\mathbf{x}, l) - f(\mathbf{y}, l)| \leq M\|\mathbf{x} - \mathbf{y}\|$ for any $\mathbf{x}, \mathbf{y}$ and l.*

Hereafter $\|\cdot\|$ denotes any matrix norm since all finite matrix norms are equivalent. For example, we may define $\|A\| = \sqrt{\sum_{i,j} a_{ij}^2}$ for a matrix $A = (a_{ij})$. The first term of this proposition illustrates a homogenous property of the rate function, while the second indicates the Lipschtiz continuous property, both with respect to states. These characteristics will be utilised to investigate the fluid approximations of PEPA models in the following chapters. The proof of Proposition 3.4.3 is given in Appendix B.2.

3.4.3 Algorithm for deriving activity matrix and transition rate functions

This section presents an algorithm for automatically deriving the activity matrix and rate functions from any PEPA model, see Algorithm 1 (on page 66).

The lines 3-12 of Algorithm 1 deal with individual activities while lines $13-32$ deal with shared activities. The calculation methods in this algorithm are the embodiment of the definitions of labelled activity and apparent rate as well as transition rate function. We would like to use an example to illustrate this algorithm. Recall the discussion in Section 3.3.2 and 3.4.1. The shared activity α in Model 2 is labelled as $\alpha^{(P_1 \to P_2, Q_1 \to Q_2)}$ and $\alpha^{(P_1 \to P_3, Q_1 \to Q_2)}$ with the following corresponding transition rate functions respectively:

$$f(\mathbf{x}, \alpha^{(P_1 \to P_2, Q_1 \to Q_2)}) = \frac{r'_\alpha}{r'_\alpha + r''_\alpha} \min\{(r'_\alpha + r''_\alpha)\mathbf{x}[P_1], r_\alpha \mathbf{x}[Q_1]\}, \tag{3.11}$$

Algorithm 1 Derive activity matrix and transition rate functions from a general PEPA model

$\mathcal{A}_{\text{label}} = \emptyset$; $\mathcal{D}$ is the set of all local derivatives
for all activity $l \in \mathcal{A}$ **do**
  if l is an independent activity **then**
    for all local derivatives $U, V \in \mathcal{D}$ **do**
      if $U \xrightarrow{(l,r)} V$ **then**
        $\mathcal{A}_{\text{label}} = \mathcal{A}_{\text{label}} \cup \{l^{U \to V}\}$ // Label l as $l^{U \to V}$
        // Form a corresponding column of the activity matrix and the rate function
        $$M_a(d, l^{U \to V}) = \begin{cases} -1, & d = U \\ 1, & d = V \\ 0, & otherwise \end{cases}$$
        $f(\mathbf{x}, l^{U \to V}) = r\mathbf{x}[U]$
      end if
    end for
  end if
  if l is a synchronised activity **then**
    $\text{pre}(l) = \emptyset, \text{post}(U, l) = \emptyset, \ \forall U \in \mathcal{D}$
    for all local derivatives $U, V \in \mathcal{D}$ **do**
      if $U \xrightarrow{(l,r)} V$ **then**
        $\text{pre}(l) = \text{pre}(l) \cup \{U\}$
        $\text{post}(U, l) = \text{post}(U, l) \cup \{V\}$
        $r_l^{U \to V} = r$
      end if
    end for
    Denote $\text{pre}(l) = \{\text{pre}(l)[1], \text{pre}(l)[2], \cdots, \text{pre}(l)[k]\}$, where $k = \#\text{pre}(l)$
    for $i = 1 \ldots k$ **do**
      $$r_l(\text{pre}(l)[i]) = \sum_{V \in \text{post}(\text{pre}(l)[i], l)} r_l^{\text{pre}(l)[i] \to V}$$
    end for
    $K(l) = \text{post}(\text{pre}(l)[1], l) \times \text{post}(\text{pre}(l)[2], l) \times \cdots \times \text{post}(\text{pre}(l)[k], l)$
    for all $(V_1, V_2, \cdots, V_k) \in K(l)$ **do**
      $w = (\text{pre}(l)[1] \to V_1, \text{pre}(l)[2] \to V_2, \cdots, \text{pre}(l)[k] \to V_k)$
      $\mathcal{A}_{\text{label}} = \mathcal{A}_{\text{label}} \cup \{l^w\}$ // Label l as l^w
      // Form a column of M_a and the rate function corresponding to l^w

$$M_a(d, l^w) = \begin{cases} -1, & d \in \text{pre}(l) \\ 1, & d \in \{V_1, V_2, \cdots, V_k\} \\ 0, & otherwise \end{cases}$$

$$f(\mathbf{x}, l^w) = \left(\prod_{i=1}^{k} \frac{r_l^{\text{pre}(l)[i] \to V_i}}{r_l(\text{pre}(l)[i])} \right) \min_{i \in \{1, \cdots, k\}} \{r_l(\text{pre}(l)[i])\mathbf{x}[\text{pre}(l)[i]]\}$$

    end for
  end if
end for
Output $\mathcal{A}_{\text{label}}$; M_a; $f(\mathbf{x}, l)$ $(\forall l \in \mathcal{A}_{\text{label}})$.

$$f(\mathbf{x}, \alpha^{(P_1 \to P_3, Q_1 \to Q_2)}) = \frac{r''_\alpha}{r'_\alpha + r''_\alpha} \min\{(r'_\alpha + r''_\alpha)\mathbf{x}[P_1], r_\alpha \mathbf{x}[Q_1]\}. \tag{3.12}$$

Now we show in detail how Algorithm 1 derives the labelled α, the corresponding columns of the activity matrix, and the rate functions (3.11) and (3.12). Since α is a shared activity, let us begin at Line 13 of the algorithm.

— First (by Lines $13-21$ of Algorithm 1),

$$\text{pre}(\alpha) = \{P_1, Q_1\},$$

$$\text{post}(P_1, \alpha) = \{P_2, P_3\}, \quad \text{post}(Q_1, \alpha) = \{Q_2\},$$

and

$$r_\alpha^{P_1 \to P_2} = r'_\alpha, \quad r_\alpha^{P_1 \to P_3} = r''_\alpha, \quad r_\alpha^{Q_1 \to Q_2} = r_\alpha.$$

— Then (by Lines $22-26$ of Algorithm 1),

$$r_\alpha(P_1) = r_\alpha^{P_1 \to P_2} + r_\alpha^{P_1 \to P_3} = r'_\alpha + r''_\alpha,$$

$$r_\alpha(Q_1) = r_\alpha^{Q_1 \to Q_2} = r_\alpha,$$

and

$$K(\alpha) = \text{post}(P_1, \alpha) \times \text{post}(Q_1, \alpha) = \{P_2, P_3\} \times \{Q_2\}.$$

— Finally (by Lines $27-33$ of Algorithm 1), since there are two labels for α: w_1 and w_2, where $w_1 = (P_1 \to P_2, Q_1 \to Q_2)$ and $w_2 = (P_1 \to P_3, Q_1 \to Q_2)$. So

$$M_a(\cdot, \alpha^{w_1}) = (-1, 1, 0, -1, 1)^T, \; M_a(\cdot, \alpha^{w_1}) = (-1, 0, 1, -1, 1)^T.$$

$$\begin{aligned} f(\mathbf{x}, \alpha^{w_1}) &= \frac{r_\alpha^{P_1 \to P_2}}{r_\alpha(P_1)} \frac{r_\alpha^{Q_1 \to Q_2}}{r_\alpha(Q_1)} \min\{r_\alpha(P_1)\mathbf{x}[P_1], r_\alpha(Q_1)\mathbf{x}[Q_1]\} \\ &= \frac{r'_\alpha}{r'_\alpha + r''_\alpha} \min\{(r'_\alpha + r''_\alpha)\mathbf{x}[P_1], r_\alpha \mathbf{x}[Q_1]\}, \\ f(\mathbf{x}, \alpha^{w_2}) &= \frac{r_\alpha^{P_1 \to P_3}}{r_\alpha(P_1)} \frac{r_\alpha^{Q_1 \to Q_2}}{r_\alpha(Q_1)} \min\{r_\alpha(P_1)\mathbf{x}[P_1], r_\alpha(Q_1)\mathbf{x}[Q_1]\} \\ &= \frac{r''_\alpha}{r'_\alpha + r''_\alpha} \min\{(r'_\alpha + r''_\alpha)\mathbf{x}[P_1], r_\alpha \mathbf{x}[Q_1]\}. \end{aligned}$$

3.5 Associated Methods for Qualitative and Quantitative Analysis of PEPA Models

The activity matrices and transition rate functions defined in the previous subsections capture the structural and timing information of PEPA models respectively. When this information is available numerically, some efficient mathematical methods, including linear programming and ordinary differential equations can be directly utilised to help to overcome the problem of state-space explosion encountered in both qualitative and quantitative analysis for PEPA. This section briefly introduces these new approaches as well as technical foundations for employing them in the context of PEPA.

3.5.1 Numerical and aggregated representation for PEPA

As we know, for quantitative performance analysis, model simplification techniques such as fluid approximation can bring acceptable accuracy at a very low computational complexity. As for qualitative analysis, e.g. deadlock checking, mathematical tools including linear algebra and linear programming have been proven very powerful. Basically, the application of these tools and techniques needs appropriate mathematical models. This motivates and stimulates the new representation—numerical rather than the original syntactical representation—for PEPA models, for the purpose of directly exploiting these new methods. The numerical state vector, transition vector, and transition rate function previously defined in this chapter, have provided a fundamental numerical platform for the utilisation of the new approaches.

Syntactic and Separated Representation	Numerical and Aggregated Representation
number of instances of components in local derivatives (system equation)	state vector
action type; pre and post local derivative; synchronisation	labelled activity
operational semantics	activity matrix, transition rate function
action rate; apparent rate	transition rate function

Table 3.4: *From syntactical and separated to numerical and aggregated representation for PEPA*

The comparison between these two representation is given in Table 3.4. The justification of the equivalence between them, i.e. the consistency of the new definitions, has been shown in Proposition 3.4.1. The word "Aggregated" in this table reflects two things: there is no distinc-

tion made between different instances of the same component type; the system states, and the transitions between them represented by transition vectors, are considered holistically rather than locally based on sequential components. "Numerical" in this table emphasises that the system states, the transitions between the states, and the average duration of the transitions, are represented numerically. Again, the main benefits brought by the new representation scheme are the significant decrease of the size of the state space and the convenience for employing new mathematical methods.

3.5.2 Place/Transition system

Overcoming the state-space explosion problem is the basic motivation and stimulation for developing new methods for PEPA. The state space of a model is the foundation for both qualitative and quantitative analysis. Typical qualitative problems which can be addressed based on state-space related analysis include state space derivation and storage, deadlock checking, etc. Generally, qualitative analysis is structure-related rather than timing-related. Since the structural information has been numerically represented in activity matrices, it is possible and feasible to do qualitative analysis such as deadlock-checking based on activity matrices directly, and thus avoid the state-space explosion problem.

The numerical representation for system states and transitions, helps to find and manifest the P/T structure underlying each PEPA model. Thus some powerful techniques and theories such as linear algebra and linear programming developed for P/T systems [STC96] can be directly utilised for the qualitative analysis of PEPA. In the next chapter, the readers will see that through this approach, the derivation and storage of the state space of a class PEPA models will no longer be a problem since the state space can be expressed using linear algebraic equations. Moreover, structure-based rather than state-space-based theories and algorithms for deadlock checking have been developed based on these equations. They are particularly efficient for large scale systems with repeated components since they can avoid searching for deadlocks in the entire state space. Further, a kind of interesting and useful structural property —invariance—has been found in many PEPA models, which can be used to reason about the system. Of course, the foundation for the applications of these new methods is the numerical and aggregated representation for PEPA, which is already presented in this chapter. The detailed investigation and discussion of the qualitative analysis of PEPA models will be given in the next chapter.

3.5.3 Aggregated CTMC and ODEs

As we have mentioned in Chapter 2, for each PEPA model, there is a CTMC underlying the model. By solving the global balance equations associated with the infinitesimal generator of this CTMC, the steady-state probability distribution can be obtained, from which performance measures can be derived. According to the original definition of the PEPA language in which each instance of the same component type is considered distinctly, the size of the state space of this CTMC (called the *original CTMC*) may increase exponentially with the number of components. Since there is no difference between components of the same type, the number rather than the identity of the components in the local derivatives can be captured, introducing the concept of numerical vector form to represent the system state, which results in the aggregated CTMC. The size of the state space can thus be significantly reduced, as Proposition 3.2.1 shows, together with the computational complexity of deriving the performance by solving the corresponding global balance equations since, the dimension of the infinitesimal generator matrix is the square of the size of the state space.

Alternatively, the aggregated CTMC can be achieved by constructing a partition over the state space of the original CTMC. In the aggregated CTMC, each partition of states in the original CTMC forms a state, which can be represented by the numerical vector form as defined in Definition 3.2.1. This partition is induced by an equivalence relation defined over the state space of the original CTMC: the state $\mathbf{s}$ and $\mathbf{s}'$ are equivalent if and only if the numbers of the components in each local derivative at $\mathbf{s}$ and $\mathbf{s}'$ are the same. The infinitesimal generator and steady state distributions of the aggregated CTMC can also be formed from the ones of the original CTMC. For detailed information of aggregation of Markov processes, please refer to Section 5.4 in [Hil96].

Unless otherwise stated, hereafter the CTMC underlying a PEPA model refers to the aggregated CTMC, and the state of a model or a system is considered in the sense of aggregation. As discussed previously, a transition between states, namely from $\mathbf{x}$ to $\mathbf{x} + l$, is represented by a transition vector l corresponding to the labelled activity, namely l. The rate of the transition l in state $\mathbf{x}$ is specified by the transition rate function $f(\mathbf{x}, l)$. That is,

$$\mathbf{x} \xrightarrow{(l, f(\mathbf{x}, l))} \mathbf{x} + l.$$

Given a starting state $\mathbf{x}_0$, the transition chain corresponding to a firing sequence $l_0, l_1, \cdots, l, \cdots$

is

$$\mathbf{x}_0 \overset{(l_0, f(\mathbf{x}_0, l_0))}{\longrightarrow} \mathbf{x}_0 + l_0 \overset{(l_1, f(\mathbf{x}_0 + l_0, l_1))}{\longrightarrow} (\mathbf{x}_0 + l_0) + l_1 \overset{\cdots}{\longrightarrow} \cdots \overset{\cdots}{\longrightarrow} \mathbf{x} \overset{(l, f(\mathbf{x}, l))}{\longrightarrow} \mathbf{x} + l \overset{\cdots}{\longrightarrow} \cdots .$$

The above sequence can be considered as one path or realisation of a simulation of the aggregated CTMC, if the enabled activity at each state is chosen stochastically, i.e. is chosen through the approach of sampling. After a long time, the steady-state of the system is assumed to be achieved. The averaged occurrence number of an activity during one unit time, and the averaged proportion of the number of components appearing in a local derivative can be calculated from the simulation, which are referred to as the *throughput* of this activity and the *utilisation* of the local state respectively. Moreover, given a starting state and a stopping state, the duration between these two states, which is called *response time*, can be also obtained. In Chapter 7, we will provide a stochastic simulation algorithm (Algorithm 3), which is based on our numerical representation scheme, to derive these performance measures from PEPA models. The weakness of the simulation method is its high computational cost, which makes it not suitable for real-time performance monitoring or prediction. See Chapter 7 for a case study and detailed discussions.

A promising approach for quantitative analysis of PEPA is fluid approximation, which brings acceptable accuracy at a very low computational complexity for some models. The state space of an underlying CTMC is inherently discrete, with the entries within the numerical vector form always being non-negative integers and always being incremented or decremented in steps of one. As pointed out in [Hil05a], when the numbers of components are large these steps are relatively small and we can approximate the behaviour by considering the movement between states to be continuous, rather than occurring in discontinuous jumps. This approach results in a set of ODEs:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \sum_{l \in \mathcal{A}_{\mathrm{label}}} l f(\mathbf{x}, l), \tag{3.13}$$

where the vector $\mathbf{x}$ is short for $\mathbf{x}(t)$, representing the populations of components in local derivatives at time t; and l is a transition vector while $f(\mathbf{x}, l)$ is a rate function.

These ODEs are immediately available as long as the activity matrix and the transition rate functions are generated by Algorithm 1. The fluid approximation approach was first proposed by Hillston in [Hil96] for a class of restricted PEPA models. The restrictions include: individual activities must occur only once within derivative definitions and cannot appear within

different component definitions; shared activities cannot have different local rates, etc.. These restrictions are relaxed here, since our derived ODEs (see (3.13)) only depend on the activity matrix and transition rate functions, which are defined for general PEPA models. At this moment, there are some natural problems: does the ODE solution exist and is it unique? what is the relationship between that solution and the CTMC? how to derive performance measures from ODEs, etc. In the following chapters these problems will be discussed in detail.

3.6 Summary

This chapter has defined the labelled activities, activity matrices and transition rate functions for PEPA, to capture the structural and timing information respectively. These definitions are used to describe PEPA models numerically rather than syntactically. This numerical representation scheme provides a platform for the direct application of some powerful mathematical tools such as linear algebra, linear programming and ODEs, and non-mathematical methods such as P/T theory, for the purpose of qualitative and quantitative analysis of large scale PEPA models. Some fundamental properties of these definitions have been discussed and will be utilised for further investigation of PEPA in the following chapters.

Chapter 4
Structural Analysis for PEPA Models

4.1 Introduction

Structural analysis provides an important route to gaining insight about how systems will perform qualitatively. However, the size and complexity of such systems challenge the capabilities of the current approaches for structural analysis. For example, the current method to check whether there is a deadlock in a PEPA model relies on exploring the entire state space of the model. Therefore the computational complexity is mainly determined by the size of the state space. For large scale PEPA models, particularly the models with more than ten million states, even the derivation of the state space becomes impossible due to the state-space explosion problem, let alone deadlock-checking for these models. This chapter will demonstrate a new approach for structural analysis of PEPA, which avoids the state-space explosion problem.

The previous chapter has presented the definitions of numerical vectors and labelled activity, activity matrix, as well as transition rate functions, for the PEPA language. These definitions will closely relate PEPA to other formalisms such as Petri nets. Petri nets or P/T nets are another modelling language which is widely used in the analysis of systems that exhibit complex behaviour due to the interleaving of parallelism and synchronisation. In this chapter we will show there is a P/T structure underlying each PEPA model. Based on the techniques developed in the context of P/T systems in Petri nets, we will demonstrate how to find invariants and how to efficiently derive and store the state space for large scale PEPA models. In particular, we will present a structure-based deadlock-checking approach for PEPA which avoids the state-space explosion problem.

The PEPA models considered in this chapter satisfy two assumptions: there is no cooperation within groups of components of the same type; and each column of the activity matrix of a model is distinct, i.e. each labelled activity is distinct in terms of pre and post local derivatives. The remainder of this chapter is organised as follows: Section 2 presents the P/T structure underlying PEPA models; Section 3 discusses how to derive invariants from PEPA models;

Linearisation of the state space for PEPA is given in Section 4, based on which a new deadlock-checking method will be provided in Section 5. Finally, Section 6 concludes the chapter.

4.2 Place/Transtion Structure underlying PEPA Models

In Chapter 3 we have defined the numerical state vector and the activity matrix for the PEPA language. With the exception of time information, a PEPA model can be recovered from the activity matrix since it captures all the structural information of the system. These definitions and representations lead to a P/T structure underlying any PEPA model.

4.2.1 Dynamics of PEPA models

As we have mentioned, the numerical vectors indicate the system states while the activity matrix embodies the rules of system operation. Regardless the activity periods, we can use the numerical vector and activity matrix to describe the operational behaviour of PEPA models. In particular, the non-timing dynamics of the model can be described using the activity matrix. For example, recall Model 1 in Chapter 3:

$$
\begin{aligned}
User_1 &\stackrel{def}{=} (task_1, a).User_2 \\
User_2 &\stackrel{def}{=} (task_2, b).User_1 \\
Provider_1 &\stackrel{def}{=} (task_1, a).Provider_2 \\
Provider_2 &\stackrel{def}{=} (reset, d).Provider_1 \\
User_1[M] &\underset{\{task_1\}}{\bowtie} Provider_1[N].
\end{aligned}
$$

The activity matrix $\mathbf{C}$ and pre activity matrix $\mathbf{C^{Pre}}$ of Model 1 are demonstrated in Table 4.1. For convenience, here the labelled activities are just denoted by l^{task_1}, l^{task_2} and l^{reset} respectively. Assume there are two users and providers in the system, i.e. $M = N = 2$. Therefore, the starting state is $\mathbf{m}_0 = (2, 0, 2, 0)^T$. The diagram of the transitions between the states is presented in Figure 4.1.

According to the semantics of PEPA, in the starting state $\mathbf{m}_0$ only $task_1$ can be fired. The requirement for enabling $task_1$ in a state $\mathbf{m}$ is that there are at least one instance of $User_1$ and one instance of $Provider_1$ in this state. The mathematical expression of this statement is

(a) Activity Matrix $\mathbf{C}$

	l^{task_1}	l^{task_2}	l^{reset}
$User_1$	-1	1	0
$User_2$	1	-1	0
$Provider_1$	-1	0	1
$Provider_2$	1	0	-1

(b) Activity Matrix $\mathbf{C^{Pre}}$

	l^{task_1}	l^{task_2}	l^{reset}
$User_1$	1	0	0
$User_2$	0	1	0
$Provider_1$	1	0	0
$Provider_2$	0	0	1

Table 4.1: *Activity matrix and pre activity matrix of Model 1*

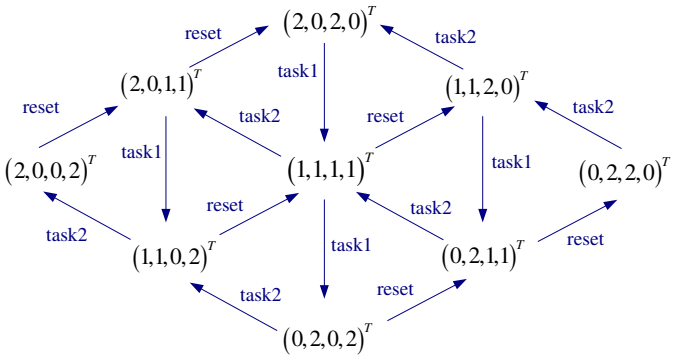


Figure 4.1: *Transition diagram of Model 1 ($M = N = 2$)*

$\mathbf{m} \geq (1,0,1,0)^T$, i.e. $\mathbf{m} \geq \mathbf{C^{Pre}}(\cdot, l^{task_1})$, where $\mathbf{C^{Pre}}$ is the pre activity matrix. Throughout this thesis, a vector being greater than another vector means that each entry of the former one is greater than the corresponding entry of the latter one. Each column of the pre activity matrix reflects the required condition for enabling the labelled activity corresponding to this column. That is, if a state $\mathbf{m}$ is equal or greater than a column of the pre activity matrix, then the corresponding activity can be enabled at $\mathbf{m}$. Otherwise, $\mathbf{m}$ cannot fire this activity. For example, at the starting state $\mathbf{m}_0$ of Model 1, $task_1$ can be fired because $\mathbf{m}_0 \geq \mathbf{C^{Pre}}(\cdot, l^{task_1})$. Both $task_2$ and $reset$ cannot be enabled due to $\mathbf{m_0} \ngeq \mathbf{C^{Pre}}(\cdot, l^{task_2})$ and $\mathbf{m_0} \ngeq \mathbf{C^{Pre}}(\cdot, l^{reset})$.

After firing $task_1$ at $\mathbf{m_0}$, there is one $User_1$ and one $Provider_1$ changing into $User_2$ and $Provider_2$ respectively and simultaneously. So the state vector becomes $\mathbf{m_1} = (1,1,1,1)^T$. Notice that the column of the activity matrix corresponding to $task_1$, i.e. l^{task_1}, can fully reflect this transition. Therefore the mathematical expression using l^{task_1} is

$$\mathbf{m_1} = \mathbf{m_0} + l^{task1}.$$

Since $\mathbf{m_1} \geq \mathbf{C^{Pre}}(\cdot, l^{task_2})$ and $\mathbf{m_1} \geq \mathbf{C^{Pre}}(\cdot, l^{reset})$, $\mathbf{m_1}$ can fire either $task_2$ or $reset$.

Suppose $task_2$ is fired, then we get $\mathbf{m_2} = (2, 0, 0, 1)^T$. That is

$$\mathbf{m_2} = \mathbf{m_1} + l^{task2} = \mathbf{m_0} + l^{task1} + l^{task2}. \tag{4.1}$$

Notice that each column of a matrix can be extracted by multiplying a corresponding vector at the right side of this matrix:

$$l^{task_1} = \mathbf{C}(1, 0, 0)^T, \quad l^{task_2} = \mathbf{C}(0, 1, 0)^T, \quad l^{reset} = \mathbf{C}(0, 0, 1)^T.$$

So (4.1) can be written as

$$\mathbf{m_2} = \mathbf{m_0} + l^{task1} + l^{task2} = \mathbf{m_0} + \mathbf{C}(1, 1, 0)^T. \tag{4.2}$$

Generally, the firing of a labelled activity l in state $\mathbf{m}$ yields the state $\mathbf{m}'$, denoted by $\mathbf{m} \xrightarrow{l} \mathbf{m}'$, can be expressed as

$$\mathbf{m}' = \mathbf{m} + \mathbf{C}(\cdot, l) \tag{4.3}$$

where $\mathbf{C}(\cdot, l)$ is the transition vector corresponding to l, i.e. the column of the activity matrix that corresponds to l. If a firing sequence $\sigma = l_1 \cdots l_k \cdots l \in \mathcal{A}^{\omega}_{\text{label}}$ from $\mathbf{m}_0$ yields the state $\mathbf{m}$, i.e.

$$\mathbf{m} \xrightarrow{l_1} \mathbf{m_1} \cdots \xrightarrow{l_k} \mathbf{m_k} \cdots \xrightarrow{l} \mathbf{m},$$

then we denote $\mathbf{m_0} \xrightarrow{\sigma} \mathbf{m}$. We define the *firing count vector* of a sequence σ as $\boldsymbol{\sigma}[l] = \sharp(l, \sigma)$, where $\sharp(l, \sigma)$ is the number of occurrences of l in σ. Integrating the evolution equation in (4.3) from $\mathbf{m_0}$ to $\mathbf{m}$ we get:

$$\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma}. \tag{4.4}$$

The formula (4.4) is called the *state equation*, reflecting that each state in the state space is related to the starting state through an algebraic equation. This is consistent with the fact that each system state results from the evolution of the system from the starting state. Of course, the state equation does not involve the time information, so it cannot be used to do quantitative analysis. However, for qualitative analysis of the system it is powerful.

4.2.2 Place/Transition Structure in PEPA Models

Observe the activity matrix $\mathbf{C}$ in Table 4.1, each local derivative is in fact like a *place* and the state vector records the population of components in each place, i.e. each local derivative. Each transition vector, i.e. each column of the activity matrix represents the *transition* of components from one place to another place. Such structure involving "place" and "transition" can be formally defined. See the following concepts of *P/T net* and *P/T system*, which originate in Petri net theory ("P/T" signifies "place/transition") but they can also be interpreted in terms of conditions and events.

Definition 4.2.1. ***(P/T net, Marking, P/T system**, [CTS98])*

1. *A P/T net is a structure* $\mathcal{N} = (P, T, \mathbf{Pre}, \mathbf{Post})$ *where:* P *and* T *are the sets of* places *and* transitions *respectively;* $\mathbf{Pre}$ *and* $\mathbf{Post}$ *are the* $|P| \times |T|$ *sized, natural valued,* incidence matrices.

2. *A* marking *is a vector* $\mathbf{m} : P \to \mathbb{N}$ *that assigns to each place of a P/T net a nonnegative integer.*

3. *A* P/T system *is a pair* $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$*: a net* $\mathcal{N}$ *with an initial marking* $\mathbf{m}_0$.

In order to take advantage of the theory developed for P/T systems in the context of PEPA models we must first establish the P/T system corresponding to a given PEPA model. This is straightforward given the definitions presented in the previous chapter.

From Definition 4.2.1, it is easy to see that the structure $\mathcal{N} = \left(\mathcal{D}, \mathcal{A}_{\text{label}}, \mathbf{C}^{\mathbf{Pre}}, \mathbf{C}^{\mathbf{Post}}\right)$ derived from a PEPA model is a P/T net, where $\mathcal{D}, \mathcal{A}_{\text{label}}$ are the sets of all local derivatives and all labelled activities of the PEPA model respectively, and $\mathbf{C}^{\mathbf{Pre}}, \mathbf{C}^{\mathbf{Post}}$ are the pre and post activity matrices respectively. Given a starting state $\mathbf{m}_0$, $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ is a P/T system. Clearly, each reachable marking $\mathbf{m}$ from $\mathbf{m}_0$ is a state of the aggregated CTMC underlying the given PEPA model. This leads us to:

Theorem 4.2.1. *There is a P/T system underlying any PEPA model, that is* $\langle \mathcal{N}, \mathbf{m}_0 \rangle$*, where* $\mathbf{m}_0$ *is the starting state;* $\mathcal{N} = \left(\mathcal{D}, \mathcal{A}_{\text{label}}, \mathbf{C}^{\mathbf{Pre}}, \mathbf{C}^{\mathbf{Post}}\right)$ *is P/T net: where* $\mathcal{D}$ *is the local derivative set,* $\mathcal{A}_{\text{label}}$ *is the labelled activity set;* $\mathbf{C}^{\mathbf{Pre}}$ *and* $\mathbf{C}^{\mathbf{Post}}$ *are the pre and post activity matrices respectively.*

Remark 4.2.1. *Throughout this thesis, the P/T structure underlying a PEPA is referred to as* $\mathcal{N} = \left(\mathcal{D}, \mathcal{A}_{\text{label}}, \mathbf{C}^{\mathbf{Pre}}, \mathbf{C}^{\mathbf{Post}}\right)$ *or* $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$*, which are constituted by the local derivative and labelled activity sets, the pre activity matrix and post activity matrix, as well as the starting state.*

A P/T net, a particular class of Petri net, like PEPA provides a mathematical modelling language for discrete, distributed systems. A Petri net or P/T net associated with time information, i.e. the transitions are associated with time delays, is called a *timed Petri net* [Chi98] or a *timed P/T net*. In particular, if the delays are random variables, usually satisfying exponentially distributions, the timed Petri net is called a *stochastic Petri net*. In the PEPA language, the delay of a transition l in a state $\mathbf{m}$ is specified by the transition rate function $f(\mathbf{m}, l)$ which are defined in Chapter 3. For any given PEPA model, the underlying P/T net with incorporated transition rate functions is obviously a stochastic Petri net.

In [Rib95] Ribaudo has defined a stochastic Petri net semantics for stochastic process algebras, including PEPA. As in our work here, her approach associated each local derivative with a place and each activity with a transition. To cope with the difference between action types and transitions, she defined a labelling function that maps transition names into action names. Similarly, our approach is to attach distinct labels to each action name, as indicated by the definition of labelled activity. However, since Ribaudo's approach does not include aggregation as we do, the mapping semantics in [Rib95] does not help with the state-space explosion problem in structural analysis for large scale PEPA models with repeated components. In fact, since the instances of the same component type are considered as distinct copies, their local derivatives are consequently distinct. So the number of places will increase with the number of repeated components, which is in contrast to the fixed number of places in our approach.

Moreover, our transition rate functions that capture the time information are defined on each system state and each transition. Therefore, our approach is more convenient for quantitative application, such as simulation and fluid approximation for PEPA models. Ribaudo's work was motivated by investigation into the relationship between formalisms whereas our work is more application-oriented. We should point out that although our approach seems more mathematical, the definitions of labelled activities and transition rate functions are essentially syntactical.

Previous work on structural analysis of PEPA models in [GHR97] has some similarities with

our approach. However, the class of PEPA considered in [GHR97] is somewhat restricted; in particular no repeated components are allowed, which is also because no aggregation technique is employed. Moreover, the problem of the difference between actions and transitions is not considered. Furthermore, there is no time information considered in [GHR97], and therefore their considerations cannot be extended to quantitative analysis. For convenience, some comparison between the work by different authors are presented in the following table.

	Ribaudo [Rib95]	Gilmore et al. [GHR97]	This thesis
Mathematical representation	No	Yes	Yes
Syntactical representation	Yes	No	Yes
Time involved	Yes	No	Yes
Generality for PEPA	Yes	No	Yes
Aggregation technique	No	No	Yes
Derivation algorithm	No	No	Yes
Suitable for qualitative analysis	No	Yes	Yes
Suitable for quantitative analysis	No	n/a	Yes
Suitable for simulation	No	n/a	Yes

Table 4.2: *Comparison between three approaches*

4.2.3 Some terminology

Now we introduce some terminology related to P/T systems for PEPA models (see [CTS98] for reference). For convenience, if the concepts and conclusions that are defined and given for P/T systems are used in the context of the PEPA language, they will be referred to as the P/T structure underlying PEPA models.

As illustrated by the example in the last subsection, a transition l is *enabled* in a state $\mathbf{m}$ if and only if $\mathbf{m} \geq \mathbf{C}^{\mathbf{Pre}}[\cdot, l]$; its firing yields a new state $\mathbf{m}' = \mathbf{m} + \mathbf{C}[\cdot, l]$. This fact is denoted by $\mathbf{m} \xrightarrow{l} \mathbf{m}'$. We should point out that l is enabled in $\mathbf{m}$ can be equivalently stated using the transition rate function $f(\mathbf{m}, l)$ defined in the previous chapter. In fact, we have a proposition:

Proposition 4.2.1. *Let $f(\mathbf{m}, l)$ be the transition rate function given by Definition 3.4.2 in Chapter 3, then*

$$\mathbf{m} \geq \mathbf{C}^{\mathbf{Pre}}[\cdot, l] \Longleftrightarrow f(\mathbf{m}, l) > 0. \tag{4.5}$$

Proof. Notice that l is a labelled activity. First, assume l is individual. Then l has only one pre local derivative, namely U. So $\mathbf{C}^{\mathbf{Pre}}[\cdot, l] = \mathbf{e}_U$, where $\mathbf{e}_U$ is a vector with all entries being zeros

except $\mathbf{e}_U[U] = 1$. Thus $\mathbf{m} \geq \mathbf{C^{Pre}}[\cdot, l]$ implies $\mathbf{m}[U] \geq \mathbf{e}_U[U] = 1$. By Definition 3.4.2, $f(\mathbf{m}, l) = r\mathbf{m}[U] > 0$, where r is a positive constant. Conversely, if $f(\mathbf{m}, l) = r\mathbf{m}[U] > 0$, then $\mathbf{m}[U] > 0$. Since each entry of a state vector is an integer, so we have $\mathbf{m}[U] \geq 1$. Therefore, $\mathbf{m} \geq \mathbf{e}_U$ and thus $\mathbf{m} \geq \mathbf{C^{Pre}}[\cdot, l]$.

Secondly, we assume l is a shared labelled activity with the pre set $\{U_1, U_2, \cdots, U_k\}$. So $\mathbf{C^{Pre}}[\cdot, l]$ is such a vector $\mathbf{e}'$ with all entries zero except $\mathbf{e}'[U_i] = 1,\ i = 1, 2, \cdots, k$. Therefore, $\mathbf{m} \geq \mathbf{C^{Pre}}[\cdot, l]$ implies $\mathbf{m}[U_i] \geq \mathbf{e}'[U_i] = 1$. So $f(\mathbf{m}, l) > 0$, where $f(\mathbf{m}, l)$ is given by (3.10) in Definition 3.4.2, i.e.

$$f(\mathbf{m}, l) = \left(\prod_{i=1}^{k} \frac{r_l^{U_i \to V_i}}{r_l(U_i)} \right) \min_{i \in \{1, \cdots, k\}} \{\mathbf{m}[U_i] r_l(U_i)\},$$

where $r_l^{U_i \to V_i}, r_l(U_i)$ are some positive constants. Conversely, $f(\mathbf{m}, l) > 0$ implies $\mathbf{m}[U_i] \geq 1$ and thus implies $\mathbf{m} \geq \mathbf{C^{Pre}}[\cdot, l]$. □

According to this proposition, if $f(\mathbf{m}, l) = 0$ then l cannot be enabled in $\mathbf{m}$. That is, $\mathbf{m} \ngeq \mathbf{C^{Pre}}[\cdot, l]$. Therefore, a transition rate function not only specifies the rate of the activity but also determines whether the activity can be enabled in a state. Since this chapter emphasises structural rather than quantitative aspects of PEPA, we prefer to use the comparison between the state vector and the column of the pre activity matrix, to determine whether a state can enable an activity.

An *occurrence sequence* from $\mathbf{m}$ is a sequence of transitions $\sigma = t_1 \cdots t_k \cdots$ such that $\mathbf{m} \xrightarrow{t_1} \mathbf{m_1} \cdots \xrightarrow{t_k} \mathbf{m_k} \cdots$. The *language* of $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$, denoted by $L(\mathcal{S})$ or $L(\mathcal{N}, \mathbf{m_0})$, is the set of all the occurrence sequences from the starting state $\mathbf{m}_0$. A state $\mathbf{m}$ is said to be *reachable* from $\mathbf{m}_0$ if there exists a σ in $L(\mathcal{S})$ such that $\mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}$, that is

$$\mathbf{m} = \mathbf{m}_0 + \mathbf{C} \cdot \boldsymbol{\sigma},$$

where $\boldsymbol{\sigma}$ is the *firing count vector* corresponding to σ. The set of all the reachable states from $\mathbf{m}$, called the *reachability set* from $\mathbf{m}$, is denoted by RS$(\mathcal{N}, \mathbf{m})$. According to the definition, the reachability set of the P/T system $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ is

$$\text{RS}(\mathcal{N}, \mathbf{m}_0) = \left\{ \mathbf{m} \in \mathbb{N}^{|P|} \mid \exists \sigma \in L(\mathcal{S}) \text{ such that } \mathbf{m} = \mathbf{m}_0 + \mathbf{C} \cdot \boldsymbol{\sigma} \right\},$$

where $\boldsymbol{\sigma}$ is the firing count vector of the occurrence sequence σ, $|P|$ represents the number of elements in P, i.e. $|P| = \#P$. Clearly, the reachability set $\mathrm{RS}(\mathcal{N}, \mathbf{m}_0)$ is the state space of the CTMC underlying the given PEPA model starting from $\mathbf{m}_0$. The correspondence between P/T systems and PEPA models is shown in Table 4.3.

P/T terminology	PEPA terminology
P: place set	$\mathcal{D}$: local derivative set
T: transition set	$\mathcal{A}_{\mathrm{label}}$: labelled activity set
$\mathbf{Pre}$: pre matrix	$\mathbf{C}^{\mathbf{Pre}}$: pre activity matrix
$\mathbf{Post}$: post matrix	$\mathbf{C}^{\mathbf{Post}}$: post activity matrix
$\mathbf{C} = \mathbf{Pre} - \mathbf{Post}$: incidence matrix	$\mathbf{C} = \mathbf{C}^{\mathbf{Pre}} - \mathbf{C}^{\mathbf{Post}}$: activity matrix
$\mathbf{m}$: marking	$\mathbf{m}$: state vector
$\mathrm{RS}(\mathcal{N}, \mathbf{m_0})$: reachbility set (from $\mathbf{m_0}$)	$\mathrm{RS}(\mathcal{N}, \mathbf{m_0})$: state space (with starting state $\mathbf{m_0}$)

Table 4.3: *P/T structure in PEPA models*

For each PEPA model, as Theorem 4.2.1 reveals, there is a underlying P/T structure. This structure involves the pre and post activity matrices and so involves the activity matrix and captures the structure information for the given PEPA model. Therefore, the fruitful theories developed for P/T systems in the past twenty years can be utilised to investigate the structural properties of PEPA models. Of course, our studies in the context of PEPA, in particular the efficient deadlock-checking method, are also valid for some classes of P/T systems.

4.3 Invariance in PEPA models

Invariance characterises a kind of structural property of each state and a relationship amongst all component types. In this section, we will show what an invariant is and how to find invariants for a given PEPA model.

4.3.1 What are invariants

Let us first consider an interesting system composed of two types of components, namely X and Y, which are synchronised through the shared activities $action1$ and $action2$. The operations of X and Y are illustrated in Figure 4.2. The PEPA model of the system is as below:

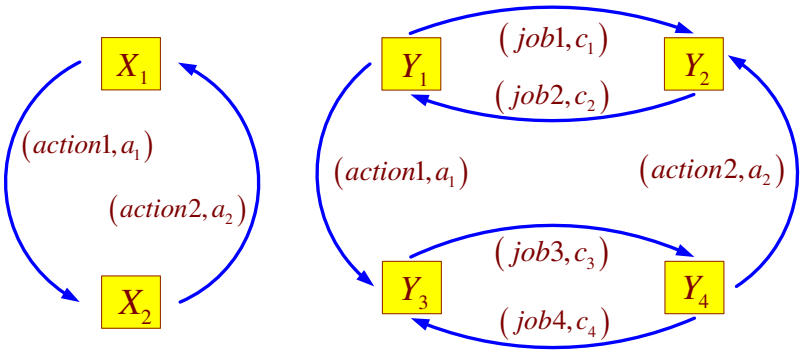


Figure 4.2: *Transition systems of the components of Model 3*

Model 3.

$$
\begin{array}{rcl}
X_1 & \stackrel{\text{def}}{=} & (action1\,,\, a_1).X_2 \\
X_2 & \stackrel{\text{def}}{=} & (action2\,,\, a_2).X_1 \\
Y_1 & \stackrel{\text{def}}{=} & (action1\,,\, a_1).Y_3 + (job1\,,\, c_1).Y_2 \\
Y_2 & \stackrel{\text{def}}{=} & (job2\,,\, c_2).Y_1 \\
Y_3 & \stackrel{\text{def}}{=} & (job3\,,\, c_3).Y_4 \\
Y_4 & \stackrel{\text{def}}{=} & (action2\,,\, a_2).Y_2 + (job4\,,\, c_4).Y_3 \\
(X_1[M_1] \parallel X_2[M_2]) & \underset{\{action1,action2\}}{\bowtie} & (Y_1[N_1] \parallel Y_2[N_2] \parallel Y_3[N_3] \parallel Y_4[N_4])\,.
\end{array}
$$

Let $\mathbf{m}[X_i], \mathbf{m}[Y_i]$ $(i = 1, 2,\ j = 1, 2, 3, 4)$ denote the numbers of the components X and Y in the local derivatives X_i, Y_j respectively. Now we state an interesting **assertion** for the specific PEPA model: the difference between the number of Y in their local derivatives Y_3 and Y_4, and the number of X in the local derivative X_2, i.e. $\mathbf{m}[Y_3] + \mathbf{m}[Y_4] - \mathbf{m}[X_2]$, is a constant in any state. This fact can be explained as follows. Notice that there is only one way to increase $\mathbf{m}[Y_3] + \mathbf{m}[Y_4]$, i.e. enabling the activity $action1$. As long as $action1$ is activated, then there is a copy of Y entering Y_3 from Y_1. Meanwhile, since $action1$ is shared by X, a corresponding copy of X will go to X_2 from X_1. In other words, $\mathbf{m}[Y_3] + \mathbf{m}[Y_4]$ and $\mathbf{m}[X_2]$ increase equally and simultaneously. On the other hand, there is also only one way to decrease $\mathbf{m}[Y_3] + \mathbf{m}[Y_4]$ and $\mathbf{m}[X_2]$, i.e. enabling the cooperated activity $action2$. This also allows $\mathbf{m}[Y_3] + \mathbf{m}[Y_4]$ and $\mathbf{m}[X_2]$ to decrease both equally and simultaneously. So, the difference $\mathbf{m}[Y_3] + \mathbf{m}[Y_4] - \mathbf{m}[X_2]$ will remain constant in any state and thus at any time.

The assertion indicates that each state and therefore the whole state space of the underlying

CTMC may have some interesting structural properties, such as invariants. A natural question is how we can easily find all invariants in a general PEPA model. Before investigating this problem, we need to define "invariant" first.

Definition 4.3.1. *(**Invariant**) An invariant of a given PEPA model is a vector* $\mathbf{y} \in \mathbb{Q}$ *such that for any state* $\mathbf{m}$ *in the state space,* $\mathbf{y}^T\mathbf{m}$ *is a constant, or equivalently*

$$\mathbf{y}^T\mathbf{m} = \mathbf{y}^T\mathbf{m_0}, \tag{4.6}$$

since the starting state $\mathbf{m_0}$ *is a state and the constant is just* $\mathbf{y}^T\mathbf{m_0}$.

The assertion discussed previously, i.e. "$\mathbf{m}[Y_3] + \mathbf{m}[Y_4] - \mathbf{m}[X_2]$ is a constant", can be illustrated by this definition. That is, $\mathbf{y} = (0, -1, 0, 0, 1, 1)^T$ is an invariant of Model 3, since

$$\mathbf{y}^T\mathbf{m} = \mathbf{m}[Y_3] + \mathbf{m}[Y_4] - \mathbf{m}[X_2] \tag{4.7}$$

is a constant.

Once discovered, invariants may have potential applications such as model-based reasoning. For example, based on the information on the server side, we may infer information about the clients via an invariance relationship between them. For instance, by (4.7) and the number of X_2 we can know the numbers of Y in the local derivatives Y_3 and Y_4. In Chapter 7, a case study shows how invariance can be used to prove the convergence of the fluid approximation of PEPA models.

4.3.2 How to find invariants

In this subsection, we demonstrate how to find invariants in a given PEPA model. For any $\mathbf{m}$ in the state space, there exists a corresponding sequence σ such that $\mathbf{m} = \mathbf{m_0} + \mathbf{C}\boldsymbol{\sigma}$. Multiplying both sides of this equation by $\mathbf{y}^T$, we have

$$\mathbf{y}^T\mathbf{m} = \mathbf{y}^T\mathbf{m_0} + \mathbf{y}^T\mathbf{C}\boldsymbol{\sigma}.$$

Obviously, $\mathbf{y}^T\mathbf{m} = \mathbf{y}^T\mathbf{m_0}$ holds if and only if $\mathbf{y}^T\mathbf{C}\boldsymbol{\sigma} = 0$. Therefore, the following lemma is ready.

Lemma 4.3.1. *If* $\mathbf{y}^T\mathbf{C} = 0$*, then* $\mathbf{y}$ *is an invariant.*

Lemma 4.3.1 provides a method to find invariants: any solution of $\mathbf{y}^T\mathbf{C} = 0$, i.e. $\mathbf{C}^T\mathbf{y} = 0$, is an invariant. For Model 3, its activity matrix $\mathbf{C}$ is listed in Table 4.4 (the labels of those labelled activities are omitted since there are no confusions).

	$action1$	$action2$	$job1$	$job2$	$job3$	$job4$
X_1	-1	1	0	0	0	0
X_2	1	-1	0	0	0	0
Y_1	-1	0	-1	1	0	0
Y_2	0	1	1	-1	0	0
Y_3	1	0	0	0	-1	1
Y_4	0	-1	0	0	1	-1

Table 4.4: *Activity matrix of Model 3*

That is,

$$C = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 1 & 0 & 0 \\ 0 & 1 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 1 \\ 0 & -1 & 0 & 0 & 1 & -1 \end{pmatrix}. \tag{4.8}$$

Now we try to solve the linear algebraic equation $\mathbf{C}^T\mathbf{y} = 0$. The rank of $\mathbf{C}$ is three. So by linear algebra theory the rank of the solution space $\{\mathbf{y} : \mathbf{C}^T\mathbf{y} = 0\}$ is $6 - 3 = 3$. We can easily find three vectors which form the bases of the solution space:

$$\mathbf{y}_1 = (1, 1, 0, 0, 0, 0)^T, \ \ \mathbf{y}_2 = (0, 0, 1, 1, 1, 1)^T, \ \ \mathbf{y}_3 = (0, 1, 1, 1, 0, 0)^T.$$

These vectors are invariants of Model 3. Check $\mathbf{y}_1 = (1, 1, 0, 0, 0, 0)^T$ first,

$$\mathbf{y}_1^T\mathbf{m} = \mathbf{y}_1^T\mathbf{m}_0 + \mathbf{y}_1^T\mathbf{C}\boldsymbol{\sigma} = \mathbf{y}_1^T\mathbf{m}_0.$$

That is, for any state $\mathbf{m}$,

$$\mathbf{m}[X_1] + \mathbf{m}[X_2] = \mathbf{m}_0[X_1] + \mathbf{m}_0[X_2].$$

In other words, the population of X, i.e. the sum of the instances of X_1 and X_2, is a constant in any state (thus at any time), which is usually termed *conservation law* satisfied by X.

Similarly, $\mathbf{y}_2 = (0,0,1,1,1,1)^T$ illustrates the conservation law satisfied by the component Y: $\mathbf{m}[Y_1]+\mathbf{m}[Y_2]+\mathbf{m}[Y_3]+\mathbf{m}[Y_4]$ is a constant for any state $\mathbf{m}$. Moreover, $\mathbf{y}_3 = (0,1,1,1,0,0)^T$ means that for any state $\mathbf{m}$,

$$\mathbf{m}[X_2] + \mathbf{m}[Y_1] + \mathbf{m}[Y_2]$$

is a constant. That is, the sum of the populations in X_2, Y_1 and Y_2 always remains unchanged.

According to the definition of invariants, any linear combination of invariants is also an invariant. For example, the following combination of $\mathbf{y}_2$ and $\mathbf{y}_3$:

$$\mathbf{y}_4 = \mathbf{y}_2 - \mathbf{y}_3 = (0,-1,0,0,1,1)^T$$

is an invariant. Notice $\mathbf{y}_4^T\mathbf{C} = 0$, which implies that

$$\mathbf{m}[Y_3] + \mathbf{m}[Y_4] - \mathbf{m}[X_2]$$

is a constant. This coincides with the assertion mentioned at the beginning of this section.

We should point out that $\mathbf{y}^T\mathbf{C} = 0$ is not a necessary condition for some invariant $\mathbf{y}$. For example, consider Model 3 with $\mathbf{m_0} = (100,0,0,0,0,3)^T$. Then the state space of the underlying aggregated CTMC has four elements: $\mathbf{m}_0$ and

$$\mathbf{m}_1 = (100,0,0,0,1,2)^T, \quad \mathbf{m}_2 = (100,0,0,0,2,1)^T, \quad \mathbf{m}_3 = (100,0,0,0,3,0)^T.$$

$\mathbf{y} = (0,1,1,0,0,0)^T$ is an invariant since $\mathbf{y}^T\mathbf{m}_k = 0$ $(k = 0,1,2,3)$, but $\mathbf{y}^T\mathbf{C} \neq 0$.

However, for a class of PEPA models, i.e. live PEPA models, the inverse of Lemma 4.3.1 is true: $\mathbf{y}$ is an invariant can imply $\mathbf{y}^T\mathbf{C} = 0$.

Definition 4.3.2. *(**Liveness for PEPA**). Denote by $\langle \mathcal{N}, \mathbf{m_0} \rangle$ the P/T structure underlying a given PEPA model.*

1. *A labelled activity l is* live *if for any derivative in the derivative set, there exists a sequence of activities such that the derivative after performing this sequence can perform an l activity.*
2. *If all activities are live, then both $\langle \mathcal{N}, \mathbf{m_0} \rangle$ and the PEPA model are said to be* live.

The liveness defined for PEPA is originally given for P/T nets (see [CTS98]). For some PEPA

models, if they have no deadlocks then they are live, see Lemma 4.5.4 in this chapter. The following proposition directly derives from a conclusion in P/T theory (page 319, [STC96]): for a live P/T net $\mathbf{y}^T\mathbf{C} = 0$ is equivalent to $\mathbf{y}^T\mathbf{m} = \mathbf{y}^T\mathbf{m_0}$.

Proposition 4.3.1. *If a given PEPA model is live, i.e. the underlying P/T structure* $\left\langle \left(\mathcal{D}, \mathcal{A}_{\text{label}}, \mathbf{C^{Pre}}, \mathbf{C^{Post}}\right), \mathbf{m_0}\right\rangle$ *is live, then for any state* $\mathbf{m}$ *in the state space,*

$$\mathbf{y}^T\mathbf{C} = 0 \Longleftrightarrow \mathbf{y}^T\mathbf{m} = \mathbf{y}^T\mathbf{m_0}.$$

For the class of live PEPA models, finding invariants is much simpler—just solve the activity-matrix-based linear algebric equation $\mathbf{C}\mathbf{y}^T = 0$.

4.3.3 Conservation law as a kind of invariance

In the last subsection, we have seen that Model 3 satisfies a conservation laws, i.e. the population of each component type is a constant in any state. In fact, this kind of conservation law is universal for any PEPA model.

Let P be an arbitrary component type of a given PEPA model. The state transition of component type P, i.e. any instance of P changing from one local derivative to another local derivative, must occur within the set of P's local derivatives. That is, it is not possible to change into any other component type's local derivative. For an arbitrary labelled activity l, if there is a pre local derivative of l, there must exist a post local derivative of l and this post local derivative must be within component type P. Therefore, in each column of the activity matrix "1" and "-1" must appear in a pair within the subvector corresponding to component type P. That is, the sum of these "1" and "-1" within any component type is zero. So we have

Lemma 4.3.2. *Let* $\mathbf{C}$ *be the activity matrix of a given PEPA model,* $\mathcal{D}$ *be the set of all local derivatives. For an arbitrary component type* P *of this model, let* $\mathcal{D}_P$ *be the local derivative set of* P*. Define a vector* $\mathbf{y}_D$ *with* $\#\mathcal{D}$ *entries:*

$$\mathbf{y}_P[U] = \begin{cases} 1 & \text{if } U \in \mathcal{D}_P \\ 0 & \text{if } U \in \mathcal{D}\backslash\mathcal{D}_P \end{cases}$$

Then $\mathbf{y}_P^T\mathbf{C} = 0$.

Remark 4.3.1. *Obviously,* $\mathbf{y}_P$ *in Lemma 4.3.2 is an invariant by Lemma 4.3.1. Let* $\mathcal{P}$ *be the*

set of all component types of the given PEPA model. According to the definition of invariant, a linear combination of invariants is also an invariant. So the sum $\sum_{P\in\mathcal{P}} \mathbf{y}_P = \mathbf{1}$ *is also an invariant. In fact,* $\mathbf{1}^T\mathbf{C} = \sum_{P\in\mathcal{P}} \mathbf{y}_P^T\mathbf{C} = 0$.

Lemma 4.3.2 and Lemma 4.3.1 imply the following: the population of each component type and therefore all component types in any state are constants. This fact is termed the conservation law.

Proposition 4.3.2. *(**Conservation Law**) For a given PEPA model, let* $\mathcal{D}$ *be the local derivative set. For an arbitrary component type* $P \in \mathcal{P}$*, let* $\mathcal{D}_P$ *be the local derivative set of* P*. Then*

$$\sum_{U\in\mathcal{D}_P} \mathbf{m}[U] = \sum_{U\in\mathcal{D}_P} \mathbf{m_0}[U]. \tag{4.9}$$

$$\sum_{U\in\mathcal{D}} \mathbf{m}[U] = \sum_{U\in\mathcal{D}} \mathbf{m_0}[U]. \tag{4.10}$$

Proof. For any $\mathbf{m}$, there exists a σ such that $\mathbf{m} = \mathbf{m}_0 + \mathbf{C}\boldsymbol{\sigma}$. By Lemma 4.3.2, $\mathbf{y}_P^T\mathbf{C} = 0$ where $\mathbf{y}_P^T$ is given in this lemma. So we have

$$\sum_{U\in\mathcal{D}_P} \mathbf{m}[U] = \mathbf{y}_P^T\mathbf{m} = \mathbf{y}_P^T(\mathbf{m_0} + \mathbf{C}\boldsymbol{\sigma}) = \mathbf{y}_P^T\mathbf{m_0} = \sum_{U\in\mathcal{D}_P} \mathbf{m_0}[U].$$

Moreover, let $\mathcal{P}$ be the set of all component types of this model, then

$$\sum_{U\in\mathcal{D}} \mathbf{m}[U] = \sum_{P\in\mathcal{P}}\sum_{U\in\mathcal{D}_P} \mathbf{m}[U] = \sum_{P\in\mathcal{P}}\sum_{U\in\mathcal{D}_P} \mathbf{m_0}[U] = \sum_{U\in\mathcal{D}} \mathbf{m}_0[U].$$

□

Proposition 4.3.2 can easily lead to the boundedness of the underlying state space.

Corollary 4.3.3. *(**Boundedness**) The state space underlying a PEPA model is bounded: for any state* $\mathbf{m}$ *and any local derivative* U*,*

$$0 \le \mathbf{m}[U] \le \max_{P\in\mathcal{P}} \left\{ \sum_{U\in\mathcal{D}_P} \mathbf{m}_0[U] \right\},$$

where $\mathcal{P}$ *is the component type set,* $\mathcal{D}_P$ *is the local derivative set corresponding to component type* P*,* $\mathbf{m}_0$ *is the starting state.*

4.4 Linearisation of State Space for PEPA

As discussed in Section 3.2.3, the size of the state space underlying Model 1 is $(M+1)\times(N+1)$. When M, N are large, the size is consequently large. Recall the following table:

(M, N)	(300,300)	(350,300)	(400,300)	(400,400)
time	2879 ms	4236 ms	"Java heap space"	"GC overhead limit exceeded"

Table 4.5: *Elapsed time of state space derivation*

If there are 400 users and providers in the system, the support tool of PEPA reports the error message of "GC overhead limit exceeded". That is, the derivation of the state space becomes impossible, let alone the storage of the state space and the deadlock-checking. Some questions are naturally proposed: is there a better representation for the underlying state space which can be derived and stored without the restriction of the number of components? Is there a more efficient deadlock-checking algorithm which does not suffer the size of the state space, i.e. avoid the state-space explosion problem? In the next section, some efficient deadlock-checking methods will be presented. This section gives a new presentation of state space, to solve the derivation and storage problems encountered in large scale PEPA models.

4.4.1 Linearisation of state space

As shown in Section 4.2, the states of a PEPA model can be expressed using the state equations. If the state space of the underlying Markov chain can be described using some linear equations, then the storage of the state space will be much more efficient and easier. This section presents the linearised description of state space for PEPA. As we mentioned, the terminology of reachability set is is also used to refer to the corresponding state space.

According to the definition, the reachability set $\mathrm{RS}(\mathcal{S})$ of a given PEPA model with the activity matrix $\mathbf{C}$ and starting state $\mathbf{m}_0$ is

$$\mathrm{RS}(\mathcal{N}, \mathbf{m_0}) = \left\{ \mathbf{m} \in \mathbb{N}^{|\mathcal{D}|} \mid \exists \sigma \in L(\mathcal{S}) \text{ such that } \mathbf{m} = \mathbf{m}_0 + \mathbf{C} \cdot \boldsymbol{\sigma} \right\}.$$

The reachability set given above is descriptive rather than constructive. So it does not help us to derive and store the entire state space. Moreover, we should point out that, for some given

$\boldsymbol{\sigma} \in \mathbb{N}^{|\mathcal{D}|}$, $\mathbf{m} = \mathbf{m_0} + \mathbf{C}\boldsymbol{\sigma} \in \mathbb{N}^{|\mathcal{D}|}$ may not be valid states because there may be no valid occurrence sequences corresponding to these $\boldsymbol{\sigma}$. However $\mathbf{m}$ is said to belong to a generalisation of the reachability set: the *linearised reachability set*. Before giving these definitions, we first define some kind of flow and semiflow in the context of PEPA. (For the definitions of these concepts in the context of P/T systems, see [CTS98]).

Definition 4.4.1. *(**Flow, Semiflow, Conservative and Consistent**). Let $\mathbf{C}$ be the activity matrix of a given PEPA model with the underlying P/T structure $\left\langle \left(\mathcal{D}, \mathcal{A}_{\text{label}}, \mathbf{C^{Pre}}, \mathbf{C^{Post}}\right), \mathbf{m_0} \right\rangle$.*

1. *A* p-flow *is a vector $\mathbf{y} : \mathcal{D} \to \mathbb{Q}$ such that $\mathbf{y}^T\mathbf{C} = 0$. Natural and nonnegative flows are called* semiflows*: vectors $\mathbf{y} : \mathcal{D} \to \mathbb{N}$ such that $\mathbf{y}^T\mathbf{C} = 0$. The model is* conservative *if there exists a p-semiflow whose support covers $\mathcal{D}$, that is $\{U \in \mathcal{D} \mid \mathbf{y}[U] > 0\} = \mathcal{D}$.*

2. *A* basis *(respectively,* fundamental set*) of p-flows (respectively p-semiflows), $\mathbf{B} = \{\mathbf{y_1}, \mathbf{y_2}, \cdots, \mathbf{y_q}\}$ (respectively, $\mathbf{\Phi} = \{\mathbf{y_1}, \mathbf{y_2}, \cdots, \mathbf{y_q}\}$) is a minimal subset which will generate any p-flow (respectively, p-semiflow) as follows: $\mathbf{y} = \sum_{\mathbf{y}_j \in \mathbf{\Psi}} k_j \mathbf{y}_j$, $k_j \in \mathbb{Q}$.*

3. *A* t-flow *is a vector $\mathbf{x} : \mathcal{A}_{\text{label}} \to \mathbb{Q}$ such that $\mathbf{Cx} = 0$. Natural and nonnegative flows are called* semiflows*: vectors $\mathbf{x} : \mathcal{A}_{\text{label}} \to \mathbb{N}$ such that $\mathbf{Cx} = 0$. The model is* consistent *if there exists a t-semiflow whose support covers $\mathcal{A}_{\text{label}}$.*

By Proposition 4.3.2, any PEPA model is conservative. Obviously, a p-semiflow is a special kind of p-flow while a t-semiflow is a special t-flow. Moreover, according to Lemma 4.3.1, a p-flow is an invariant.

Let $\mathbf{B}$ and $\mathbf{\Phi}$ be a basis of p-flows and a fundamental set of p-semiflows respectively. Then for any $\mathbf{m} \in \text{RS}(\mathcal{N}, \mathbf{m_0})$, we have $\mathbf{Bm} = 0$ and $\mathbf{\Phi m} = 0$. However this does not imply that any $\mathbf{m} \in \mathbb{N}^{|\mathcal{D}|}$ that satisfies $\mathbf{Bm} = 0$ or $\mathbf{\Phi m} = 0$ is in $\text{RS}(\mathcal{N}, \mathbf{m_0})$. But they do belong to generalised reachability sets. See the following definitions.

Definition 4.4.2. *(**Linearised Reachability Set**, [STC96]) Let $\mathcal{S}$ be a P/T system.*

1. *Its* linearised reachability set *using the* state equation *is defined as*

$$\text{LRS}^{\text{SE}}(\mathcal{S}) = \left\{ \mathbf{m} \in \mathbb{N}^{|P|} \mid \exists \boldsymbol{\sigma} \in \mathbb{N}^{|T|} \text{ such that } \mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \right\}.$$

2. *Its* linearised reachability set *using the* state equation over reals *is defined as*

$$\mathrm{LRS}^{\mathrm{SER}}(\mathcal{S}) = \left\{ \mathbf{m} \in \mathbb{N}^{|P|} \mid \exists \boldsymbol{\sigma} \geq \mathbf{0} \textit{ such that } \mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \right\}.$$

3. *Its* linearised reachability set *using a* basis B of p-flows *is defined as*

$$\mathrm{LRS}^{\mathrm{Pf}}(\mathcal{S}) = \left\{ \mathbf{m} \in \mathbb{N}^{|P|} \mid \mathbf{B} \cdot \mathbf{m} = \mathbf{B} \cdot \mathbf{m_0} \right\}.$$

4. *Its* linearised reachability set *using a* fundamental set of p-semiflows *is defined as*

$$\mathrm{LRS}^{\mathrm{Psf}}(\mathcal{S}) = \left\{ \mathbf{m} \in \mathbb{N}^{|P|} \mid \boldsymbol{\Phi} \cdot \mathbf{m} = \boldsymbol{\Phi} \cdot \mathbf{m_0} \right\}.$$

All the sets defined above are characterised using linear algebraic equations which makes the set structure simpler. The job of determining whether a state belongs to a set is reduced to verifying the equations.

Obviously, $\mathrm{RS}(\mathcal{S}) \subseteq \mathrm{LRS}^{\mathrm{SE}}(\mathcal{S})$. The difference between the definitions of $\mathrm{LRS}^{\mathrm{SE}}(\mathcal{S})$ and $\mathrm{LRS}^{\mathrm{SER}}(\mathcal{S})$ is embodied in the different conditions imposed on $\boldsymbol{\sigma}$. There is no doubt that $\mathrm{LRS}^{\mathrm{SE}}(\mathcal{S}) \subseteq \mathrm{LRS}^{\mathrm{SER}}(\mathcal{S})$. Since for any $\mathbf{m} \in \mathrm{LRS}^{\mathrm{SER}}(\mathcal{S})$, $\mathbf{m} = \mathbf{m}_0 + \mathbf{C} \cdot \boldsymbol{\sigma}$, then

$$\mathbf{Bm} = \mathbf{Bm}_0 + \mathbf{BC} \cdot \boldsymbol{\sigma} = \mathbf{Bm}_0,$$

so $\mathbf{m} \in \mathrm{LRS}^{\mathrm{Pf}}(\mathcal{S})$ and thus $\mathrm{LRS}^{\mathrm{SER}}(\mathcal{S}) \subseteq \mathrm{LRS}^{\mathrm{Pf}}(\mathcal{S})$. The definitions of $\mathrm{LRS}^{\mathrm{Pf}}(\mathcal{S})$ and $\mathrm{LRS}^{\mathrm{Psf}}(\mathcal{S})$ are directly related to the invariants in the system. Clearly, $\mathrm{LRS}^{\mathrm{Pf}}(\mathcal{S}) \subseteq \mathrm{LRS}^{\mathrm{Psf}}(\mathcal{S})$. The relationships between these reachability sets are shown in the following lemmas.

Lemma 4.4.1. *[STC96]. Let $\mathcal{S}$ be a P/T system, then*

1. $\mathrm{RS}(\mathcal{S}) \subseteq \mathrm{LRS}^{\mathrm{SE}}(\mathcal{S}) \subseteq \mathrm{LRS}^{\mathrm{SER}}(\mathcal{S}) \subseteq \mathrm{LRS}^{\mathrm{Pf}}(\mathcal{S}) \subseteq \mathrm{LRS}^{\mathrm{Psf}}(\mathcal{S})$.
2. *If $\mathcal{N}$ is conservative, then* $\mathrm{LRS}^{\mathrm{Pf}}(\mathcal{S}) = \mathrm{LRS}^{\mathrm{Psf}}(\mathcal{S})$.
3. *If $\mathcal{N}$ is consistent, then* $\mathrm{LRS}^{\mathrm{SER}}(\mathcal{S}) = \mathrm{LRS}^{\mathrm{Pf}}(\mathcal{S})$.

For the P/T structure $\mathcal{S}$ underlying a given PEPA model, we will see $\mathrm{LRS}^{\mathrm{SE}}(\mathcal{S}) = \mathrm{LRS}^{\mathrm{SER}}(\mathcal{S})$.

Lemma 4.4.2. *Let $\mathcal{S}$ be the P/T system underlying a PEPA model.*

$$\mathrm{LRS}^{\mathrm{SE}}(\mathcal{S}) = \mathrm{LRS}^{\mathrm{SER}}(\mathcal{S}).$$

Proof. For any $\mathbf{m} \in \text{LRS}^{\text{SER}}(\mathcal{S})$, there exists $\boldsymbol{\sigma} \geq \mathbf{0}$, such that $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma}$. Notice $\mathbf{m}, \mathbf{m_0} \in \mathbb{N}^{|\mathcal{D}|}$, all elements of $\boldsymbol{C}$ are either $-1, 0$ or 1, and the assumption that each column of $\mathbf{C}$ is distinct. So all elements of $\boldsymbol{\sigma}$ must be integers. Since $\boldsymbol{\sigma} \geq \mathbf{0}$, thus $\boldsymbol{\sigma} \in \mathbb{N}^{|\mathcal{A}_{\text{label}}|}$. That is $\mathbf{m} \in \text{LRS}^{\text{SE}}(\mathcal{S})$. So $\text{LRS}^{\text{SE}}(\mathcal{S}) \supset \text{LRS}^{\text{SER}}(\mathcal{S})$. Since $\text{LRS}^{\text{SE}}(\mathcal{S}) \subset \text{LRS}^{\text{SER}}(\mathcal{S})$, therefore $\text{LRS}^{\text{SE}}(\mathcal{S}) = \text{LRS}^{\text{SER}}(\mathcal{S})$. □

We should point out that this proof is based on the assumption of distinct columns of the activity matrix. However, this assumption can be relaxed for the analysis of state space. If there are two columns the same, i.e. the two corresponding labelled activities have the same pre and post local derivatives, we can modify the model in this way: combine these two labelled activities into one. The activity matrix of the modified model then satisfies the assumption. But the state space of the modified model as well as the associated transition relationship without timing information, is the same as the original one. So the structural analysis based on the new state space is the same as the analysis based on the original state space.

Now we introduce the concept of equal conflict (see [CTS98]).

Definition 4.4.3. *(**Equal Conflict**) Let $\mathcal{N} = (P, T, \mathbf{Pre}, \mathbf{Post})$ be a P/T net.*

1. *The P/T net $\mathcal{N}$ is called* equal conflict *(EQ), if $\text{pre}(l) \cap \text{pre}(l') \neq \emptyset$ implies $\mathbf{Pre}[\cdot, l] = \mathbf{Pre}[\cdot, l']$.*

2. *The P/T net $\mathcal{N}$ is* ordinary *if each entry of $\mathbf{Pre}$ and $\mathbf{Post}$ is either zero or one.*

3. *An ordinary EQ net is a* free choice *(FC) net.*

4. *A PEPA model is called an* EQ model *if the P/T net underlying this model is EQ.*

The following proposition will give an equivalent statement for an EQ PEPA model.

Proposition 4.4.1. *For a PEPA model, we have*

1. *A PEPA model is an EQ model if and only if for any two labelled activities l and l', their pre sets are either equal or distinct, i.e., either $\text{pre}(l) \cap \text{pre}(l') = \emptyset$ or $\text{pre}(l) = \text{pre}(l')$.*

2. *An EQ PEPA model is a FC model.*

Proof. Let us first prove term 1. Let l and l' be two arbitrary labelled activities. Suppose $\mathrm{pre}(l) \cap \mathrm{pre}(l') \neq \emptyset$. What we need to prove is :

$$\mathbf{C}^{\mathbf{Pre}}[\cdot, l] = \mathbf{C}^{\mathbf{Pre}}[\cdot, l'] \Longleftrightarrow \mathrm{pre}(l) = \mathrm{pre}(l').$$

It is easy to see that $\mathbf{C}^{\mathbf{Pre}}[\cdot, l] = \mathbf{C}^{\mathbf{Pre}}[\cdot, l']$ implies $\mathrm{pre}(l) = \mathrm{pre}(l')$, since each nonzero entry of a column of $\mathbf{C}^{\mathbf{Pre}}$ represents a pre local derivative of the corresponding activity. Now we prove "$\Longleftarrow$". Assume $\mathrm{pre}(l) = \mathrm{pre}(l') = \{U_1, U_2, \cdots, U_k\}$. By the definition of pre activity matrix, in the columns corresponding to l and l', all entries are set zeros except the entries corresponding to the local derivatives U_i $(i = 1, 2, \cdots, k)$ are set ones. So these two columns are the same, i.e. $\mathbf{C}^{\mathbf{Pre}}[\cdot, l] = \mathbf{C}^{\mathbf{Pre}}[\cdot, l']$.

Now let us to prove term 2. According to the definition of pre and post activity matrices, each element of them is either zero or one, that is, any PEPA model is ordinary. So an EQ PEPA model is a FC model. □

Remark 4.4.1. *For an arbitrary PEPA model, if the labelled activities l and l' are both individual, i.e. $\#pre(l) = \#pre(l') = 1$, it is easy to see that either $pre(l) \cap pre(l') = \emptyset$ or $pre(l) = pre(l')$. Suppose l is individual but l' shared, then $\#pre(l) = 1 < 2 \leq \#pre(l')$ and thus $pre(l) \neq pre(l')$. Therefore, as long as a local derivative can enable both an individual and a shared activities, then the PEPA model is not an EQ model. For example, notice in Model 3*

$$Y_1 \stackrel{\text{def}}{=} (action1, a1).Y_3 + (job1, c1).Y_2$$

where $action1$ is shared while $job1$ is individual, so Model 3 is not an EQ model.

Definition 4.4.4. *[CTS98]. A P/T system $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$ is* reversible[1] *if the starting state is reachable from any state in the reachability set, i.e., for any $\mathbf{m} \in \mathrm{RS}(\mathcal{S})$, there exists $\mathbf{m}' \in \mathrm{RS}(\mathcal{N}, \mathbf{m})$ such that $\mathbf{m}' = \mathbf{m}_0$.*

A reversible system means that the starting state is reachable from any state in the reachability set. A live, bounded, and reversible FC system has a good characteristic.

Lemma 4.4.3. *(page 225, [CTS98]) If $\mathcal{S}$ is a live, bounded, and reversible FC system, then $\mathrm{RS}(\mathcal{S}) = \mathrm{LRS}^{\mathrm{SE}}(\mathcal{S})$.*

[1] Here the definition of reversible follows the convention of [CTS98], which is different from the reversible definition in stochastic processes (see Section C.2 in Appendix C).

Based on the above lemmas, we show that for a class of P/T system, all the generalised sets are the same.

Theorem 4.4.4. *If $\mathcal{S}$ underlying a PEPA model is a live, reversible and EQ P/T system, then*

$$\mathrm{RS}(\mathcal{S}) = \mathrm{LRS}^{\mathrm{SE}}(\mathcal{S}) = \mathrm{LRS}^{\mathrm{SER}}(\mathcal{S}) = \mathrm{LRS}^{\mathrm{Pf}}(\mathcal{S}) = \mathrm{LRS}^{\mathrm{Psf}}(\mathcal{S}). \tag{4.11}$$

Proof. By Proposition 4.5.1 in Section 4.5, $\mathcal{S}$ is live implies that $\mathcal{S}$ is consistent. Since $\mathcal{S}$ is conservative by Proposition 4.3.2, then according to Lemma 4.4.1,

$$\mathrm{LRS}^{\mathrm{SER}}(\mathcal{S}) = \mathrm{LRS}^{\mathrm{Pf}}(\mathcal{S}) = \mathrm{LRS}^{\mathrm{Psf}}(\mathcal{S}).$$

By Lemma 4.4.2,

$$\mathrm{LRS}^{\mathrm{SE}}(\mathcal{S}) = \mathrm{LRS}^{\mathrm{SER}}(\mathcal{S}) = \mathrm{LRS}^{\mathrm{Pf}}(\mathcal{S}) = \mathrm{LRS}^{\mathrm{Psf}}(\mathcal{S}).$$

Since $\mathcal{S}$ is a bounded FC system according to Corollary 4.3.3 and Proposition 4.4.1, therefore by Lemma 4.4.3,

$$\mathrm{RS}(\mathcal{S}) = \mathrm{LRS}^{\mathrm{SE}}(\mathcal{S}) = \mathrm{LRS}^{\mathrm{SER}}(\mathcal{S}) = \mathrm{LRS}^{\mathrm{Pf}}(\mathcal{S}) = \mathrm{LRS}^{\mathrm{Psf}}(\mathcal{S}).$$

□

For live, reversible and EQ PEPA models, because all the states can be described by a matrix equation, the state space derivation is always available and easy. The storage memory can be significantly reduced since what needs to be stored is an equation rather than its solutions. The validation of a state, i.e. judging a vector belongs to the state space, is reduced to checking whether this vector satisfies the matrix equation and thus avoids searching the entire state space.

4.4.2 Example

Let us see an example. Recall Model 1. Suppose the starting state is $\mathbf{m_0} = (M, 0, N, 0)^T$. So this is a live, reversible and EQ PEPA model. According to the operational semantics of PEPA, it is easy to determine the state space:

$$\mathrm{RS}(\mathcal{S}) = \{(x_1, M - x_1, y_1, N - y_1)^T \mid x_1, y_1 \in \mathbb{N}, 0 \leq x_1 \leq M, 0 \leq y_1 \leq N)\}.$$

There are $(M+1)\times(N+1)$ states in the state space $\mathrm{RS}(\mathcal{S})$.

Now we determine $\mathrm{LRS}^{\mathrm{Psf}}(\mathcal{S})$ based only on the activity matrix and the starting state. The activity matrix of Model 1 (that was already shown in Table 4.1) is

$$\mathbf{C} = \begin{pmatrix} -1 & 1 & 0 \\ 1 & -1 & 0 \\ -1 & 0 & 1 \\ 1 & 0 & -1 \end{pmatrix}. \tag{4.12}$$

Solve $\mathbf{C}^T\mathbf{y} = 0$, we get a basis of the solution space, which forms the rows of the fundamental set Φ:

$$\mathbf{\Phi} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}. \tag{4.13}$$

Notice that

$$\begin{aligned}
\mathrm{LRS}^{\mathrm{Psf}}(\mathcal{S}) &= \{\mathbf{m} \in \mathbb{N}^4 \mid \Phi\mathbf{m} = \mathbf{\Phi}\mathbf{m_0}\} \\
&= \left\{ \mathbf{m} \in \mathbb{N}^4 \;\middle|\; \begin{array}{c} \mathbf{m}[x_1] + \mathbf{m}[x_2] = \mathbf{m_0}[x_1] + \mathbf{m_0}[x_2] \\ \mathbf{m}[y_1] + \mathbf{m}[y_2] = \mathbf{m_0}[y_1] + \mathbf{m_0}[y_2] \end{array} \right\} \\
&= \left\{\mathbf{m} \in \mathbb{N}^4 \mid \mathbf{m}[x_1] + \mathbf{m}[x_2] = M; \mathbf{m}[y_1] + \mathbf{m}[y_2] = N\right\} \\
&= \left\{(x_1, M - x_1, y_1, N - y_1)^T \mid x_1, y_1 \in \mathbb{N}, 0 \le x_1 \le M, 0 \le y_1 \le N\right\} \\
&= \mathrm{RS}(\mathcal{S}).
\end{aligned}$$

By Proposition 4.4.1, we have

$$\mathrm{RS}(\mathcal{S}) = \mathrm{LRS}^{\mathrm{SE}}(\mathcal{S}) = \mathrm{LRS}^{\mathrm{SER}}(\mathcal{S}) = \mathrm{LRS}^{\mathrm{Pf}}(\mathcal{S}) = \mathrm{LRS}^{\mathrm{Psf}}(\mathcal{S}).$$

This is consistent with Theorem 4.4.4.

4.5 Improved Deadlock-Checking Methods for PEPA

A deadlock in a set characterises the existence of a state in this set, from which no transition can be enabled. Deadlock-checking is an important topic in qualitative analysis of computer and communication systems. It has popular applications, in particular in protocol validation.

The current deadlock-checking algorithm for PEPA relies on exploring the entire state space to find whether a deadlock exists. For large scale PEPA models, deadlock-checking can become impossible since this kind of algorithm suffers from the state-space explosion problem. This section will present an efficient deadlock-checking method which does not heavily depend on the size of the state space.

4.5.1 Preliminary

We first introduce the definitions:

Definition 4.5.1. ***(Deadlock-free for PEPA)*** *Let the P/T structure underlying any given PEPA model be* $\langle \mathcal{N}, \mathbf{m_0} \rangle$.

1. *A* deadlock *of the model or the underlying P/T structure is a state in the state space which cannot enable any transition, i.e. cannot fire any activity.*
2. *The model or the structure* $\langle \mathcal{N}, \mathbf{m_0} \rangle$ *is* deadlock-free *if it has no deadlock.*

We should point out that, as shown in Table 4.3, the state space of a PEPA model is the same as the reachability set of the P/T structure underlying this model. So it is equivalent to give the deadlock definition in the context of the model or the corresponding P/T structure.

When the activity l is disabled in state $\mathbf{m}$ it can be expressed as:

$$\mathbf{m} \ngeq \mathbf{C}^{\mathbf{Pre}}[\cdot, l] \tag{4.14}$$

or

$$\bigvee_{U \in \text{pre}(l)} \mathbf{m}[U] < \mathbf{C}^{\mathbf{Pre}}[U, l]. \tag{4.15}$$

If (4.14) or (4.15) holds for all activities in $\mathcal{A}_{\text{label}}$, i.e., there is no activity that can be enabled at $\mathbf{m}$, then $\mathbf{m}$ is a deadlock. The following Theorem 4.5.1 gives a mathematical statement for deadlock-free models.

Theorem 4.5.1. *(Theorem 30, [STC96]). Let* $\mathcal{S}$ *be a P/T system. If there is no (integer) solution to*

$$\begin{cases} \mathbf{m} - \mathbf{C}\boldsymbol{\sigma} = \mathbf{m}_0, \\ \mathbf{m}, \boldsymbol{\sigma} \geq 0, \\ \bigvee_{U \in \text{pre}(l)} \mathbf{m}[U] < \mathbf{C}^{\mathbf{Pre}}[U, l] \quad \forall l \in \mathcal{A}_{\text{label}}, \end{cases}$$

then $\mathcal{S}$ is deadlock-free.

According to this theorem, to decide whether there is a deadlock, it is necessary to compare each state in the state space to each column of the pre activity matrix. This is not an efficient way to find a deadlock, especially for large scale PEPA models.

$\bigvee_{U\in \text{pre}(l)} \mathbf{m}[U] < \mathbf{C}^{\mathbf{Pre}}[U,l]$ means that there exists U such that $\mathbf{m}[U] < \mathbf{C}^{\mathbf{Pre}}[U,l]$. Notice that all elements of $\mathbf{m}$ are nonnegative integers, and any entry of $\mathbf{C}^{\mathbf{Pre}}[U,l]$ other than 1 is zero for all U and l. So

$$\mathbf{m}[U] < \mathbf{C}^{\mathbf{Pre}}[U,l] \Longleftrightarrow \mathbf{m}[U] = 0 \text{ and } \mathbf{C}^{\mathbf{Pre}}[U,l] = 1.$$

Of course, $\mathbf{m} \geq \mathbf{1}$ cannot be a deadlock, since $\mathbf{m} \geq \mathbf{C}^{\mathbf{Pre}}[\cdot,l]$ for any l. Thus, only a state $\mathbf{m}$ with zeros in some particular places can possibly be a deadlock. This observation is very helpful but not enough for deadlock-checking. All such states with zeros in some entries, have to be found for checking, which is not always feasible especially in the situation of the state space derivation or storage being a problem. Theorem 4.4.4 specifies the structure of the state space, but it requires the condition of "liveness" in advance, which has already guaranteed the deadlock-freeness.

In the following first subsection, an equivalent deadlock-checking theorem for a class of PEPA models is illustrated, which allows equivalent deadlock-checking in the linearised state space. The second subsection illustrates an efficient checking algorithm with some examples.

4.5.2 Equivalent deadlock-checking

Before stating our main results, we first list several lemmas which are used in the proof of our theorem.

The reachability set of a P/T net is in fact a directed graph, i.e., each state is a node, the transition from a state to another state is essentially a directed edge between two nodes. A directed graph is called *strongly connected* if it contains a directed path from u to v and a directed path from v to u for every pair of vertices u, v. A graph is called *connected* if every pair of distinct vertices in the graph can be connected through some path. Obviously, a strongly connected graph is a connected graph. The two definitions have been introduced to nets (see [MR80]). For a net, the

following lemma provides two sufficient conditions of strongly connected.

Lemma 4.5.2. *(Property 6.10, page 217, [CTS98]) Let $\mathcal{N}$ be a graph and $\mathbf{C}$ its incidence matrix.*

1. *If $\mathcal{N}$ is connected, consistent and conservative, then it is strongly connected.*
2. *If $\mathcal{N}$ is live and bounded then it is strongly connected and consistent.*

If $\mathcal{N}$ is the P/T net underlying a PEPA model, these conditions can be simplified.

Proposition 4.5.1. *Suppose $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ be a P/T system underlying a PEPA model.*

1. *If $\mathcal{N}$ is consistent, then the state space is strongly connected.*
2. *If S is live, then $\mathcal{N}$ is strongly connected and consistent.*

Proof. By Proposition 4.3.2 and Corollary 4.3.3 the P/T systems underlying PEPA models are conservative and bounded. Notice that each state in the state space is reachable from the initial state, i.e. the state space is connected. So according to Lemma 4.5.2, Proposition 4.5.1 holds. □

Lemma 4.5.3. *(Theorem 6.19, page 223, [CTS98]) Let $\mathcal{S}$ be a bounded strongly connected EQ system. Then $\mathcal{S}$ is live iff it is deadlock-free.*

Lemma 4.5.4. *(Theorem 6.22, page 225, [CTS98]) If $\mathcal{S}$ is a live EQ system, then for any $\boldsymbol{m_a}, \boldsymbol{m_b} \in \mathrm{LRS}^{\mathrm{SE}}(\mathcal{S})$, $\mathrm{RS}(\mathcal{N}, \boldsymbol{m_a}) \cap \mathrm{RS}(\mathcal{N}, \boldsymbol{m_b}) \neq \emptyset$.*

This lemma implies that there are no spurious deadlocks in live EQ systems, i.e. there are no deadlocks in $\mathrm{LRS}^{\mathrm{SE}}(\mathcal{S})$.

Theorem 4.5.5. *If the P/T structure $\mathcal{S}$ underlying a PEPA model is a consistent, EQ system, then*

1. $\mathrm{LRS}^{\mathrm{SE}}(\mathcal{S})$ *is deadlock-free* $\Longleftrightarrow$ $\mathrm{RS}(\mathcal{S})$ *is deadlock-free.*
2. $\mathrm{LRS}^{\mathrm{SER}}(\mathcal{S})$ *is deadlock-free* $\Longleftrightarrow$ $\mathrm{RS}(\mathcal{S})$ *is deadlock-free.*
3. $\mathrm{LRS}^{\mathrm{Pf}}(\mathcal{S})$ *is deadlock-free* $\Longleftrightarrow$ $\mathrm{RS}(\mathcal{S})$ *is deadlock-free.*

4. $\text{LRS}^{\text{Psf}}(\mathcal{S})$ *is deadlock-free* $\Longleftrightarrow$ $\text{RS}(\mathcal{S})$ *is deadlock-free.*

Proof. It is easy to see that all "$\Longrightarrow$" holds because

$$\text{LRS}^{\text{Psf}}(\mathcal{S}) \supset \text{LRS}^{\text{Pf}}(\mathcal{S}) \supset \text{LRS}^{\text{SER}}(\mathcal{S}) \supset \text{LRS}^{\text{SE}}(\mathcal{S}) \supset \text{RS}(\mathcal{S}).$$

Now we show that each "$\Longleftarrow$" holds. Notice $\mathcal{S}$ is consistent, by Proposition 4.5.1, $\mathcal{S}$ is strongly connected. Then according to Lemma 4.5.3, $\text{RS}(\mathcal{S})$ is deadlock-free implies that $\mathcal{S}$ is live. Since $\mathcal{S}$ is an EQ system, then by Lemma 4.5.4, $\text{LRS}^{\text{SE}}(\mathcal{S})$ is deadlock-free. Since now we have $\text{LRS}^{\text{SE}}(\mathcal{S}) = \text{LRS}^{\text{SER}}(\mathcal{S})$ by Lemma 4.4.2, so $\text{LRS}^{\text{SER}}(\mathcal{S})$ is deadlock-free. Notice by Lemma 4.4.1, the conservativeness and consistence of the system imply that $\text{LRS}^{\text{Pf}}(\mathcal{S}) = \text{LRS}^{\text{Psf}}(\mathcal{S}) = \text{LRS}^{\text{SER}}(\mathcal{S})$. So $\text{LRS}^{\text{Pf}}(\mathcal{S})$ and $\text{LRS}^{\text{Psf}}(\mathcal{S})$ are deadlock-free. □

Theorem 4.5.5 allows us to check the corresponding linearised state space to determine whether a consistent and EQ model has deadlocks. We should point out that "consistent" and "EQ" can be efficiently checked as properties of the activity matrix.

4.5.3 Deadlock-checking algorithm in LRS^{Psf}

According to Theorem 4.5.5, for a consistent, EQ system $\mathcal{S}$, to tell whether $\text{RS}(\mathcal{S})$ has deadlocks it is sufficient to check whether $\text{LRS}^{\text{Psf}}(\mathcal{S})$ has deadlocks.

As we mentioned, the activity l is disabled in $\mathbf{m}$ means that there exists a U such that $\mathbf{m}[U] < \mathbf{C}^{\mathbf{Pre}}[U, l]$. Because

$$\mathbf{m}[U] < \mathbf{C}^{\mathbf{Pre}}[U, l] \Longleftrightarrow \mathbf{m}[U] = 0 \text{ and } \mathbf{C}^{\mathbf{Pre}}[U, l] = 1,$$

so only the state $\mathbf{m}$ with zeros in some particular places can possibly be a deadlock. Based on this idea, we provide a deadlock-checking algorithm, see Algorithm 2. In this algorithm, $K(l)$ is the set of vectors that cannot enable l. The intersected set of all $K(l)$, i.e. $K = \bigcap_{l \in \mathcal{A}_{\text{label}}} K(l)$, is the deadlock candidate set, in which each vector cannot fire any activity. $K \bigcap \text{LRS}^{\text{Psf}}$ is used to check whether the deadlock candidates are in the linearised state space LRS^{Psf}.

Since this algorithm depends on the system structure rather than the repeat instances of the components, so does its computational complexity. Therefore, it is efficient for large scale systems with repeated components. Our deadlock-checking algorithm is structure- or equation-based,

Algorithm 2 Deadlock-checking in LRS^{Psf}

1: **for all** $l \in \mathcal{A}_{\text{label}}$ **do**
2: **if** l is an individual activity **then**
3: $K(l) = \{\mathbf{m} \in \mathbb{N}^{|\mathcal{D}|} \mid \mathbf{m}[U] = 0, \mathbf{C}^{\mathbf{Pre}}[U, l] = 1\}$ // where $\{U\} = \text{pre}(l)$
4: **else if** l is a shared activity **then**
5: $K(l) = \bigcup_{U \in \text{pre}(l)} \{\mathbf{m} \in \mathbb{N}^{|\mathcal{D}|} \mid \mathbf{m}[U] = 0, \mathbf{C}^{\mathbf{Pre}}[U, l] = 1\}$
6: **end if**
7: **end for**
8: $K = \bigcap_{l \in \mathcal{A}_{\text{label}}} K(l)$
9: If $K \bigcap \text{LRS}^{\text{Psf}} = \emptyset$, then LRS^{Psf} is deadlock-free. Otherwise, LRS^{Psf} at least has one deadlock.

rather than state-space-based, so it avoids searching the entire state space and thus avoids the state-space explosion problem. Although Theorem 4.5.5 requires the conditions of consistent and EQ, Algorithm 2 is free from these restrictions since it deals with the linearised state space. That means, for any general PEPA model with or without the consistency and EQ restrictions, if the generalised state space has no deadlocks reported by using Algorithm 2, then the model has no deadlocks. But if it reports deadlocks in the generalised state space, it cannot tell whether there is a deadlock in the model, except for a consistent and EQ model.

We should point out that each entry of each numerical state (regardless of whether it is in the state space or the linearised state space) is an integer bounded between zero and the population of the corresponding component type. So all the sets appearing in this algorithm are finite. Thus, this algorithm is computable. However we have not implemented this algorithm. The weakness of this approach is that if the populations of the entities are not specified then symbolic computation is needed. But at this cost, a non-negligible advantage has been obtained: this method can tell when or how a system structure may lead to deadlocks. The next subsection will demonstrate the application of Algorithm 2 to some small examples.

4.5.4 Examples

This section presents two examples to illustrate how to use Algorithm 2 to check deadlocks for PEPA models.

4.5.4.1 Example 1: always deadlock-free

Recall Model 1,

$$\begin{aligned} User_1 &\stackrel{def}{=} (task_1, a).User_2 \\ User_2 &\stackrel{def}{=} (task_2, b).User_1 \\ Provider_1 &\stackrel{def}{=} (task_1, a).Provider_2 \\ Provider_2 &\stackrel{def}{=} (reset, d).Provider_1 \\ User_1[M] &\underset{\{task_1\}}{\bowtie} Provider_1[N]. \end{aligned}$$

The activity matrix $\mathbf{C}$ and pre activity $\mathbf{C^{Pre}}$ of Model 1 are

$$\mathbf{C} = \begin{pmatrix} -1 & 1 & 0 \\ 1 & -1 & 0 \\ -1 & 0 & 1 \\ 1 & 0 & -1 \end{pmatrix}, \quad \mathbf{C^{Pre}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

According to Algorithm 2,

$$K(task1) = \{\mathbf{m} \mid \mathbf{m}[User_1] = 0 \text{ or } \mathbf{m}[Provider_1] = 0\},$$

$$K(task2) = \{\mathbf{m} \mid \mathbf{m}[User_2] = 0\},$$

$$K(reset) = \{\mathbf{m} \mid \mathbf{m}[Provider_2] = 0\},$$

So

$$\begin{aligned} K &= K(task1) \cap K(task2) \cap K(rest) \\ &= \{\mathbf{m} \mid \mathbf{m}[User_1] = 0, \mathbf{m}[User_2] = 0, \mathbf{m}[Provider_2] = 0\} \\ &\cup \{\mathbf{m} \mid \mathbf{m}[Provider_1] = 0, \mathbf{m}[User_2] = 0, \mathbf{m}[Provider_2] = 0\}. \end{aligned}$$

We have determined the $\text{LRS}^{\text{Psf}}(\mathcal{S})$ in Section 4.4.2:

$$\begin{aligned} \text{LRS}^{\text{Psf}}(\mathcal{S}) &= \{\mathbf{m} \in \mathbb{N}^4 \mid \Phi\mathbf{m} = \mathbf{\Phi m_0}\} \\ &= \{(x_1, M - x_1, y_1, N - y_1)^T \mid x_1, y_1 \in \mathbb{N}, 0 \leq x_1 \leq M, 0 \leq y_1 \leq N\}. \end{aligned}$$

So $K \cap \text{LRS}^{\text{Psf}} = \emptyset$. That is to say, the system has no deadlocks.

4.5.4.2 Example 2: deadlocks in some situations

Now we consider Model 4, which has a consistent and EQ P/T structure.

Model 4.

$$User_1 \stackrel{\text{def}}{=} (task_1, 1).User_2$$

$$User_2 \stackrel{\text{def}}{=} (task_2, 1).User_1$$

$$Provider_1 \stackrel{\text{def}}{=} (task_1, 1).Provider_2$$

$$Provider_2 \stackrel{\text{def}}{=} (task_2, 1).Provider_1$$

$$(User_1[M_1] \parallel User_2[M_2]) \underset{\{task_1, task_2\}}{\bowtie} (Provider_1[N_1] \parallel Provider_2[N_2]).$$

	$task_1$	$task_2$	
$User_1$	-1	1	
$User_2$	1	-1	
$Provider_1$	-1	1	
$Provider_2$	1	-1	

Table 4.6: *Activity matrix and pre activity matrix of Model 4*

Table 4.6 lists the activity matrix of Model 4. The activity matrix $\mathbf{C}$ and pre activity matrix $\mathbf{C^{Pre}}$ are listed below:

$$\mathbf{C} = \begin{pmatrix} -1 & 1 \\ 1 & -1 \\ -1 & 1 \\ 1 & -1 \end{pmatrix}, \quad \mathbf{C^{Pre}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

First, let us determine $\text{LRS}^{\text{Psf}}(\mathcal{S})$. Solving $\mathbf{C}^T\mathbf{y} = 0$, we get a basis of the solution space which forms the rows of Φ:

$$\mathbf{\Phi} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

Notice $\mathbf{m_0} = (M_1, M_2, N_1, N_2)^T$, so

$$\begin{aligned}\text{LRS}^{\text{Psf}}(\mathcal{S}) &= \{\mathbf{m} \in \mathbb{N}^4 \mid \Phi\mathbf{m} = \mathbf{\Phi m_0}\} \\ &= \left\{ \mathbf{m} \in \mathbb{N}^4 \;\middle|\; \begin{array}{c} \mathbf{m}[User_1] + \mathbf{m}[User_2] = M_1 + M_2; \\ \mathbf{m}[User_2] + \mathbf{m}[Provider_1] = M_2 + N_1; \\ \mathbf{m}[Provider_1] + \mathbf{m}[Provider_2] = N_1 + N_2 \end{array} \right\}\end{aligned}$$

Note that each of the semiflows corresponds to an invariant of the model. The first and third express the fact that the number of users, and the number of providers respectively, is constant within the model. The second expresses the coupling between the components, i.e. the co-operations ensure that the numbers of local derivatives in the two components always change together.

Secondly, we determine the potential deadlock set K. According to Algorithm 2,

$$K(task1) = \{\mathbf{m} \mid \mathbf{m}[User_1] = 0 \text{ or } \mathbf{m}[Provider_1] = 0\},$$

$$K(task2) = \{\mathbf{m} \mid \mathbf{m}[User_2] = 0 \text{ or } \mathbf{m}[Provider_2] = 0\},$$

$$\begin{aligned}K =& K(task1) \cap K(task2) \\ =& \{\mathbf{m} \mid \mathbf{m}[User_1] = 0, \mathbf{m}[User_2] = 0\} \cup \{\mathbf{m} \mid \mathbf{m}[User_1] = 0, \mathbf{m}[Provider_2] = 0\} \\ & \cup \{\mathbf{m} \mid \mathbf{m}[Provider_1] = 0, \mathbf{m}[User_2] = 0\} \\ & \cup \{\mathbf{m} \mid \mathbf{m}[Provider_1] = 0, \mathbf{m}[Provider_2] = 0\}\end{aligned}$$

Finally, the deadlock set in LRS^{Psf} is

$$
\begin{aligned}
& K \cap \text{LRS}^{\text{Psf}} \\
= & \left\{ \mathbf{m} \in \mathbb{N}^4 \;\middle|\; \begin{array}{c} \mathbf{m}[User_1] + \mathbf{m}[User_2] = M_1 + M_2; \\ \mathbf{m}[User_2] + \mathbf{m}[Provider_1] = M_2 + N_1; \\ \mathbf{m}[Provider_1] + \mathbf{m}[Provider_2] = N_1 + N_2 \end{array} \right\} \\
& \bigcap \{\mathbf{m} \mid (\mathbf{m}[User_1] = \mathbf{m}[Provider_2] = 0) \vee (\mathbf{m}[Provider_1] = \mathbf{m}[User_2] = 0)\} \\
= & \left\{ \mathbf{m} \in \mathbb{N}^4 \mid \begin{array}{c} (\mathbf{m} = (0, M_1 + M_2, N_1 + N_2, 0)^T \wedge M_1 + N_2 = 0) \\ \vee\, (\mathbf{m} = (M_1 + M_2, 0, 0, N_1 + N_2)^T \wedge M_2 + N_1 = 0) \end{array} \right\} \\
= & \left\{ \mathbf{m} \in \mathbb{N}^4 \mid \begin{array}{c} (\mathbf{m} = (0, M_1 + M_2, N_1 + N_2, 0)^T \wedge M_1 = N_2 = 0) \\ \vee\, (\mathbf{m} = (M_1 + M_2, 0, 0, N_1 + N_2)^T \wedge M_2 = N_1 = 0) \end{array} \right\}.
\end{aligned}
$$

In other words, for Model 4 with $\mathbf{m_0} = (M_1, M_2, N_1, N_2)^T$, only when $M_1 = N_2 = 0$ or $M_2 = N_1 = 0$, $K \cap \text{LRS}^{\text{Psf}} \neq \emptyset$, i.e. the system has at least one deadlock. Otherwise, the system is deadlock-free as long as $M_1 + N_2 \neq 0$ and $M_2 + N_1 \neq 0$.

This example illustrates that our deadlock-checking method can not only tell whether a particular system is deadlock-free but also how a system structure may lead to deadlocks.

4.6 Summary

This chapter has revealed the P/T structure underlying PEPA models. Based on the techniques developed for P/T systems, we have solved the derivation and storage problems of state space for a class of large scale PEPA models. For any general PEPA models, we demonstrated how to find their invariants. These invariants can be used to reason about systems in practise, and used to prove convergence results in the theoretical development for PEPA (see Chapter 6). Our main contribution in this chapter, is the structure-based deadlock-checking method for PEPA. This method can efficiently reduce the computational complexity of deadlock-checking and avoid the state-space explosion problem. The philosophy behind our approach, i.e. structure-based and equation-based considerations, can be applied to other problems such as logical model-checking.

Chapter 5
Fluid Analysis for Large Scale PEPA Models—Part I: Probabilistic Approach

5.1 Introduction

In the previous chapter we have discussed the techniques of structural analysis for large scale PEPA models, to deal with the state-space explosion problem. Quantitative analysis of large scale PEPA models also suffers from this problem, which is encountered in the calculation of steady-state probability distributions of the CTMCs. The state-space explosion problem is inherent in the discrete state approach employed in stochastic process algebras and many other formal modelling approaches. Recently, for the stochastic process algebra PEPA [Hil96], Hillston has developed a novel approach—continuous state-space approximation—to avoid this problem [Hil05a]. This approach results in a set of ODEs, leading to the evaluation of transient and, in the limit, steady state measures.

More recently, an extension of the previous mapping from PEPA to ODEs, relaxing some structure restrictions, has been presented in [BGH07]. In particular, synchronisations are allowed between active and passive actions. The relationship between the derived ODEs and the CTMCs for a special example in the context of PEPA, was revealed in [GHS08]: the set of ODEs automatically extracted from the PEPA description are the limits of the sequence of underlying CTMCs. However, for general applications the structure restrictions in both [BGH07] and [GHS08] should be further relaxed. For example, an individual activity may occur more than once within derivative definitions and can appear within different component definitions. Moreover, shared activities may have different local rates in realistic scenarios.

In this chapter, we relax these conditions and extend the mapping semantics, by employing the activity matrices and transition rate functions introduced in Chapter 3. Moreover, we will establish some fundamental characteristics of the solutions of the derived ODEs, such as the

existence, uniqueness and convergence. For PEPA models without synchronisations, we will show that the solutions of the derived ODEs have finite limits and the limits coincide with the steady-state probability distributions of the underlying CTMCs. For general PEPA models with synchronisations, based on the pioneering work in [GHS08], the consistency between the derived ODEs and a family of underlying density dependent CTMCs has been demonstrated. Furthermore, we will show the convergence of the solutions of the ODEs generated from general PEPA models under a particular condition.

We assume in this chapter that the CTMCs underlying considered PEPA models are irreducible and positive-recurrent [And91]. By Theorem 3.5.3 in [Hil96], the CTMC underlying a PEPA model is irreducible if and only if the initial component of the model is cyclic. According to the conclusion (Proposition 1.7, page 163, [And91]) that an irreducible Markov chain is positive-recurrent if and only if there exists a steady-state distribution, it is natural to assume the underlying CTMCs to be positive-recurrent because some discussions in the following are based on the existence of steady-state distributions. Under the assumption of these two conditions, i.e. irreducible and positive-recurrent, the CTMCs underlying considered PEPA models have steady-state probability distributions, which has also been illustrated in Theorem 3.5.2 in [Hil96].

The remainder of this chapter is structured as follows. Section 2 describes the fluid approximations of general PEPA models, as well as the existence and uniqueness theorem for the derived ODEs, while the convergence of the solutions in the nonsynchronisation case is presented in Section 3. Section 4 presents the concept of density dependent CTMCs and the relationship between this concept and the derived ODEs, based on which the convergence of the ODE solutions under a particular condition for general PEPA models will be shown in Section 5. Section 6 presents the further investigation of this particular condition. Finally, we conclude the chapter in Section 7.

5.2 Fluid Approximations for PEPA Models

The section will introduce the fluid-flow approximations for PEPA models, which leads to some kind of nonlinear ODEs. The existence and uniqueness of the solutions of the ODEs will be established. Moreover, a conservation law satisfied by the ODEs will be shown.

5.2.1 Deriving ODEs from PEPA models

Chapter 3 introduces a numerical vector form to capture the state information of models with repeated components. In this vector form there is one entry for each local derivative of each component type in the model. The entries in the vector are no longer syntactic terms representing the local derivative of the sequential component, but the number of components currently exhibiting this local derivative. Each numerical vector represents a single state of the system. The rates of the transitions between states are specified by the transition rate functions defined in Chapter 3. For example, the transition from state $\mathbf{x}$ to $\mathbf{x}+l$ can be written as

$$\mathbf{x} \xrightarrow{(l,f(\mathbf{x},l))} \mathbf{x}+l,$$

where l is a transition vector corresponding to the labelled activity l (for convenience, hereafter each pair of transition vectors and corresponding labelled activities shares the same notation), and $f(\mathbf{x},l)$ is the transition rate function, reflecting the intensity of the transition from $\mathbf{x}$ to $\mathbf{x}+l$.

The state space is inherently discrete with the entries within the numerical vector form always being non-negative integers and always being incremented or decremented in steps of one. As pointed out in [Hil05a], when the numbers of components are large these steps are relatively small and we can approximate the behaviour by considering the movement between states to be continuous, rather than occurring in discontinuous jumps. In fact, let us consider the evolution of the numerical state vector. Denote the state at time t by $\mathbf{x}(t)$. In a short time Δt, the change to the vector $\mathbf{x}(t)$ will be

$$\mathbf{x}(\cdot,t+\Delta t)-\mathbf{x}(\cdot,t)=F(\mathbf{x}(\cdot,t))\Delta t=\Delta t\sum_{l\in\mathcal{A}_{\text{label}}} lf(\mathbf{x}(\cdot,t),l).$$

Dividing by Δt and taking the limit, $\Delta t\to 0$, we obtain a set of ordinary differential equations (ODEs):

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}=F(\mathbf{x}), \tag{5.1}$$

where

$$F(\mathbf{x})=\sum_{l\in\mathcal{A}_{\text{label}}} lf(\mathbf{x},l). \tag{5.2}$$

Once the activity matrix and the transition rate functions are generated, the ODEs are immedi-

ately available. All of them can be obtained automatically by Algorithm 1 in Chapter 3.

Let U be a local derivative. For any transition vector l, $l[U]$ is either ± 1 or 0. If $l[U] = -1$ then U is in the pre set of l, i.e. $U \in \mathrm{pre}(l)$, while $l[U] = 1$ implies $U \in \mathrm{post}(l)$. According to (5.1) and (5.2),

$$\begin{aligned}\frac{\mathrm{d}\mathbf{x}(U,t)}{\mathrm{d}t} &= \sum_{l} l[U] f(\mathbf{x}, l) \\ &= -\sum_{l:l[U]=-1} f(\mathbf{x}, l) + \sum_{l:l[U]=1} f(\mathbf{x}, l) \\ &= -\sum_{\{l|U\in \mathrm{pre}(l)\}} f(\mathbf{x}, l) + \sum_{\{l|U\in \mathrm{post}(l)\}} f(\mathbf{x}, l).\end{aligned} \tag{5.3}$$

The term $\sum_{\{l|U\in \mathrm{pre}(l)\}} f(\mathbf{x}, l)$ represents the "exit rates" in the local derivative U, while the term $\sum_{\{l|U\in \mathrm{post}(l)\}} f(\mathbf{x}, l)$ reflects the "entry rates" in U. The formulae (5.1) and (5.2) are activity centric while (5.3) is local derivative centric. Our approach has extended previous results presented in the literature, see Table 5.1.

No.	Restrictions	paper [Hil05a]	paper [BGH07]	paper [GHS08]	this thesis
1	The cooperation set between interacting groups of components is not restricted to be the set of common action labels between these groups of components.		$\surd$		$\surd$
2	Shared activities may have different local rates.				$\surd$
3	Allow passive rate		$\surd$		$\surd$
4	Each action name may appear in different local derivatives within the definition of a sequential component, and may occur multiple times with that derivative definition.				$\surd$
5	Action hiding is considered.				$\surd^a$

[a] Action hiding is not discussed in this thesis, but can be employed based on our scheme. In our scheme each unknown action τ can be distinguished since they have distinct attached labels.

Table 5.1: *Comparison with respect to restrictions*

For an arbitrary CTMC, there are backward and forward equations describing the evolution of the transition probabilities. From these equations the evolution of probabilities distributed on each state can be easily induced ([BGdMT98], page 52). For example, for the (aggregated)

CTMC underlying a PEPA model, the corresponding differential equations describing the evolution of the probability distributions are

$$\frac{\mathrm{d}\pi}{\mathrm{d}t} = Q^T\pi, \tag{5.4}$$

where each entry of $\pi(t)$ represents the probability of the system being in each state at time t, and Q is an infinitesimal generator matrix corresponding to the CTMC. Obviously, the dimension of the coefficient matrix Q is the square of the size of the state space, which increases as the number of components increases.

The derived ODEs (5.1) describe the evolution of the population of the components in *each local derivative*, while (5.4) reflects the the probability evolution at *each state*. Since the scale of (5.1), i.e. the number of the ODEs, is only determined by the number of local derivatives and is unaffected by the size of the state space, so it avoids the state-space explosion problem. But the scale of (5.4) depends on the size of the state space, so it suffers from the explosion problem. The price paid is that the ODEs (5.1) are generally nonlinear due to synchronisations, while (5.4) is linear. However, if there is no synchronisation contained then (5.1) becomes linear, and there is some correspondence and consistency between these two different types of ODEs, which will be demonstrated in Section 5.3.

It is well known that for an irreducible and positive-recurrent CTMC, the solution of the corresponding ODEs (5.4) has a unique limit, which is the unique steady-state probability distribution of the given CTMC. From this distribution, the performance measures such as throughput and utilisation can be derived. Analogously, we want to know whether the solution of (5.1) has a finite limit, from which similar performance measures can also be derived. If the limit exists, what is the relationship between the limit and the steady-state probability distribution of the underlying CTMC? These problems are the main topics of this chapter.

5.2.2 Example

Now we show an example. Recall Model 2 presented in Chapter 3:

$$P_1 \stackrel{def}{=} (\alpha, r'_\alpha).P_2 + (\alpha, r''_\alpha).P_3$$
$$P_2 \stackrel{def}{=} (\beta, r_\beta).P_1 + (\beta, r'_\beta).P_3$$
$$P_3 \stackrel{def}{=} (\gamma, r_\gamma).P_1$$
$$Q_1 \stackrel{def}{=} (\alpha, r_\alpha).Q_2$$
$$Q_2 \stackrel{def}{=} (\gamma, r'_\gamma).Q_1$$
$$P_1[A] \underset{\{\alpha\}}{\bowtie} Q_1[B].$$

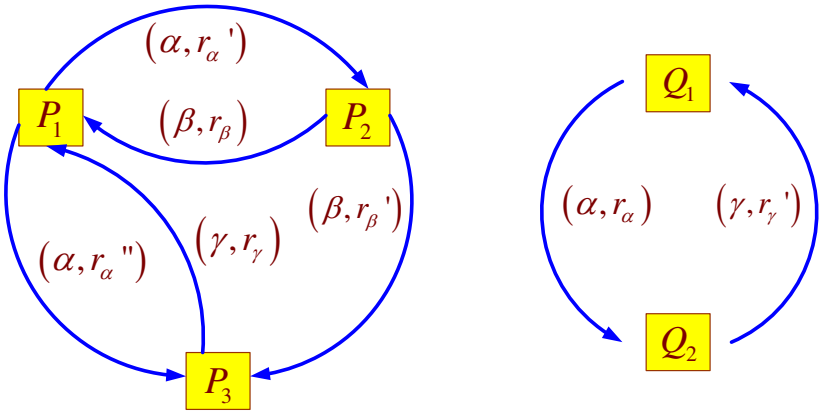


Figure 5.1: *Transition diagram of Model 2*

l	$\alpha^{(P_1\to P_2, Q_1\to Q_2)}$	$\alpha^{(P_1\to P_3, Q_1\to Q_2)}$	$\beta^{P_2\to P_1}$	$\beta^{P_2\to P_3}$	$\gamma^{P_3\to P_1}$	$\gamma^{Q_2\to Q_1}$
P_1	-1	-1	1	0	1	0
P_2	1	0	-1	-1	0	0
P_3	0	1	0	1	-1	0
Q_1	-1	-1	0	0	0	1
Q_2	1	1	0	0	0	-1
$f(\mathbf{x}, l)$	R_1	R_2	$r_\beta \mathbf{x}[P_2]$	$r'_\beta \mathbf{x}[P_2]$	$r_\gamma \mathbf{x}[P_3]$	$r'_\gamma \mathbf{x}[Q_2]$

Table 5.2: *Activity matrix and transition rate functions of Model 2*

The activity matrix and rate functions are listed in Table 5.2. In Table 5.2,

$$R_1 = f(\mathbf{x}, \alpha^{(P_1\to P_2, Q_1\to Q_2)}) = \frac{r'_\alpha}{r'_\alpha + r''_\alpha} \min((r'_\alpha + r''_\alpha)\mathbf{x}[P_1], r_\alpha \mathbf{x}[Q_1]),$$

$$R_2 = f(\mathbf{x}, \alpha^{(P_1 \to P_3, Q_1 \to Q_2)}) = \frac{r''_\alpha}{r'_\alpha + r''_\alpha} \min((r'_\alpha + r''_\alpha)\mathbf{x}[P_1], r_\alpha \mathbf{x}[Q_1]).$$

According to our approach, the derived ODEs are

$$\begin{aligned}
&\left(\frac{\mathrm{d}\mathbf{x}(P_1)}{\mathrm{d}t}, \frac{\mathrm{d}\mathbf{x}(P_2)}{\mathrm{d}t}, \frac{\mathrm{d}\mathbf{x}(P_3)}{\mathrm{d}t}, \frac{\mathrm{d}\mathbf{x}(Q_1)}{\mathrm{d}t}, \frac{\mathrm{d}\mathbf{x}(Q_2)}{\mathrm{d}t}\right)^T \\
= \quad & \sum_l l f(\mathbf{x}, l) \\
= \quad & f(\mathbf{x}, \alpha^{(P_1 \to P_2, Q_1 \to Q_2)}) \begin{pmatrix} -1 \\ 1 \\ 0 \\ -1 \\ 1 \end{pmatrix} + f(\mathbf{x}, \alpha^{(P_1 \to P_3, Q_1 \to Q_2)}) \begin{pmatrix} -1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \\
+ \quad & r_\beta \mathbf{x}(P_2) \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + r'_\beta \mathbf{x}(P_2) \begin{pmatrix} 0 \\ -1 \\ 1 \\ 0 \\ 0 \end{pmatrix} + r_\gamma \mathbf{x}(P_3) \begin{pmatrix} 1 \\ 0 \\ -1 \\ 0 \\ 0 \end{pmatrix} + r'_\gamma \mathbf{x}(Q_2) \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{pmatrix},
\end{aligned}$$

that is

$$\begin{cases}
\frac{\mathrm{d}\mathbf{x}(P_1)}{\mathrm{d}t} = -\min((r'_\alpha + r''_\alpha)\mathbf{x}(P_1), r_\alpha \mathbf{x}(Q_1)) + r'_\beta \mathbf{x}(P_2)) + r_\gamma \mathbf{x}(P_3) \\
\frac{\mathrm{d}\mathbf{x}(P_2)}{\mathrm{d}t} = \frac{r'_\alpha}{r'_\alpha + r''_\alpha} \min((r'_\alpha + r''_\alpha)\mathbf{x}(P_1), r_\alpha \mathbf{x}(Q_1)) - (r_\beta + r'_\beta)\mathbf{x}(P_2) \\
\frac{\mathrm{d}\mathbf{x}(P_3)}{\mathrm{d}t} = \frac{r''_\alpha}{r'_\alpha + r''_\alpha} \min((r'_\alpha + r''_\alpha)\mathbf{x}(P_1), r_\alpha \mathbf{x}(Q_1)) + r_\beta \mathbf{x}(P_2) - r_\gamma \mathbf{x}(P_3) \\
\frac{\mathrm{d}\mathbf{x}(Q_1)}{\mathrm{d}t} = -\min((r'_\alpha + r''_\alpha)\mathbf{x}(P_1), r_\alpha \mathbf{x}(Q_1)) + r'_\gamma \mathbf{x}(Q_2) \\
\frac{\mathrm{d}\mathbf{x}(Q_2)}{\mathrm{d}t} = \min((r'_\alpha + r''_\alpha)\mathbf{x}(P_1), r_\alpha \mathbf{x}(Q_1)) - r'_\gamma \mathbf{x}(Q_2)
\end{cases}.$$

Hereafter the notation $\mathbf{x}[\cdot]$ indicates a discrete state $\mathbf{x}$, while $\mathbf{x}(\cdot)$ or $\mathbf{x}(\cdot, t)$ reflects a continuous state $\mathbf{x}$ at time t.

Notice that in the PEPA language the passive rate "$\top$" is in fact a notation rather than a number. Since $0 \cdot \top = 0$ is assumed in Chapter 3 (see Remark 3.4.1), in the above ODEs the terms such as "$\min\{A\top, rB\}$" are therefore interpreted as [BGH07]:

$$\min\{A\top, rB\} = \begin{cases} rB, & A > 0, \\ 0, & A = 0. \end{cases}$$

For example, if r_α is a passive rate, i.e. $r_\alpha = \top$, then the derived ODEs are as

$$\begin{cases} \frac{\mathrm{d}\mathbf{x}(P_1)}{\mathrm{d}t} = -\min((r'_\alpha + r''_\alpha)\mathbf{x}(P_1), \mathbf{x}(Q_1)\top) + r'_\beta \mathbf{x}(P_2) + r_\gamma \mathbf{x}(P_3) \\ \frac{\mathrm{d}\mathbf{x}(P_2)}{\mathrm{d}t} = \frac{r'_\alpha}{r'_\alpha + r''_\alpha} \min((r'_\alpha + r''_\alpha)\mathbf{x}(P_1), \mathbf{x}(Q_1)\top) - (r_\beta + r'_\beta)\mathbf{x}(P_2) \\ \frac{d\mathbf{x}(P_3)}{dt} = \frac{r''_\alpha}{r'_\alpha + r''_\alpha} \min((r'_\alpha + r''_\alpha)\mathbf{x}(P_1), \mathbf{x}(Q_1)\top) + r_\beta \mathbf{x}(P_2) - r_\gamma \mathbf{x}(P_3) \\ \frac{\mathrm{d}\mathbf{x}(Q_1)}{\mathrm{d}t} = -\min((r'_\alpha + r''_\alpha)\mathbf{x}(P_1), \mathbf{x}(Q_1)\top) + r'_\gamma \mathbf{x}(Q_2) \\ \frac{\mathrm{d}\mathbf{x}(Q_2)}{\mathrm{d}t} = \min((r'_\alpha + r''_\alpha)\mathbf{x}(P_1), \mathbf{x}(Q_1)\top) - r'_\gamma \mathbf{x}(Q_2) \end{cases}.$$

Introducing the passive rate into ODEs is first considered in [BGH07] and [Hay07b]. We should point out that the above definition of "$\min\{A\top, rB\}$" may result in jumps in the functions on the right side of the ODEs. Then, by the theory of ordinary differential equations, these ODEs may have no solutions. In this case, in order to guarantee the existence of solutions, these ODEs should be interpreted as difference rather than differential equations, or considered as integration equations.

In the remainder of this thesis, unless otherwise stated there are no passive rates involved in the derived ODEs. As we will show in the following subsection, if there are no passive rates, then the derived ODEs must have solutions in the time interval $[0, \infty)$.

5.2.3 Existence and uniqueness of ODE solution

For any set of ODEs, it is important to consider if the equations have a solution, and if so whether that solution is unique.

Theorem 5.2.1. *For a given PEPA model without passive rates, the derived ODEs from this model have a unique solution in the time interval* $[0, \infty)$.

Proof. Notice that each entry of $F(\mathbf{x}) = \sum_l l f(\mathbf{x}, l)$ is a linear combination of the rate functions $f(\mathbf{x}, l)$, so $F(\mathbf{x})$ is Lipschitz continuous since each $f(\mathbf{x}, l)$ is Lipschitz continuous by Proposition 3.4.3 in Chapter 3. That is, there exits $M > 0$ such that $\forall \mathbf{x}, \mathbf{y}$,

$$||F(\mathbf{x}) - F(\mathbf{y})|| \leq M||\mathbf{x} - \mathbf{y}||. \tag{5.5}$$

By Theorem C.1.1 in Appendix C.1, the derived ODEs have a unique solution in $[0, \infty)$. □

In the remainder of this subsection, we introduce a proposition, which states that the ODEs

derived from PEPA models satisfy a conservation law. As we have mentioned, in the formula

$$\frac{\mathrm{d}\mathbf{x}(U,t)}{\mathrm{d}t} = -\sum_{\{l|U\in \mathrm{pre}(l)\}} f(\mathbf{x},l) + \sum_{\{l|U\in \mathrm{post}(l)\}} f(\mathbf{x},l),$$

the term $\sum_{\{l|U\in \mathrm{pre}(l)\}} f(\mathbf{x},l)$ represents the exit rates in the local derivative U, while the term $\sum_{\{l|U\in \mathrm{post}(l)\}} f(\mathbf{x},l)$ reflects the entry rates in U. For each type of component at any time, the sum of all exit activity rates must be equal to the sum of all entry activity rates, since the system is closed and there is no exchange with the environment. This leads to the following proposition.

Proposition 5.2.1. *Let C_{i_j} be a local derivative of component type C_i. Then for any i and t, $\sum_j \frac{\mathrm{d}\mathbf{x}\left(C_{i_j},t\right)}{\mathrm{d}t} = 0$, and $\sum_j \mathbf{x}\left(C_{i_j},t\right) = \sum_j \mathbf{x}\left(C_{i_j},0\right)$.*

Proof. We have mentioned in Chapter 3 that the numbers of -1 and 1 appearing in the entries of any transition vector l, which correspond to the component type C_i, are the same, i.e.

$$\#\{j : l[C_{i_j}] = -1\} = \#\{j : l[C_{i_j}] = 1\}. \tag{5.6}$$

Let $\mathbf{y}$ be an indicator vector with the same dimension as l satisfying:

$$\mathbf{y}[C_{i_j}] = \begin{cases} 1, & \text{if } l[C_{i_j}] = \pm 1, \\ 0, & \text{otherwise.} \end{cases}$$

So $\mathbf{y}^T l = 0$ by (5.6). Thus

$$\mathbf{y}^T \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{y}^T \sum_l l f(\mathbf{x},l) = \sum_l \mathbf{y}^T l f(\mathbf{x},l) = 0.$$

That is, $\sum_j \frac{\mathrm{d}\mathbf{x}\left(C_{i_j},t\right)}{\mathrm{d}t} = \mathbf{y}^T \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = 0$. So $\sum_j \mathbf{x}\left(C_{i_j},t\right)$ is a constant and equal to $\sum_j \mathbf{x}\left(C_{i_j},0\right)$, i.e. the number of the copies of component type C_i in the system initially. □

Proposition 5.2.1 means that the ODEs satisfy a *Conservation Law*, i.e. the number of each kind of component remains constant at all times.

5.3 Convergence of ODE Solution: without Synchronisations

Now we consider PEPA models without synchronisation. For this special class of PEPA models, we will show that the solutions of the derived ODEs have finite limits. Moreover, the limits coincide with the steady-state probability distributions of the underlying CTMCs.

5.3.1 Features of ODEs without synchronisations

Suppose the PEPA model has no synchronisation. Without loss of generality, we suppose that there is only one kind of component C in the system. In fact, if there are several types of component in the system, the ODEs related to the different types of component can be separated and treated independently since there are no interactions between them. Thus, we assume there is only one kind of component C and that C has k local derivatives: $C_1, C_2, \cdots, C_k$. Then (5.1) is

$$\frac{\mathrm{d}\left(\mathbf{x}(C_1,t),\cdots,\mathbf{x}(C_k,t)\right)^T}{\mathrm{d}t} = \sum_l l f(\mathbf{x},l). \tag{5.7}$$

Since (5.7) are linear ODEs, we may rewrite (5.7) as the following matrix form:

$$\frac{\mathrm{d}\left(\mathbf{x}(C_1,t),\cdots,\mathbf{x}(C_k,t)\right)^T}{\mathrm{d}t} = Q^T\left(\mathbf{x}(C_1,t),\cdots,\mathbf{x}(C_k,t)\right)^T, \tag{5.8}$$

where $Q = (q_{ij})$ is a $k \times k$ matrix.

Q has many good properties.

Proposition 5.3.1. *$Q = (q_{ij})_{k\times k}$ in (5.8) is an infinitesimal generator matrix, that is, $(q_{ij})_{k\times k}$ satisfies*

1. $0 \le -q_{ii} < \infty$ *for all* i*;*
2. $q_{ij} \ge 0$ *for all* $i \ne j$*;*
3. $\sum_{j=1}^{k} q_{ij} = 0$ *for all* i*.*

Proof. According to (5.8), we have

$$\frac{\mathrm{d}\mathbf{x}(C_i,t)}{\mathrm{d}t} = \sum_{j=1}^{k} \mathbf{x}(C_j,t) q_{ji}. \tag{5.9}$$

Notice by (5.3),

$$\frac{\mathrm{d}\mathbf{x}(C_i,t)}{\mathrm{d}t} = -\sum_{\{l|C_i\in \mathrm{pre}(l)\}} f(\mathbf{x},l) + \sum_{\{l|C_i\in \mathrm{post}(l)\}} f(\mathbf{x},l).$$

So

$$\sum_{j=1}^{k} \mathbf{x}(C_j,t)q_{ji} = -\sum_{\{l|C_i\in \mathrm{pre}(l)\}} f(\mathbf{x},l) + \sum_{\{l|C_i\in \mathrm{post}(l)\}} f(\mathbf{x},l). \tag{5.10}$$

Since there is no synchronisation in the system, the transition function $f(\mathbf{x},l)$ is linear with respect to $\mathbf{x}$ and there is no nonlinear term ,"$\min$", in it. In particular, if $C_i \in \mathrm{pre}(l)$, then $f(\mathbf{x},l) = r(C_i)_l\mathbf{x}[C_i]$, which is the apparent rate of l in C_i in state $\mathbf{x}$ defined in Definition 3.4.1 in Chapter 3. We should point out that according to our semantics of mapping PEPA models to ODEs, the fluid approximation-version of $f(\mathbf{x},l)$ also holds, i.e. $f(\mathbf{x}(t),l) = r_l(C_i)\mathbf{x}(C_i,t)$. So (5.10) becomes

$$\mathbf{x}(C_i,t)q_{ii} + \sum_{j\neq i} \mathbf{x}(C_j,t)q_{ji} = \mathbf{x}(C_i,t) \sum_{\{l|C_i\in \mathrm{pre}(l)\}} (-r_l(C_i)) + \sum_{\{l|C_i\in \mathrm{post}(l)\}} f(\mathbf{x},l). \tag{5.11}$$

Moreover, as long as $f(\mathbf{x},l) = r_l(C_i)\mathbf{x}(C_i)$ for some l and some positive constants $r_l(C_i)$, which implies that l can be fired at C_i, we must have $C_i \in \mathrm{pre}(l)$. That is to say, if $C_i \in \mathrm{post}(l)$ then $f(\mathbf{x},l)$ cannot be of the form of $r\mathbf{x}(C_i,t)$ for any constant $r > 0$. Otherwise, we have $C_i \in \mathrm{pre}(l)$, which results a contradiction[1] to $C_i \in \mathrm{post}(l)$. So according to (5.11), we have

$$\mathbf{x}(C_i,t)q_{ii} = \mathbf{x}(C_i,t) \sum_{\{l|C_i\in \mathrm{pre}(l)\}} (-r_l(C_i)), \tag{5.12}$$

$$\sum_{j\neq i} \mathbf{x}(C_j,t)q_{ji} = \sum_{\{l|C_i\in \mathrm{post}(l)\}} f(\mathbf{x},l). \tag{5.13}$$

Thus by (5.12), $q_{ii} = \sum_{\{l|C_i\in \mathrm{pre}(l)\}} (-r_l(C_i))$, and $0 \leq -q_{ii} < \infty$ for all i. Item 1 is proved.

Similarly, for any $C_j,\ j \neq i$, if $f(\mathbf{x},l) = r\mathbf{x}(C_j,t)$ for some l and positive constant r, then obviously C_j is in the pre set of l. That is $C_j \in \mathrm{pre}(l)$. So by (5.13),

$$\mathbf{x}(C_j,t)q_{ji} = \sum_{\{l|C_j\in \mathrm{pre}(l), C_i\in \mathrm{post}(l)\}} f(\mathbf{x},l) = \mathbf{x}(C_j,t) \sum_{l} r_l^{C_j\to C_i}, \tag{5.14}$$

[1] In this chapter we do not allow a self-loop in the considered model. That is, any PEPA definition like "$C \stackrel{def}{=} (\alpha, r).C$" which results in $C \in \mathrm{pre}(\alpha)$ and $C \in \mathrm{post}(\alpha)$ simultaneously, is not allowed.

which implies $q_{ji} = \sum_l r_l^{C_j \to C_i} \geq 0$ for all $i \neq j$, i.e. item 2 holds.

We now prove item 3. By Proposition 5.2.1,

$$\frac{\mathbf{x}(C_1,t)}{dt} + \frac{\mathbf{x}(C_2,t)}{dt} + \cdots + \frac{\mathbf{x}(C_k,t)}{dt} = 0. \tag{5.15}$$

Then by (5.9) and (5.15), for all t,

$$\begin{aligned}
&\mathbf{x}(C_1,t)\sum_{j=1}^{k} q_{1j} + \mathbf{x}(C_2,t)\sum_{j=1}^{k} q_{2j} + \cdots + \mathbf{x}(C_k,t)\sum_{j=1}^{k} q_{kj} \\
=&\sum_{i=1}^{k} \mathbf{x}(C_i,t)\sum_{j=1}^{k} q_{ij} \\
=&\sum_{j=1}^{k}\sum_{i=1}^{k} \mathbf{x}(C_i,t) q_{ij} \\
=&\sum_{j=1}^{k} \frac{\mathrm{d}\mathbf{x}(C_j,t)}{\mathrm{d}t} \\
=&0.
\end{aligned}$$

This implies $\sum_{j=1}^{k} q_{ij} = 0$ for all i. □

We point out that this infinitesimal generator matrix $Q_{k\times k}$ may not be the infinitesimal generator matrix of the CTMC derived via the usual semantics of PEPA (we call it the "original" CTMC for convenience). In fact, the original CTMC has a state space with k^N states and the dimension of its infinitesimal generator matrix is $k^N \times k^N$, where N is the total number of components in the system. However, this $Q_{k\times k}$ is the infinitesimal generator matrix of a CTMC underlying the PEPA model in which there is only one copy of the component, i.e. $N = 1$. To distinguish this from the original one, we refer to this CTMC as the *"singleton" CTMC*.

In the proof of Proposition 5.3.1, we have shown the relationship between the coefficient matrix Q and the activity rates:

$$q_{ii} = -\sum_{\{l|C_i \in \mathrm{pre}(l)\}} r_l(C_i), \quad q_{ij} = \sum_l r_l^{C_i \to C_j} \ (i \neq j).$$

We use an example to illustrate the above equalities:

Model 5.

$$\begin{aligned} P_1 &\stackrel{\text{def}}{=} (\alpha, r_{\alpha'}).P_2 + (\alpha, r_{\alpha''}).P_3 \\ P_2 &\stackrel{\text{def}}{=} (\beta, r_\beta).P_3 \\ P_3 &\stackrel{\text{def}}{=} (\gamma, r_\gamma).P_1 \\ \textit{Model 5} &= (P_1[A] \parallel P_2[B] \parallel P_3[C]) \end{aligned}$$

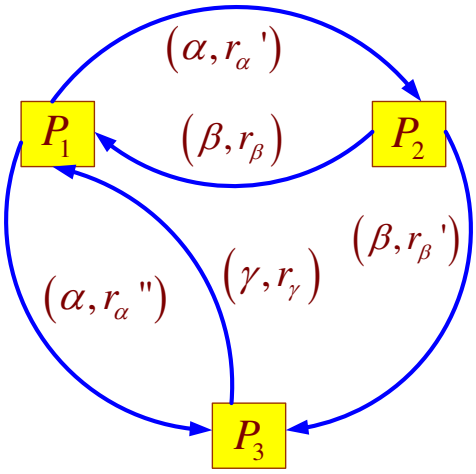


Figure 5.2: *Transition diagram of Model 5*

l	$\alpha^{P_1 \to P_2}$	$\alpha^{P_1 \to P_3}$	$\beta^{P_2 \to P_3}$	$\gamma^{P_3 \to P_1}$
P_1	-1	-1	0	1
P_2	1	0	-1	0
P_3	0	1	1	-1
$f(\mathbf{x}, l)$	$r_{\alpha'}\mathbf{x}[P_1]$	$r_{\alpha''}\mathbf{x}[P_1]$	$r_\beta \mathbf{x}[P_2]$	$r_\gamma \mathbf{x}[P_3]$

Table 5.3: *Activity matrix and transition rate function of Model 5*

The transition diagram of Model 5 is shown in Figure 5.2. The activity matrix and transition rate functions are presented in Table 5.3. The derived ODEs are

$$\begin{cases} \frac{\mathrm{d}\mathbf{x}(P_1)}{\mathrm{d}t} = -(r'_\alpha + r''_\alpha)\mathbf{x}(P_1) + r_\gamma \mathbf{x}(P_3) \\ \frac{\mathrm{d}\mathbf{x}(P_2)}{\mathrm{d}t} = r'_\alpha \mathbf{x}(P_1) - r_\beta \mathbf{x}(P_2) \\ \frac{\mathrm{d}\mathbf{x}(P_3)}{\mathrm{d}t} = r''_\alpha \mathbf{x}(P_1) + r_\beta \mathbf{x}(P_2) - r_\gamma \mathbf{x}(P_3) \end{cases}$$

or

$$\begin{pmatrix} \frac{\mathrm{d}\mathbf{x}(P_1)}{\mathrm{d}t} \\ \frac{\mathrm{d}\mathbf{x}(P_2)}{\mathrm{d}t} \\ \frac{\mathrm{d}\mathbf{x}(P_3)}{\mathrm{d}t} \end{pmatrix} = \begin{pmatrix} -(r_{\alpha'}+r_{\alpha''}) & 0 & r_\gamma \\ r_{\alpha'} & -r_\beta & 0 \\ r_{\alpha''} & r_\beta & -r_\gamma \end{pmatrix} \begin{pmatrix} \mathbf{x}(P_1) \\ \mathbf{x}(P_2) \\ \mathbf{x}(P_3) \end{pmatrix} = Q^T \begin{pmatrix} \mathbf{x}(P_1) \\ \mathbf{x}(P_2) \\ \mathbf{x}(P_3) \end{pmatrix},$$

where

$$Q = \begin{pmatrix} -(r_{\alpha'}+r_{\alpha''}) & r_{\alpha'} & r_{\alpha''} \\ 0 & -r_\beta & r_\beta \\ r_\gamma & 0 & -r_\gamma \end{pmatrix}$$

is clearly an infinitesimal generator matrix. Obviously, $q_{11} = -(r_{\alpha'}+r_{\alpha''}) = -\sum_{\{l|P_1\in \mathrm{pre}(l)\}} r_l(P_1)$. Similarly, $q_{ij} = \sum_l r_l^{P_i\to P_j}$ $(i \neq j)$.

5.3.2 Convergence and consistency for the ODEs

Proposition 5.3.1 illustrates that the coefficient matrix of the derived ODEs is an infinitesimal generator. If there is only one component in the system, then equation (5.8) captures the probability distribution evolution equations of the original CTMC. Based on this proposition, we can furthermore determine the convergence of the solutions.

Theorem 5.3.1. *Suppose* $\mathbf{x}(C_j,t)$ $(j = 1,2,\cdots,k)$ *satisfy (5.7), then for any given initial values* $\mathbf{x}(C_j,0) \geq 0$ $(j = 1,2,\cdots,k)$*, there exist constants* $\mathbf{x}(C_j,\infty)$*, such that*

$$\lim_{t\to\infty} \mathbf{x}(C_j,t) = \mathbf{x}(C_j,\infty), \quad j = 1,2,\cdots,k. \tag{5.16}$$

Proof. By Proposition 5.3.1, the matrix Q in (5.8) is an infinitesimal generator matrix. Consider a "singleton" CTMC which has the state space $S = \{C_1, C_2, \cdots, C_k\}$, the infinitesimal generator matrix Q in (5.8) and the initial probability distribution $\pi(C_j,0) = \frac{\mathbf{x}(C_j,0)}{N}$ $(j = 1,2,\cdots,k)$. Then according to Markov theory ([BGdMT98], page 52), $\pi(C_j,t)$ $(j = 1,2,\cdots,k)$, the probability distribution of this new CTMC at time t, satisfies

$$\frac{\mathrm{d}\left(\pi(C_1,t),\cdots,\pi(C_k,t)\right)}{\mathrm{d}t} = \left(\pi(C_1,t),\cdots,\pi(C_k,t)\right) Q \tag{5.17}$$

Since the singleton CTMC is assumed irreducible and positive-recurrent, it has a steady-state

probability distribution $\{\pi(C_j, \infty)\}_{j=1}^{k}$, and

$$\lim_{t\to\infty} \pi(C_j, t) = \pi(C_j, \infty), \quad j = 1, 2, \cdots, k. \tag{5.18}$$

Note that $\frac{\mathbf{x}(C_j,t)}{N}$ also satisfies (5.17) with the initial values $\frac{\mathbf{x}(C_j,0)}{N}$ equal to $\pi(C_j, 0)$, where N is the population of the components. By the uniqueness of the solutions of (5.17), we have

$$\frac{\mathbf{x}(C_j, t)}{N} = \pi(C_j, t), \quad j = 1, 2, \cdots, k, \tag{5.19}$$

and hence by (5.18),

$$\lim_{t\to\infty} \mathbf{x}(C_j, t) = \lim_{t\to\infty} N\pi(C_j, t) = N\pi(C_j, \infty), \quad j = 1, 2, \cdots, k.$$

□

Remark 5.3.1. *Suppose there are m types of components in the system: $C_1, C_2, \cdots, C_m$, each with $k_1, k_2, \cdots, k_m$ local derivatives respectively. Since there is no cooperation between different types of component, we can deal with each type independently. Thus, by Theorem 5.3.1, for each component type C_i,*

$$\lim_{t\to\infty} \mathbf{x}(C_{i_j}, t) = N_i\pi(C_{i_j}, \infty), \quad j = 1, 2, \cdots, k_i, \tag{5.20}$$

where $\{\pi(C_{i_j})\}_{j=1,2,\cdots,k_i}$ are the corresponding steady state distributions, and N_i is the population of C_i, $i = 1, 2, \cdots, m$.

It is shown in [Gil05] that for some special examples the equilibrium solutions of the ODEs coincide with the steady state probability distributions of the underlying original CTMC. This theorem states that this holds for all for PEPA models without synchronisations.

5.4 Relating to Density Dependent CTMCs

For a PEPA model without synchronisation, the solution of the derived ODEs through the fluid approximation has a finite limit that is consistent with the steady-state distribution of the corresponding singleton CTMC, as Theorem 5.3.1 exposes. However, a general PEPA model may have synchronisations, which result in the nonlinearity of the derived ODEs. Generally, it is difficult to rely on pure analytical methods to explore the asymptotic behaviour of the solution

of the derived ODEs from an arbitrary PEPA model (except for some special classes of models, see the next chapter).

Fortunately, Kurtz's theorem [Kur70, EK86] establishes the relationship between a sequence of Markov chains and a corresponding set of ODEs: the complete solution of some ODEs is the limit of a sequence of Markov chains. In the context of PEPA, the derived ODEs can be considered as the limit of pure jump Markov processes, as first exposed in [GHS08] for a special case. Thus we may investigate the convergence of the ODE solutions by alternatively studying the corresponding property of the Markov chains through this consistency relationship.

This approach leads to the result presented in the next section: under a particular condition the solution will converge and the limit is consistent with the limit steady-state probability distribution of a family of CTMCs underlying the given PEPA model. Let us first introduce the concept of density dependent Markov chains underlying PEPA models.

5.4.1 Density dependent Markov chains from PEPA models

In the numerical state vector representation scheme, each vector is a single state and the rates of the transitions between states are specified by the rate functions. For example, the transition from state $\mathbf{s}$ to $\mathbf{s}+l$ can be written as

$$\mathbf{s} \xrightarrow{(l,f(\mathbf{s},l))} \mathbf{s}+l.$$

Since all the transitions are only determined by the current state rather than the previous ones, given any starting state a CTMC can be obtained. More specifically, the state space of the CTMC is the set of all reachable numerical state vectors $\mathbf{s}$. The infinitesimal generator is determined by the transition rate function,

$$q_{\mathbf{s},\mathbf{s}+l} = f(\mathbf{s},l). \tag{5.21}$$

Because the transition rate function is defined according to the semantics of PEPA, the CTMC mentioned above is in fact the aggregated CTMC underlying the given PEPA model. In other words, the transition rate of the aggregated CTMC is specified by the transition rate function in Definition 3.4.2.

It is obvious that the aggregated CTMC depends on the starting state of the given PEPA model. By altering the population of components presented in the model, which can be done by varying the initial states, we may get a sequence of aggregated CTMCs. Moreover, Proposition 3.4.3 indicates that the transition rate function has the homogenous property: $Hf(\mathbf{s}/H, l) = f(\mathbf{s}, l)$, $\forall H > 0$. This property identifies the aggregated CTMC to be *density dependent*.

Definition 5.4.1. *[Kur70]. A family of CTMCs $\{X_n\}_n$ is called* density dependent *if and only if there exists a continuous function $f(\mathbf{x}, l),\ \mathbf{x} \in \mathbb{R}^d,\ l \in \mathbb{Z}^d$, such that the infinitesimal generators of X_n are given by:*

$$q^{(n)}_{\mathbf{s},\mathbf{s}+l} = nf(\mathbf{s}/n, l), \quad l \neq 0,$$

where $q^{(n)}_{\mathbf{s},\mathbf{s}+l}$ denotes an entry of the infinitesimal generator of X_n, $\mathbf{s}$ a numerical state vector and l a transition vector.

This allows us to conclude the following proposition.

Proposition 5.4.1. *Let $\{X_n\}$ be a sequence of aggregated CTMCs generated from a given PEPA model (by scaling the initial state), then $\{X_n\}$ is density dependent.*

Proof. For any n, the transition between states is determined by

$$q^{(n)}_{\mathbf{s},\mathbf{s}+l} = f(\mathbf{s}, l),$$

where $\mathbf{s}, \mathbf{s} + l$ are state vectors, l corresponds to an activity, $f(\mathbf{s}, l)$ is the rate of the transition from state $\mathbf{s}$ to $\mathbf{s} + l$. By Proposition 3.4.3,

$$nf(\mathbf{s}/n, l) = f(\mathbf{s}, l).$$

So the infinitesimal generator of X_n is given by:

$$q^{(n)}_{\mathbf{s},\mathbf{s}+l} = f(\mathbf{s}, l) = nf(\mathbf{s}/n, l), \quad l \neq 0.$$

Therefore, $\{X_n\}$ is a sequence of density dependent CTMCs. □

In particular, the family of density dependent CTMCs, $\{X_n(t)\}$, derived from a given PEPA model with the starting condition $X_n(0) = n\mathbf{x}_0\ (\forall n)$, is called the *density dependent CTMCs*

associated with $\mathbf{x}_0$. The CTMCs $\frac{X_n(t)}{n}$ are called the *concentrated density dependent* CTMCs. Here n is called the *concentration level*, indicating that the entries within the numerical vector states (of $\frac{X_n(t)}{n}$) are incremented and decremented in steps of $\frac{1}{n}$.

For example, recall Model 1,

$$\begin{aligned} User_1 &\stackrel{def}{=} (task_1, a).User_2 \\ User_2 &\stackrel{def}{=} (task_2, b).User_1 \\ Provider_1 &\stackrel{def}{=} (task_1, a).Provider_2 \\ Provider_2 &\stackrel{def}{=} (reset, d).Provider_1 \\ (User_1[M]) &\underset{\{task1\}}{\bowtie} (Provider_1[N]) \end{aligned}$$

The activity matrix and transition rate functions have been specified in Table 5.4. In this table, $U_i, P_i \ (i = 1, 2)$ are the local derivatives representing $User_i$ and $Provider_i$ respectively. For convenience, the labelled activities or transition vectors ${task_1}^{(U_1 \to U_2, P_1 \to P_2)}$, $task_2^{U_2 \to U_1}$, ${reset}^{P_2 \to P_1}$ will subsequently be denoted by $l^{task_1}, l^{task_2}, l^{reset}$ respectively.

l	${task_1}^{(U_1 \to U_2, P_1 \to P_2)}$	$task_2^{U_2 \to U_1}$	${reset}^{P_2 \to P_1}$
U_1	-1	1	0
U_2	1	-1	0
P_1	-1	0	1
P_2	1	0	-1
$f(\mathbf{x}, l)$	$a \min(\mathbf{x}[U_1], \mathbf{x}[P_1])$	$b\mathbf{x}[U_2]$	$d\mathbf{x}[P_2]$

Table 5.4: *Activity matrix and transition rate function of Model 1*

Suppose $\mathbf{x}_0 = (M, 0, N, 0)^T = (1, 0, 1, 0)^T$. Let $X_1(t)$ be the aggregated CTMC underlying Model 1 with initial state $\mathbf{x}_0$. Then the state space of $X_1(t)$, denoted by S_1, is composed of

$$\begin{aligned} \mathbf{s}_1 &= (1, 0, 1, 0)^T, \quad \mathbf{s}_2 = (0, 1, 0, 1)^T, \\ \mathbf{s}_3 &= (1, 0, 0, 1)^T, \quad \mathbf{s}_4 = (0, 1, 1, 0)^T. \end{aligned} \tag{5.22}$$

According to the transition rate functions presented in Table 5.4, we have, for instance,

$$q^{(1)}_{\mathbf{s}_1, \mathbf{s}_2} = q_{\mathbf{s}_1, \mathbf{s}_1 + l^{task_1}} = f(\mathbf{s}_1, l^{task_1}) = a \min(\mathbf{s}_1[U_1], \mathbf{s}_1[P_1]) = a.$$

Varying the initial states we may get other aggregated CTMCs. For example, let $X_2(t)$ be the

aggregated CTMC corresponding to the initial state $X_2(0) = 2\mathbf{x}_0 = (2,0,2,0)^T$. Then the state space S_2 of $X_2(t)$ has the states

$$\begin{aligned} &\mathbf{s}_1 = (2,0,2,0)^T, \quad \mathbf{s}_2 = (1,1,1,1)^T, \quad \mathbf{s}_3 = (1,1,2,0)^T, \\ &\mathbf{s}_4 = (1,1,0,2)^T, \quad \mathbf{s}_5 = (0,2,1,1)^T, \quad \mathbf{s}_6 = (2,0,1,1)^T, \\ &\mathbf{s}_7 = (0,2,0,2)^T, \quad \mathbf{s}_8 = (0,2,2,0)^T, \quad \mathbf{s}_9 = (2,0,0,2)^T. \end{aligned} \tag{5.23}$$

The rate of transition from $\mathbf{s}_1$ to $\mathbf{s}_2$ is determined by

$$q^{(2)}_{\mathbf{s}_1,\mathbf{s}_2} = q_{\mathbf{s}_1,\mathbf{s}_1+l^{task_1}} = f(\mathbf{s}_1, l^{task_1}) = 2a = 2f(\mathbf{s}_1/2, l^{task_1}).$$

Similarly, let $X_n(t)$ be the aggregated CTMC corresponding to the initial state $X_n(0) = n\mathbf{x}_0$. Then the transition from $\mathbf{s}$ to $\mathbf{s}+l$ is determined by

$$q^{(n)}_{\mathbf{s},\mathbf{s}+l} = f(\mathbf{s}, l) = nf(\mathbf{s}/n, l).$$

Thus a family of aggregated CTMCs, i.e. $\{X_n(t)\}$, has been obtained from Model 1. These derived $\{X_n(t)\}$ are density dependent CTMCs associated with $\mathbf{x}_0$. As illustrated by this example, the density dependent CTMCs are obtained by scaling the starting state $\mathbf{x}_0$. So the starting state of each CTMC is different, because $X_n(0) = n\mathbf{x}_0$, i.e. $X_n(0) = n(M,0,N,0)^T$.

5.4.2 Consistency between the derived ODEs and the aggregated CTMCs

As discussed above, a set of ODEs and a sequence of density dependent Markov chains can be derived from the same PEPA model. The former one is deterministic while the latter is stochastic. However, both of them are determined by the same activity matrix and the same rate functions that are uniquely generated from the given PEPA model. Therefore, it is natural to believe that there is some kind of consistency between them.

As we have mentioned, the complete solution of some ODEs can be the limit of a sequence of Markov chains according to Kurtz's theorem [Kur70, EK86]. Such consistency in the context of PEPA has been previously illustrated for a particular PEPA model [GHS08]. Here we give a modified version of this result for general PEPA models, in which the convergence is in the sense of almost surely rather than probabilistically as in [GHS08]. A sequence converges to a limit almost surely means that events for which this sequence does not converge to this limit have probability zero. Convergence in this sense can imply the convergence in probability, so

it is stronger.

Theorem 5.4.1. *Let $X(t)$ be the solution of the ODEs (5.1) derived from a given PEPA model with initial condition $\mathbf{x}_0$, and let $\{X_n(t)\}$ be the density dependent CTMCs associated with $\mathbf{x}_0$ underlying the same PEPA model. Let $\hat{X}_n(t) = \frac{X_n(t)}{n}$, then for any $t > 0$,*

$$\lim_{n\to\infty} \sup_{u\le t} \|\hat{X}_n(u) - X(u)\| = 0 \qquad a.s. \tag{5.24}$$

Proof. According to Kurtz's theorem, which is listed in Appendix C.1, it is sufficient to prove: for any compact set $K \subset \mathbb{R}^{N_d}$,

1. $\exists M_K > 0$ such that $\|F(\mathbf{x}) - F(\mathbf{y})\| \le M_K \|\mathbf{x} - \mathbf{y}\|$;
2. $\sum_l \|l\| \sup_{\mathbf{x}\in K} f(\mathbf{x}, l) < \infty$.

Obviously, above term 1 is satisfied. Since $f(\mathbf{x}, l)$ is continuous by Proposition 3.4.3 in Chapter 3, it is bounded on any compact K. Notice that any entry of l takes values in $\{1, -1, 0\}$, so $\|l\|$ is bounded. Thus term 2 is satisfied, which completes the proof. □

Based on such a relationship between the derived ODEs and the aggregated CTMCs, the convergence problem of the solutions will be discussed in the next section, while the boundedness and nonnegativeness will be first presented in the following subsection.

5.4.3 Boundedness and nonnegativeness of ODE solutions

Theorem 5.4.1 allows us to investigate the properties of $X(t)$ through studying the characteristics of the family of CTMCs $\hat{X}_n(t) = \frac{X_n(t)}{n}$. Notice that $X_n(t)$ takes values in the state space which corresponds to the starting state $n\mathbf{x}_0$. The structural properties of these state spaces, such as boundedness and nonnegativeness of each entry in the numerical state vectors, have been demonstrated in Chapter 4. So the ODE solution $X(t)$ inherits these characteristics since $X(t)$ is the limit of $\hat{X}_n(t)$ as n goes to infinity. That is, $X(t)$ is bounded and nonnegative. The proof is trivial and omitted here. Instead, a purely analytic proof of these properties will be given in the next chapter. Moreover, the proposition of $X(t)$ satisfying a conservation law, i.e. Proposition 5.2.1, can also been easily obtained because any state in the state space satisfies the law as presented in Chapter 4.

Theorem 5.4.1 just states the approximation but does not state how to approximate the ODEs by the family of CTMCs, i.e. it does not provide the error bounds of the approximation. Detailed techniques and discussions of this topic have been presented in [DN08]. The reader is referred to that paper since this thesis does not discuss this topic theoretically. However, experimental results will be provided to show the error bounds of this kind of approximation (see the case study in Chapter 7). Moreover, some other topics about the fluid approximations, such as the comparison between accuracy and computational cost, numerical solution methods, and how to derive performance measures from ODEs, will be discussed in detail in Chapter 7.

5.5 Convergence of ODE Solution: under a Particular Condition

Analogous to the steady-state probability distributions of the Markov chains underlying PEPA models, upon which performance measures such as throughput and utilisation can be derived, we expect the solution of the generated ODEs to have similar equilibrium conditions. In particular, if the solution has a limit as time goes to infinity we will be able to similarly obtain the performance from the steady state, i.e. the limit. Therefore, whether the solution of the derived ODEs converges becomes an important problem.

We should point out that Kurtz's theorem cannot directly apply to the problem of whether or not the solution the derived ODEs converges. This is because Kurtz's theorem only deals with the approximation between the ODEs and Markov chains during any finite time, rather than considering the asymptotic behaviour of the ODEs as time goes to infinity. This section will present our investigation and results about this problem.

We follow the assumptions in Theorem 5.4.1. Denote the expectation of $\hat{X}_n(t)$ as $\hat{M}_n(t)$, i.e. $\hat{M}_n(t) = E[\hat{X}_n(t)]$. For any t, the stochastic processes $\{\hat{X}_n(t)\}_n$ converge to the deterministic $X(t)$ when n tends to infinity, as Theorem 5.4.1 shows. It is not surprising to see that $\{\hat{M}_n(t)\}_n$, the expectations of $\{\hat{X}_n(t)\}_n$, also converge to $X(t)$ as $n \to \infty$:

Lemma 5.5.1. *For any t,*

$$\lim_{n\to\infty} \hat{M}_n(t) = X(t).$$

Proof. Since $X(t)$ is deterministic, then $E[X(t)] = X(t)$. By Theorem 5.4.1, for all t, $\hat{X}_n(t)$ converges to $X(t)$ almost surely as n goes to infinity. Notice that $\hat{X}_n(t)$ is bounded (see the discussion in Section 5.4.3), then by Lebesgue's dominant convergence theorem given in Ap-

pendix C.1, we have

$$\lim_{n\to\infty} E\|\hat{X}_n(t) - X(t)\| = 0.$$

Since a norm $\|\cdot\|$ can be considered as a convex function, by Jensen's inequality (Theorem2.2 [BZ99]), we have $\|(E[\cdot])\| \leq E[\|\cdot\|]$. Therefore,

$$\begin{aligned}\lim_{n\to\infty} \|\hat{M}_n(t) - X(t)\| &= \lim_{n\to\infty} \|E[\hat{X}_n(t)] - E[X(t)]\| \\ &\leq \lim_{n\to\infty} E\|\hat{X}_n(t) - X(t)\| \\ &= 0.\end{aligned}$$

□

Lemma 5.5.1 states that the ODE solution $X(t)$ is just the limit function of the sequence of the expectation functions of the corresponding density dependent Markov chains. This provides some clues: the characteristics of the limit $X(t)$ depend on the properties of $\{\hat{M}_n(t)\}_n$. Therefore, we expect to be able to investigate $X(t)$ by studying $\{\hat{M}_n(t)\}_n$.

Since $\hat{M}_n(t)$ is the expectation of the Markov chain $\hat{X}_n(t)$, $\hat{M}_n(t)$ can be expressed by a formula in which the transient probability distribution is involved. That is,

$$\hat{M}_n(t) = E[\hat{X}_n(t)] = \sum_{\mathbf{x}\in\hat{S}_n} \mathbf{x}\hat{\pi}_t^n(\mathbf{x}),$$

where $\hat{S}_n$ is the state space, $\hat{\pi}_t^n(\cdot)$ is the probability distribution of $\hat{X}_n$ at time t. Let S_n and $\pi_t^n(\cdot)$ be the state space and the probability distribution of $X_n(t)$ respectively[2]. Then

$$\hat{M}_n(t) = E[\hat{X}_n(t)] = E\left[\frac{X_n(t)}{n}\right] = \sum_{\mathbf{x}\in S^n} \frac{\mathbf{x}}{n}\pi_t^n(\mathbf{x}).$$

We have assumed the Markov chains underlying PEPA models to be irreducible and positive-recurrent. Then the transient probability distributions of these Markov chains will converge to the corresponding steady-state probability distributions. We denote the steady-state probability distributions of $X_n(t)$ and $\hat{X}_n(t)$ as $\pi_\infty^n(\cdot)$ and $\hat{\pi}_\infty^n(\cdot)$ respectively. Then, we have a lemma.

Lemma 5.5.2. *For any n, there exists a $\hat{M}_n(\infty)$, such that*

$$\lim_{t\to\infty} \hat{M}_n(t) = \hat{M}_n(\infty).$$

[2]We should point out that the probability distributions of $X_n(t)$ and $\hat{X}_n(t)$ are the same, i.e. $\pi_t^n(\mathbf{x}) = \hat{\pi}_t^n(\mathbf{x}/n)$.

Proof.

$$\begin{aligned}\lim_{t\to\infty}\hat{M}_n(t) &= \lim_{t\to\infty}\sum_{\mathbf{x}\in S_n}\frac{\mathbf{x}}{n}\pi_t^n(\mathbf{x})\\ &= \sum_{\mathbf{x}\in S_n}\lim_{t\to\infty}\frac{\mathbf{x}}{n}\pi_t^n(\mathbf{x})\\ &= \sum_{\mathbf{x}\in S_n}\frac{\mathbf{x}}{n}\pi_\infty^n(\mathbf{x})\\ &\equiv \hat{M}_n(\infty).\end{aligned}$$

□

Clearly, we also have $\hat{M}_n(\infty) = \sum_{\mathbf{x}\in\hat{S}_n}\mathbf{x}\hat{\pi}_\infty^n(\mathbf{x})$.

Remark 5.5.1. *Currently, we do not know whether the sequence $\{\hat{M}_n(\infty)\}_n$ converges as $n\to\infty$. But since $\{\hat{M}_n(\infty)\}_n$ is bounded which is due to the conservation law that PEPA models satisfy, there exists $\{n'\}\subset\{n\}$ such that $\{\hat{M}_{n'}(\infty)\}$ converges to a limit, namely $\hat{M}_\infty(\infty)$. That is*

$$\lim_{n'\to\infty}\hat{M}_{n'}(\infty) = \hat{M}_\infty(\infty).$$

Thus,

$$\lim_{n'\to\infty}\lim_{t\to\infty}\hat{M}_{n'}(t) = \lim_{n'\to\infty}\hat{M}_{n'}(\infty) = \hat{M}_\infty(\infty). \tag{5.25}$$

At the moment, there are two questions:

1. Whether $\lim_{t\to\infty}\lim_{n'\to\infty}\hat{M}_{n'}(t)$ exists?

2. If $\lim_{t\to\infty}\lim_{n'\to\infty}\hat{M}_{n'}(t)$ exists, whether

$$\lim_{t\to\infty}\lim_{n'\to\infty}\hat{M}_{n'}(t) = \lim_{n'\to\infty}\lim_{t\to\infty}\hat{M}_{n'}(t)?$$

If the answer to the first question is yes, then the solution of the ODEs converges, since by Lemma 5.5.2,

$$\lim_{t\to\infty}X(t) = \lim_{t\to\infty}\lim_{n'\to\infty}\hat{M}_{n'}(t).$$

If the answer to the second question is yes, then the limit of $X(t)$ is consistent with the station-

ary distributions of the Markov chains since

$$\lim_{t\to\infty} X(t) = \lim_{t\to\infty}\lim_{n'\to\infty} \hat{M}_{n'}(t) = \lim_{n'\to\infty}\lim_{t\to\infty} \hat{M}_{n'}(t) = \hat{M}_\infty(\infty).$$

$$\begin{array}{ccc} \hat{M}_{n'}(t) & \xrightarrow[\text{Lemma 5.5.1}]{n'\to\infty} & X(t)(=\hat{M}_\infty(t)) \\ {\scriptstyle t\to\infty}\downarrow \text{Lemma 5.5.2} & & ???\downarrow{\scriptstyle t\to\infty} \\ \hat{M}_{n'}(\infty) & \xrightarrow[n'\to\infty]{\text{Remark 5.5.1}} & \hat{M}_\infty(\infty) \end{array}$$

Figure 5.3: *Convergence and consistency diagram for derived ODEs*

In short, the positive answers to these two questions determine the convergence and consistency for the ODE solution, see Figure 5.3. Fortunately, the two answers can be guaranteed by the condition (5.26) in the following Proposition 5.5.1.

Proposition 5.5.1. ***(A particular condition)*** *If there exist $A, B > 0$, such that*

$$\sup_{n'} \left\| \hat{M}_{n'}(t) - \hat{M}_{n'}(\infty) \right\| < Be^{-At}, \tag{5.26}$$

then $\lim_{t\to\infty} X(t) = \hat{M}_\infty(\infty)$.

Proof.

$$\begin{aligned} \left\| X(t) - \hat{M}_\infty(\infty) \right\| &= \left\| \lim_{n'\to\infty} [\hat{M}_{n'}(t) - \hat{M}_{n'}(\infty)] \right\| \\ &\leq \limsup_{n'\to\infty} \left\| \hat{M}_{n'}(t) - \hat{M}_{n'}(\infty) \right\| \\ &\leq \limsup_{n'\to\infty} \left[\sup_{n'} \left\| \hat{M}_{n'}(t) - \hat{M}_{n'}(\infty) \right\| \right] \\ &\leq \limsup_{n'\to\infty} Be^{-At} \\ &= Be^{-At} \longrightarrow 0, \text{as } t \longrightarrow \infty. \end{aligned}$$

So $\lim_{t\to\infty} X(t) = \hat{M}_\infty(\infty)$. □

Notice that $\hat{M}_n(t) = \sum_{\mathbf{x}\in S^n} \frac{\mathbf{x}}{n}\pi_t^n(\mathbf{x})$ and $\hat{M}_n(\infty) = \sum_{\mathbf{x}\in S^n} \frac{\mathbf{x}}{n}\pi_\infty^n(\mathbf{x})$, so in order to estimate $\left\| \hat{M}_n(t) - \hat{M}_n(\infty) \right\|$ in (5.26), we need first to estimate the difference between π_t^n and π_∞^n.

Lemma 5.5.3. *If there exists $A > 0$ and $B_1 > 0$, such that for any n' and all $\mathbf{x} \in S^{n'}$,*

$$|\pi_t^{n'}(\mathbf{x}) - \pi_\infty^{n'}(\mathbf{x})| \leq \pi_\infty^{n'}(\mathbf{x}) B_1 e^{-At}, \tag{5.27}$$

then there exists $B > 0$ such that $\sup_{n'} \left\| \hat{M}_{n'}(t) - \hat{M}_{n'}(\infty) \right\| < Be^{-At}$ holds.

Proof. We know that $\hat{X}_n(0) = \dfrac{X_n(0)}{n} = \mathbf{x}_0$ for any n. By the conservation law, the population of each entity in any state is determined by the starting state. So for any n' and all $\mathbf{x} \in S^{n'}$, $||\mathbf{x}/n'|| \leq C_1 \sum_{P \in \mathcal{D}} \mathbf{x}_0[P] < \infty$, where $\mathcal{D}$ is the set of all local derivatives and C_1 is a constant independent of n'. Let $C = \sup_{n'} \max_{\mathbf{x} \in S^{n'}} ||\mathbf{x}/n'||$, then $C < \infty$.

$$\begin{aligned}
\left\| \hat{M}_{n'}(t) - \hat{M}_{n'}(\infty) \right\| &= \left\| \sum_{\mathbf{x} \in S^{n'}} \frac{\mathbf{x}}{n'} \pi_t^{n'}(\mathbf{x}) - \sum_{\mathbf{x} \in S^{n'}} \frac{\mathbf{x}}{n'} \pi_\infty^{n'}(\mathbf{x}) \right\| \\
&= \left\| \sum_{\mathbf{x} \in S^{n'}} \frac{\mathbf{x}}{n'} (\pi_t^{n'}(\mathbf{x}) - \pi_\infty^{n'}(\mathbf{x})) \right\| \\
&\leq \sup_{n'} \max_{\mathbf{x} \in S^{n'}} \left\| \frac{\mathbf{x}}{n'} \right\| \sum_{\mathbf{x} \in S^{n'}} |\pi_t^{n'}(\mathbf{x}) - \pi_\infty^{n'}(\mathbf{x})| \\
&= C \sum_{\mathbf{x} \in S^{n'}} |\pi_t^{n'}(\mathbf{x}) - \pi_\infty^{n'}(\mathbf{x})| \\
&\leq C \sum_{\mathbf{x} \in S^{n'}} \pi_\infty^{n'}(\mathbf{x}) B_1 e^{-tA} \\
&= C B_1 e^{-tA}.
\end{aligned}$$

Let $B = CB_1$. Then $\sup_{n'} \left\| \hat{M}_{n'}(t) - \hat{M}_{n'}(\infty) \right\| < Be^{-At}$. □

5.6 Investigation of the Particular Condition

This section will present the study of the particular condition (5.26). We will expose that the condition is related to well-known constants of Markov chains such as the spectral gap and the Log-Sobolev constant. The methods and results developed in the field of functional analysis of Markov chains are utilised to investigate the condition.

5.6.1 An important estimation in the context of Markov kernel

We first give an estimation for the Markov kernel which is defined below. Let Q be the infinitesimal generator of a Markov chain X on a finite state S. Let

$$K_{ij} = \begin{cases} \frac{Q_{ij}}{m}, & i \neq j \\ 1 + \frac{Q_{ii}}{m}, & i = j \end{cases} \qquad \text{where } m = \sup_i(-Q_{ii}).$$

K is a transition probability matrix, satisfying

$$K(\mathbf{x}, \mathbf{y}) \geq 0, \qquad \sum_{\mathbf{y}} K(\mathbf{x}, \mathbf{y}) = 1.$$

K is called an Markov *kernel* (K is also called the *uniformisation* of the CTMC in some literature). A Markov chain on a finite state space S can be described through its *kernel* K. The continuous time semigroup associated with K is defined by

$$H_t = \exp(-t(I - K)).$$

Let π be the unique stationary measure of the Markov chain. Then $H_t(\mathbf{x}, \mathbf{y}) \to \pi(\mathbf{y})$ as t tends to infinity. Following the convention in the literature we will also use (K, π) to represent a Markov chain.

Notice

$$Q = m(K - I), \qquad K = \frac{Q}{m} + I.$$

Clearly,

$$P_t = e^{tQ} = e^{mt(K-I)} = e^{-mt(I-K)} = H_{mt},$$

and thus $H_t = P_{\frac{t}{m}}$, where P_t is called the semigroup associated with the infinitesimal generator Q. An estimation of H_t is given below.

Lemma 5.6.1. *(Corollary 2.2.6, [SC97]) Let (K, π) be a finite Markov chain, and $\pi(*) = \min_{\mathbf{x} \in S} \pi(\mathbf{x})$. Then*

$$\sup_{\mathbf{x},\mathbf{y}} \left| \frac{H_t(\mathbf{x}, \mathbf{y})}{\pi(\mathbf{y})} - 1 \right| \leq e^{2-c} \quad \textit{for } t = \frac{1}{\alpha} \log\log \frac{1}{\pi(*)} + \frac{c}{\lambda}, \tag{5.28}$$

where $\lambda > 0, \alpha > 0$ are the spectral gap and the Log-Sobolev constant respectively, which are defined and interpreted in Appendix C.2.

It can be implied from (5.28) that $\forall \mathbf{x}, \mathbf{y} \in S$,

$$|H_t(\mathbf{x},\mathbf{y}) - \pi(\mathbf{y})| \leq \pi(\mathbf{y}) \left[e^{2+\frac{\lambda}{\alpha} \log\log \frac{1}{\pi(*)}} \right] e^{-\lambda t}. \tag{5.29}$$

Since $H_t = P_{\frac{t}{m}}$, so

$$\left| P_{\frac{t}{m}}(\mathbf{x},\mathbf{y}) - \pi(\mathbf{y}) \right| \leq \pi(\mathbf{y}) \left[e^{2+\frac{\lambda}{\alpha} \log\log \frac{1}{\pi(*)}} \right] e^{-\lambda t}, \tag{5.30}$$

and thus (replacing "t" by "mt" on the both sides of (5.30)),

$$|P_t(\mathbf{x},\mathbf{y}) - \pi(\mathbf{y})| \leq \pi(\mathbf{y}) \left[e^{2+\frac{\lambda}{\alpha} \log\log \frac{1}{\pi(*)}} \right] e^{-m\lambda t}. \tag{5.31}$$

The investigation and utilisation of the formulae (5.31) will be presented in the next subsection.

5.6.2 Investigation of the particular condition

For each n, let Q^n be the infinitesimal generator of the density dependent Markov chain $X_n(t)$ underlying a given PEPA model and thus the transition probability matrix is $P_t^n = e^{tQ^n}$. For each $X_n(t)$, the initial state of the corresponding system is $n\mathbf{x}_0$, so the initial probability distribution of $X_n(t)$ is $\pi_0^n(n\mathbf{x}_0) = 1$ and $\pi_0^n(\mathbf{x}) = 0,\ \forall \mathbf{x} \in S^n, \mathbf{x} \neq n\mathbf{x}_0$. So $\forall \mathbf{y} \in S^n$,

$$\pi_t^n(\mathbf{y}) = \sum_{\mathbf{x} \in S^n} \pi_0^n(\mathbf{x}) P_t^n(\mathbf{x},\mathbf{y}) = \pi_0^n(n\mathbf{x}_0) P_t^n(n\mathbf{x}_0,\mathbf{y}) = P_t^n(n\mathbf{x}_0,\mathbf{y}). \tag{5.32}$$

The formula (5.31) in the context of $X_n(t)$ is

$$|P_t^n(\mathbf{x},\mathbf{y}) - \pi_\infty^n(\mathbf{y})| \leq \pi_\infty^n(\mathbf{y}) \left[e^{2+\frac{\lambda_n}{\alpha_n} \log\log \frac{1}{\pi_\infty^n(*)}} \right] e^{-m_n\lambda_n t}, \tag{5.33}$$

where $\lambda_n, \lambda_n, m_n, \pi_\infty^n(*)$ are the respective parameters associated with $X_n(t)$.

Let $\mathbf{x} = n\mathbf{x}_0$ in (5.33), and notice $P_t^n(n\mathbf{x}_0,\mathbf{y}) = \pi_t^n(\mathbf{y})$ by (5.32), then we have

$$|\pi_t^n(\mathbf{y}) - \pi_\infty^n(\mathbf{y})| \leq \pi_\infty^n(\mathbf{y}) \left[e^{2+\frac{\lambda_n}{\alpha_n} \log\log \frac{1}{\pi_\infty^n(*)}} \right] e^{-m_n\lambda_n t}, \qquad \forall \mathbf{y} \in S^n. \tag{5.34}$$

From the comparison of (5.34) and (5.27), we know that if there are some conditions imposed on $\frac{\lambda_n}{\alpha_n} \log\log \frac{1}{\pi_\infty^n(*)}$ and $-m_n\lambda_n t$, then (5.27) can be induced from (5.34). See the following

Lemma.

Lemma 5.6.2. *If there exists $T > 0, B_2 > 0, A > 0$ such that*

$$\sup_n \left\{ -m_n \lambda_n T + \frac{\lambda_n}{\alpha_n} \log\log \frac{1}{\pi^n_\infty(*)} \right\} \leq B_2, \tag{5.35}$$

and

$$\inf_n \{m_n \lambda_n\} \geq A, \tag{5.36}$$

then

$$|\pi^n_t(\mathbf{x}) - \pi^n_\infty(\mathbf{x})| \leq \pi^n_\infty(\mathbf{x}) B_1 e^{-At}, \qquad \forall \mathbf{x} \in S^n, \tag{5.37}$$

where $B_1 = e^{A_1 T + B_2 + 2}$, and the "particular condition" (5.26) holds.

Proof.

$$\begin{aligned}
\left[e^{2 + \frac{\lambda_n}{\alpha_n} \log\log \frac{1}{\pi^n_\infty(*)}} \right] e^{-m_n \lambda_n t} &= \left[e^{-m_n \alpha_n T + 2 + \frac{\lambda_n}{\alpha_n} \log\log \frac{1}{\pi^n_\infty(*)}} \right] e^{-m_n \lambda_n (t-T)} \\
&\leq e^{B_2 + 2} e^{-A(t-T)} \\
&= B_1 e^{-At}.
\end{aligned}$$

Then by (5.34), (5.37) holds. Thus by Lemma 5.5.3, (5.26) holds. □

Remark 5.6.1. *When n tends to infinity, the discrete space is approximating a continuous space and thus the probability of any single point is tending to 0, that is $\pi^n_\infty(*)$ tends to 0. So $\log\log \frac{1}{\pi^n_\infty(*)} \to \infty$ as n goes to infinity. Notice that by Lemma C.2.1 in Appendix C.2,*

$$\frac{\lambda_n}{\alpha_n} \leq \log(1/\pi^n_\infty(*)).$$

Therefore, in order to have

$$-m_n \lambda_n T + \frac{\lambda_n}{\alpha_n} \log\log \frac{1}{\pi^n_\infty(*)} \leq B_2,$$

it is sufficient to let

$$T m_n \lambda_n \geq O([\log(1/\pi^n_\infty(*))]^2). \tag{5.38}$$

Moreover, (5.38) can imply both (5.35) and (5.36).

According to the above analysis, our problem is simplified to checking that whether (5.38) is

satisfied by the density dependent Markov chains $\{X_n(t)\}$.

By Remark C.2.2 in Appendix C.2, λ_n is the smallest non-zero eigenvalue of $I - \frac{K^n + K^{*,n}}{2}$, where

$$K^n = \frac{Q^n}{m_n} + I$$

and $K^{*,n}$ is adjoint to K^n. A matrix $Q^{*,n}$ is said to be adjoint to the generator Q^n, if $Q^n(\mathbf{x}, \mathbf{y})\pi^n_\infty(\mathbf{x})$ equals $Q^{*,n}(\mathbf{y}, \mathbf{x})\pi^n_\infty(\mathbf{y})$. Clearly, $Q^{*,n} = m_n(K^{*,n} - I)$, or equivalently,

$$K^{*,n} = \frac{Q^{*,n}}{m_n} + I.$$

So

$$m_n \left(I - \frac{K^n + K^{*,n}}{2} \right) = -\frac{Q^n + Q^{*,n}}{2}. \tag{5.39}$$

Denote the smallest non-zero eigenvalue of $-\frac{Q^n + Q^{*,n}}{2}$ by σ_n. Then by (5.39),

$$m_n \lambda_n = \sigma_n. \tag{5.40}$$

Now we state our main result in this chapter.

Theorem 5.6.3. *Let $\{X_n(t)\}$ be the density dependent Markov chain derived from a given PEPA model. For each $n \in \mathbb{N}$, let S^n and π^n_∞ be the state space and steady-state probability distribution of $X_n(t)$ respectively. Q^n is the infinitesimal generator of $X_n(t)$ and σ_n is the smallest non-zero eigenvalue of $-\frac{Q^n + Q^{*,n}}{2}$, where $Q^{*,n}$ is adjoint to Q^n in terms of π^n_∞. If*

$$\pi^n_\infty(*) \stackrel{\text{def}}{=} \min_{\mathbf{x} \in S^n} \pi^n_\infty(\mathbf{x}) \geq \frac{1}{\exp(O(\sqrt{\sigma_n}))} \tag{5.41}$$

for sufficiently large n, then $X(t)$ has a finite limit as time tends to infinity, where $X(t)$ is the solution of the corresponding derived ODEs from the same PEPA model.

Proof. By the given condition of $\pi^n_\infty(*) \geq \frac{1}{\exp(O(\sqrt{\sigma_n}))}$,

$$\log \left[\frac{1}{\pi^n_\infty(*)} \right] \leq \log[\exp(O(\sqrt{\sigma_n}))] = O(\sigma_n^{1/2}).$$

Thus

$$\left(\log \left[\frac{1}{\pi^n_\infty(*)} \right] \right)^2 \leq O(\sigma_n).$$

Choose a large T such that

$$Tm_n\lambda_n = T\sigma_n \geq O(\sigma_n) \geq \left(\log\left[\frac{1}{\pi^n_\infty(*)}\right]\right)^2.$$

By Remark 5.6.1 and Lemma 5.6.2, the particular condition holds. Therefore

$$\lim_{t\to\infty} X(t) = \hat{M}_\infty(\infty).$$

□

According to the above theorem, our problem is simplified to checking that whether (5.41) is satisfied by the density dependent Markov chains $\{X_n(t)\}$. In (5.41) both the spectral gap λ_n $(= \frac{\sigma_n}{m_n})$ and $\pi^n_\infty(*)$ are unknown. In fact, due to the state space explosion problem, π^n_∞ cannot be easily solved from $\pi^n_\infty Q^n = 0$ or equivalently $\pi^n_\infty K^n = \pi^n_\infty$. Moreover, the estimation of the spectral gap in current literature, for a given Markov chain, is heavily based on the known stationary distribution, see Theorem C.2.2 and Theorem C.2.3 in Appendix C.2. Thus, these current results cannot provide a practical check for (5.41).

The convergence and consistency are supported by many numerical experiments (see the case study in Chapter 7), so we believe that (5.41) is unnecessary. In other words, we believe (5.41) always holds in the context of PEPA, although at this moment we cannot prove it. Before concluding this section, we leave an open problem:

Conjecture 1. *The formula (5.41) in Theorem 5.6.3 holds for any PEPA model.*

5.7 Summary

This chapter has presented the semantics of mapping general PEPA models to ODEs, which generalises the results of fluid approximations for PEPA in current literature. The fundamental characteristics of the solutions of the derived ODEs, such as existence, uniqueness, nonnegativeness, boundedness, convergence, as well as the consistency with the underlying Markov chains, have been established in this chapter. For convenience, we organise them into the following Table 5.5.

In particular, for nonsynchronised PEPA models, as time goes to infinity the solutions of the derived ODEs have finite limits, which coincide with the stationary probability distributions

.

	Existence & Uniqueness	Boundedness & Nonnegativeness	Convergence	Consistency
nonsyn. case	Yes	Yes	Yes	Yes
syn. case	Yes	Yes	Yes under a cond.	Yes

Table 5.5: *Fundamental characteristics of derived ODEs from PEPA models*

of the underlying CTMCs. As for PEPA models with synchronisations, the solutions converge under a particular condition. If they converge, the limits are consistent with the stationary distributions of a family of corresponding density dependent Markov chains underlying the models. The main approaches of investigating these properties are probabilistic. The next chapter will present our further study on the convergence problem using purely analytical methods, focussing on some classes of synchronised PEPA models.

Chapter 6
Fluid Analysis for Large Scale PEPA Models—Part II: Analytical Approach

6.1 Introduction

The previous chapter has demonstrated the fluid approximation and relevant analysis for PEPA. Some fundamental results about the derived ODEs such as the boundedness, nonnegativeness and convergence of the solutions, have been established through a probabilistic approach. In this chapter we will discuss the boundedness and nonnegativeness again, and prove them by a purely analytical argument. The convergence presented in the previous chapter is proved under a particular condition that cannot currently be easily checked. This chapter will present alternative approaches to deal with the convergence problem. In particular, for an interesting model with two synchronisations, its structural invariance as revealed in Chapter 4, will be shown to play an important role in the proof of the convergence. Moreover, for a class of PEPA models which have two component types and one synchronisation, an analytical proof of the convergence under some mild conditions on the populations will be presented. These discussions and investigations will provide new insight into the fluid approximation of PEPA.

The remainder of this chapter is structured as follows. Section 2 gives a purely analytical proof for the boundedness and nonnegativeness of the solutions of the ODEs derived from general PEPA models. A case study on convergence for a model with two synchronisations will be shown in Section 3. In Sections 4 and 5, we present proofs of convergence for PEPA models with two component types and one synchronisation. Finally, Section 6 concludes this chapter.

6.2 Analytical Proof of Boundedness and Nonnegativeness

Recall that the set of derived ODEs from a general PEPA model is

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \sum_{l \in \mathcal{A}_{\mathrm{label}}} l f(\mathbf{x}, l). \tag{6.1}$$

In the previous chapter we have proved that the solution of (6.1) is bounded as well as nonnegative. This section will present a new analytical proof for these characteristics.

6.2.1 Features of the derived ODEs

Let U be a local derivative, then from (6.1) we have

$$\frac{\mathrm{d}\mathbf{x}(U,t)}{\mathrm{d}t} = -\sum_{\{l|U\in \mathrm{pre}(l)\}} f(\mathbf{x},l) + \sum_{\{l|U\in \mathrm{post}(l)\}} f(\mathbf{x},l). \tag{6.2}$$

As mentioned in Chapter 5, in this formula the term $\sum_{\{l|U\in \mathrm{pre}(l)\}} f(\mathbf{x},l)$ represents the exit rates in the local derivative U, while the term $\sum_{\{l|U\in \mathrm{post}(l)\}} f(\mathbf{x},l)$ indicates the entry rates to U. We have noted an important fact: all exit rates in a component type are balanced by all entry rates. That is Proposition 5.2.1 in Chapter 5. Below, for convenience, this proposition is stated again:

Proposition 5.2.1. *Let C_{i_j} be the local derivatives of component type C_i. Then for any i and t, $\sum_j \frac{\mathrm{d}\mathbf{x}\left(C_{i_j},t\right)}{\mathrm{d}t} = 0$, and $\sum_j \mathbf{x}\left(C_{i_j},t\right) = \sum_j \mathbf{x}\left(C_{i_j},0\right)$.*

This proposition states that the ODEs (6.1) satisfy a conservation law: at any time the population of each component type is constant, since the system is closed and there is no exchange with the environment.

Another important fact to note is: the exit rates in a local derivative C_{i_j} in state $\mathbf{x}$ are bounded by all the apparent rates in this local derivative. In fact, according to Proposition 3.4.2 in Chapter 3, if $C_{i_j} \in \mathrm{pre}(l)$ where l is a labelled activity, then the transition rate function $f(\mathbf{x},l)$ is bounded by $r_l(\mathbf{x},C_{i_j})$, the apparent rates of l in C_{i_j} in state $\mathbf{x}$. That is, $f(\mathbf{x},l) \le r_l(\mathbf{x},C_{i_j})$. Notice $r_l(\mathbf{x},C_{i_j}) = \mathbf{x}[C_{i_j}]r_l(C_{i_j})$ by Definition 3.4.1 in Chapter 3, where $r_l(C_{i_j})$ represents the apparent rate for a single instance of C_{i_j}. Thus, we have

$$f(\mathbf{x},l) \le \mathbf{x}[C_{i_j}]r_l(C_{i_j}). \tag{6.3}$$

We should point out that (6.3) is based on the discrete state space underlying the given model. According to our semantics of mapping PEPA model to ODEs, the fluid approximation-version of (6.3) also holds, i.e. $f(\mathbf{x}(t),l) \le \mathbf{x}(C_{i_j},t)r_l(C_{i_j})$. Therefore, we have the following

Proposition 6.2.1. *For any local derivative C_{i_j},*

$$\sum_{\{l|C_{i_j}\in \mathrm{pre}(l)\}} f(\mathbf{x}(t),l) \leq \mathbf{x}(C_{i_j},t) \sum_{\{l|C_{i_j}\in \mathrm{pre}(l)\}} r_l(C_{i_j}), \tag{6.4}$$

where $r_l(C_{i_j})$ is the apparent rate of l in C_{i_j} for a single instance of C_{i_j} defined in Definition 3.4.1.

Propositions 5.2.1 and 6.2.1 are two important characteristics of the ODEs derived from PEPA models, which can guarantee the boundedness and nonnegativeness of the solutions.

6.2.2 Boundedness and nonnegativeness of solutions

We know that the solution of (6.1) exists and is unique. Furthermore, the solution is bounded and nonnegative, which has been proved in the previous chapter through an approximation approach. This subsection will present an analytical proof of these properties, based on the two propositions described in the previous subsection.

Suppose the initial values $\mathbf{x}\left(C_{i_j},0\right)$ are given, and we denote $\sum_j \mathbf{x}\left(C_{i_j},0\right)$ by N_{C_i}. We have a theorem:

Theorem 6.2.1. *If $\mathbf{x}\left(C_{i_j},t\right)$ satisfies (6.1) with nonnegative initial values, then*

$$0 \leq \mathbf{x}\left(C_{i_j},t\right) \leq N_{C_i}, \quad \textit{for any } t \geq 0. \tag{6.5}$$

Moreover, if the initial values are positive, then the solutions are always positive, i.e.,

$$0 < \mathbf{x}\left(C_{i_j},t\right) < N_{C_i}, \quad \textit{for any } t \geq 0. \tag{6.6}$$

Proof. By Proposition 5.2.1, $\sum_j \mathbf{x}\left(C_{i_j},t\right) = N_{C_i}$ for all t. All that is left to do is to prove that $\mathbf{x}(C_{i_j},t)$ is positive or nonnegative. The proof is divided into two cases.

Case 1: Suppose all the initial values are positive, i.e. $\min_{i_j}\left\{\mathbf{x}(C_{i_j},0)\right\} > 0$. We will show that $\min_{i_j}\left\{\mathbf{x}(C_{i_j},t)\right\} > 0$ for all $t \geq 0$. Otherwise, if there exists a $t > 0$ such that $\min_{i_j}\left\{\mathbf{x}(C_{i_j},t)\right\} \leq 0$, then there exists a point $t' > 0$ such that $\min_{i_j}\left\{\mathbf{x}(C_{i_j},t')\right\} = 0$. Let

t^* be the first such point, i.e.

$$t^* = \inf\left\{t > 0 \mid \min_{i_j}\left\{\mathbf{x}(C_{i_j}, t)\right\} = 0\right\},$$

then $0 < t^* < \infty$. Without loss of generality, we assume $\mathbf{x}(C_{1_1}, t)$ reaches zero at t^*, i.e.,

$$\mathbf{x}(C_{1_1}, t^*) = 0, \quad \mathbf{x}(C_{i_j}, t^*) \geq 0 \ \ (i \neq 1 \vee j \neq 1)$$

and

$$\mathbf{x}(C_{i_j}, t) > 0, \quad t \in [0, t^*), \quad \forall i, j.$$

Notice that the transition rate function is nonnegative (see Proposition 3.4.2), i.e. $f(\mathbf{x}, l) \geq 0$. Then for $t \in [0, t^*]$, by Proposition 6.2.1,

$$\begin{aligned}
\frac{\mathrm{d}\mathbf{x}(C_{1_1}, t)}{\mathrm{d}t} &= -\sum_{\{l|C_{1_1} \in \mathrm{pre}(l)\}} f(\mathbf{x}, l) + \sum_{\{l|C_{1_1} \in \mathrm{post}(l)\}} f(\mathbf{x}, l) \\
&\geq -\sum_{\{l|C_{1_1} \in \mathrm{pre}(l)\}} f(\mathbf{x}, l) \\
&\geq -\mathbf{x}(C_{1_1}, t) \sum_{\{l|C_{1_1} \in \mathrm{pre}(l)\}} r_l(C_{1_1}).
\end{aligned}$$

Set $R = \sum_{\{l|C_{1_1} \in \mathrm{pre}(l)\}} r_l(C_{1_1})$, then

$$\frac{\mathrm{d}\mathbf{x}(C_{1_1}, t)}{\mathrm{d}t} \geq -R\mathbf{x}(C_{1_1}, t). \tag{6.7}$$

By Lemma D.1.1 in Appendix D.1, (6.7) implies

$$\mathbf{x}(C_{1_1}, t^*) \geq \mathbf{x}(C_{1_1}, 0)e^{-Rt^*} > 0.$$

This is a contradiction to $\mathbf{x}(C_{1_1}, t^*) = 0$. Therefore $0 < \mathbf{x}\left(C_{i_j}, t\right)$, and thus by Proposition 5.2.1,

$$0 < \mathbf{x}\left(C_{i_j}, t\right) < N_{C_0}, \quad \forall t.$$

Case 2: Suppose $\min_{i_j}\left\{\mathbf{x}(C_{i_j}, 0)\right\} = 0$. Let $u_\delta(i_j, 0) = \mathbf{x}(C_{i_j}, 0) + \delta$ where $\delta > 0$. Let $u_\delta(i_j, t)$ be the solution of (6.1), given the initial value $u_\delta(i_j, 0)$. By the proof of Case 1, $u_\delta(i_j, t) > 0$ $(\forall t \geq 0)$. Noticing $\min(\cdot)$ is a Lipschitz function, by the Fundamental Inequality

(Theorem D.1.2) in Appendix D.1, we have

$$|u_\delta(i_j,t) - \mathbf{x}(C_{i_j},t)| \leq \delta e^{Kt}, \tag{6.8}$$

where K is a constant. Thus, for any given $t \geq 0$,

$$\mathbf{x}(C_{i_j},t) \geq u_\delta(i_j,t) - \delta e^{Kt} > -\delta e^{Kt}. \tag{6.9}$$

Let $\delta \downarrow 0$ in (6.9), then we have $\mathbf{x}(C_{i_j},t) \geq 0$. The proof is completed. □

6.3 A Case Study on Convergence with Two Synchronisations

If a model has synchronisations, then the derived ODEs are nonlinear. The nonlinearity results in the complexity of the dynamic behaviour of fluid approximations. However, for some special models, we can still determine the convergence of the solutions. What follows is a case study for an interesting PEPA model, in which the structural property of invariance will be shown to play an important role in the proof of the convergence.

6.3.1 ODEs derived from an interesting model

The model considered in this section is given below, which is Model 3 presented in Chapter 4.

$$\begin{aligned}
X_1 &\stackrel{def}{=} (action1, a_1).X_2 \\
X_2 &\stackrel{def}{=} (action2, a_2).X_1 \\
Y_1 &\stackrel{def}{=} (action1, a_1).Y_3 + (job1, c_1).Y_2 \\
Y_2 &\stackrel{def}{=} (job2, c_2).Y_1 \\
Y_3 &\stackrel{def}{=} (job3, c_3).Y_4 \\
Y_4 &\stackrel{def}{=} (action2, a_2).Y_2 + (job4, c_4).Y_3 \\
(X_1[M_1]||X_2[M_2]) &\underset{\{action1,action2\}}{\bowtie} (Y_1[N_1]||Y_2[N_2]||Y_3[N_3]||Y_4[N_4]).
\end{aligned}$$

The operations of X and Y are illustrated in Figure 6.1. According to the mapping semantics, the derived ODEs from this model are

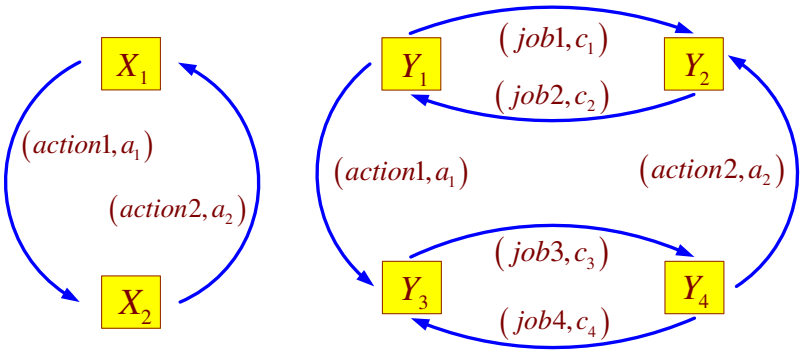


Figure 6.1: *Transition systems of the components of Model 3*

$$\begin{cases} \dfrac{\mathrm{d}x_1}{\mathrm{d}t} = -a_1 \min\{x_1, y_1\} + a_2 \min\{x_2, y_4\} \\ \dfrac{\mathrm{d}x_2}{\mathrm{d}t} = a_1 \min\{x_1, y_1\} - a_2 \min\{x_2, y_4\} \\ \dfrac{\mathrm{d}y_1}{\mathrm{d}t} = -a_1 \min\{x_1, y_1\} - c_1 y_1 + c_2 y_2 \\ \dfrac{\mathrm{d}y_2}{\mathrm{d}t} = -c_2 y_2 + a_2 \min\{x_2, y_4\} + c_1 y_1 \\ \dfrac{\mathrm{d}y_3}{\mathrm{d}t} = -c_3 y_3 + c_4 y_4 + a_1 \min\{x_1, y_1\} \\ \dfrac{\mathrm{d}y_4}{\mathrm{d}t} = -a_2 \min\{x_2, y_4\} - c_4 y_4 + c_3 y_3 \end{cases}, \tag{6.10}$$

where x_i, y_j $(i = 1, 2,\ j = 1, 2, \cdots, 4)$ denote the populations of X and Y in the local derivatives X_i, Y_j respectively. Throughout this section, let M and N be the total populations of the X and Y respectively, i.e. $M = M_1 + M_2$ and $N = N_1 + N_2 + N_3 + N_4$.

In Chapter 4 we have revealed an invariant in this model, i.e., $y_3 + y_4 - x_2$ is a constant in any state. Notice $y_1 + y_2 + y_3 + y_4 = N$ and $x_1 + x_2 = M$ by the conservation law, so $y_1 + y_2 - x_1 = N - M - (y_3 + y_4 - x_2)$ is another invariant because $y_3 + y_4 - x_2$ is a constant. The fluid-approximation version of these two invariants also holds, which is illustrated by the following Lemma 6.3.1.

Lemma 6.3.1. *For any $t \geq 0$,*

$$y_1(t) + y_2(t) - x_1(t) = y_1(0) + y_2(0) - x_1(0),$$
$$y_3(t) + y_4(t) - x_2(t) = y_3(0) + y_4(0) - x_2(0).$$

Proof. According to (6.10), for any $t \geq 0$,

$$\frac{\mathrm{d}(y_1(t) + y_2(t) - x_1(t))}{\mathrm{d}t} = \frac{\mathrm{d}y_1}{\mathrm{d}t} + \frac{\mathrm{d}y_2}{\mathrm{d}t} - \frac{\mathrm{d}x_1}{\mathrm{d}t} = 0.$$

So $y_1(t) + y_2(t) - x_1(t) = y_1(0) + y_2(0) - x_1(0), \forall t \geq 0$. By a similar argument, we also have $y_3(t) + y_4(t) - x_2(t) = y_3(0) + y_4(0) - x_2(0), \forall t \geq 0$. □

In the following we show how to use this kind of invariance to prove the convergence of the solution of (6.10) as time goes to infinity. Before presenting the results and the proof, we first rewrite (6.10) as follows:

$$\begin{aligned}\begin{pmatrix} \frac{\mathrm{d}x_1}{\mathrm{d}t} \\ \frac{\mathrm{d}x_2}{\mathrm{d}t} \\ \frac{\mathrm{d}y_1}{\mathrm{d}t} \\ \frac{\mathrm{d}y_2}{\mathrm{d}dt} \\ \frac{\mathrm{d}y_3}{\mathrm{d}t} \\ \frac{\mathrm{d}y_4}{\mathrm{d}t} \end{pmatrix} =& I_{\{x_1<y_1, x_2<y_4\}} Q_1 \begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} + I_{\{x_1<y_1, x_2\geq y_4\}} Q_2 \begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} \\ &+ I_{\{x_1\geq y_1, x_2<y_4\}} Q_3 \begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} + I_{\{x_1\geq y_1, x_2\geq y_4\}} Q_4 \begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix},\end{aligned} \tag{6.11}$$

where the matrices Q_i $(i = 1, 2, 3, 4)$ are given as below:

$$Q_1 = \left(\begin{array}{cc|cccc} -a_1 & a_2 & & & & \\ a_1 & -a_2 & & & & \\ \hline -a_1 & & -c_1 & c_2 & & \\ & a_2 & c_1 & -c_2 & & \\ a_1 & & & & -c_3 & c_4 \\ & -a_2 & & & c_3 & -c_4 \end{array}\right),$$

$$Q_2 = \left(\begin{array}{cc|cccc} -a_1 & 0 & & & & a_2 \\ a_1 & 0 & & & & -a_2 \\ \hline -a_1 & 0 & -c_1 & c_2 & & \\ & 0 & c_1 & -c_2 & & a_2 \\ a_1 & 0 & & & -c_3 & c_4 \\ & 0 & & & c_3 & -(c_4+a_2) \end{array}\right),$$

$$Q_3 = \left(\begin{array}{cc|cccc} 0 & a_2 & -a_1 & & & \\ 0 & -a_2 & a_1 & & & \\ \hline 0 & & -a_1-c_1 & c_2 & & \\ 0 & a_2 & c_1 & -c_2 & & \\ 0 & & a_1 & & -c_3 & c_4 \\ 0 & -a_2 & & & c_3 & -c_4 \end{array}\right),$$

$$Q_4 = \left(\begin{array}{cc|cccc} 0 & 0 & -a_1 & & & a_2 \\ 0 & 0 & a_1 & & & -a_2 \\ \hline 0 & 0 & -a_1-c_1 & c_2 & & \\ 0 & 0 & c_1 & -c_2 & & a_2 \\ 0 & 0 & a_1 & & -c_3 & c_4 \\ 0 & 0 & & & c_3 & -a_2-c_4 \end{array}\right).$$

As (6.11) illustrates, the derived ODEs are piecewise linear and they may be dominated by Q_i $(i = 1, 2, 3, 4)$ alternately. If the system is always dominated by only one specific matrix after a time, then the ODEs become linear after this time. For linear ODEs, as long as the eigenvalues of their coefficient matrices are either zeros or have negative real parts, then bounded solutions will converge as time tends to infinity, see Corollary D.2.3 in Appendix D.2. Fortunately, here the eigenvalues of the matrices Q_i $(i = 1, 2, 3, 4)$ in Model 3 satisfy this property, the proof of which is shown in Appendix D.3. In addition, the solution of the derived ODEs from any PEPA model is bounded, as Theorem 6.2.1 illustrated. Therefore, if we can guarantee that after a time the ODEs (6.11) become linear, which means that one of the four matrices Q_i $(i = 1, 2, 3, 4)$ will be the coefficient matrix of the linear ODEs, then by Corollary D.2.3 the solution will converge. So the convergence problem is reduced to determining whether the

linearity can be finally guaranteed.

It is easy to see that the comparisons between x_1 and y_1, x_2 and y_4 determine the linearity. For instance, if after a time T, we always have $x_1 > y_1$ and $x_2 > y_4$, then the matrix Q_4 will dominate the system. Fortunately, the invariance in the model, as Lemma 6.3.1 reveals, can determine the comparisons in some circumstances. This is because this invariance reflects the relationship between different component types that are connected through synchronisations. This leads to several conclusions as follows.

Proposition 6.3.1. *If $y_1(0) + y_2(0) \leq x_1(0)$ and $y_3(0) + y_4(0) \leq x_2(0)$, then the solution of (6.11) converges.*

Proof. By Lemma 6.3.1, $y_1(t) + y_2(t) \leq x_1(t)$ and $y_3(t) + y_4(t) \leq x_2(t)$ for all time t. Since both $y_2(t)$ and $y_4(t)$ are nonnegative by Theorem 6.2.1, we have $y_1(t) \leq x_1(t)$ and $y_4(t) \leq x_2(t)$ for any t. Thus, (6.11) becomes

$$\begin{pmatrix} \frac{\mathrm{d}x_1}{\mathrm{d}t} \\ \frac{\mathrm{d}x_2}{\mathrm{d}t} \\ \frac{\mathrm{d}y_1}{\mathrm{d}t} \\ \frac{\mathrm{d}y_2}{\mathrm{d}t} \\ \frac{\mathrm{d}y_3}{\mathrm{d}t} \\ \frac{\mathrm{d}y_4}{\mathrm{d}t} \end{pmatrix} = Q_4 \begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}. \tag{6.12}$$

Notice that (6.12) is linear, and all eigenvalues of Q_4 other than zeros have negative real parts, then according to Corollary D.2.3, the solution of (6.12) converges as time goes to infinity. □

Proposition 6.3.2. *Suppose $y_1(0) + y_2(0) > x_1(0)$ and $y_3(0) + y_4(0) \leq x_2(0)$. If either (I). $N > \left(2 + \dfrac{a_1 + c_1}{c_2}\right) M$, or (II). $N > \dfrac{2(c_1 + c_2) + a_2}{c_2} M$, where $N > M > 0$ are the populations of Y and X respectively, then the solution of (6.11) converges.*

Proof. Suppose (I) holds. According to the conservation law, $\sum_{i=1}^{4} y_i(t) = N$. By the boundedness of the solution, we have $x_2(t) \leq M$. Then by Lemma 6.3.1, $y_3(t)+y_4(t) \leq x_2(t) \leq M$. Therefore,

$$y_2(t) = N - (y_3(t) + y_4(t)) - y_1(t) \geq N - M - y_1(t). \tag{6.13}$$

Since $\min\{x_1, y_1\} \leq y_1$, so $-a\min\{x_1, y_1\} \geq -a_1 y_1$. Thus

$$\begin{aligned}\frac{\mathrm{d}y_1}{\mathrm{d}t} &= -a_1 \min\{x_1, y_1\} - c_1 y_1 + c_2 y_2 \\ &\geq -a_1 y_1 - c_1 y_1 + c_2 y_2 \\ &\geq -(a_1 + c_1) y_1 + c_2 (N - M - y_1) \\ &= -(a_1 + c_1 + c_2) y_1 + c_2 (N - M).\end{aligned} \tag{6.14}$$

That is

$$\frac{\mathrm{d}y_1}{\mathrm{d}t} \geq -(a_1 + c_1 + c_2) y_1 + c_2 (N - M).$$

Applying Lemma D.1.1 in Appendix D.1 to this formula, we have

$$y_1(t) \geq \left(y_1(0) - \frac{c_2(N-M)}{a_1 + c_1 + c_2} \right) e^{-(a_1+c_1+c_2)t} + \frac{c_2(N-M)}{a_1 + c_1 + c_2}. \tag{6.15}$$

Since the first term of the right side of (6.15) converges to zero as time goes to infinity, i.e. $\lim_{t\to\infty} \left(y_1(0) - \frac{c_2(N-M)}{a_1 + c_1 + c_2} \right) e^{-(a_1+c_1+c_2)t} = 0$, and the second term $\frac{c_2(N-M)}{a_1 + c_1 + c_2} > M$ which results from the condition $N > \left(2 + \frac{a_1 + c_1}{c_2} \right) M$, then there exists $T > 0$ such that for any $t > T$, $y_1(t) > M \geq x_1(t)$. Then after time T, (6.11) becomes linear, and is dominated by Q_2. Because all eigenvalues of Q_2 are either zeros or have negative real parts, the solution converges as time goes to infinity.

Now we assume (II) holds. Similarly, since $\min\{x_2, y_4\} \leq x_2 \leq M$, and $y_1 \leq N - y_2$ which is due to $y_1 + y_2 \leq N$, we have

$$\begin{aligned}\frac{\mathrm{d}y_2}{\mathrm{d}t} &= -c_2 y_2 + a_2 \min\{x_2, y_4\} + c_1 y_1 \\ &\leq -c_2 y_2 + a_2 M + c_1 y_1 \\ &\leq -c_2 y_2 + a_2 M + c_1 (N - y_2) \\ &= -(c_1 + c_2) y_2 + a_2 M + c_1 N.\end{aligned} \tag{6.16}$$

By Lemma D.1.1 in Appendix D.1,

$$y_2 \leq \left(y_2(0) - \frac{a_2 M + c_1 N}{c_1 + c_2} \right) e^{-(c_1+c_2)t} + \frac{a_2 M + c_1 N}{c_1 + c_2}. \tag{6.17}$$

Therefore, since $e^{-(c_1+c_2)t}$ in above formula converges to zero as time tends to infinity, then

for any $\epsilon > 0$, there exists $T > 0$ such that for any time $t > T$,

$$y_2 \leq \frac{a_2 M + c_1 N}{c_1 + c_2} + \epsilon. \tag{6.18}$$

Notice that the condition $N > \dfrac{2(c_1 + c_2) + a_2}{c_2} M$ implies

$$\frac{c_2 N - a_2 M - (c_1 + c_2) M}{c_1 + c_2} > M,$$

and let ϵ be small enough that $\dfrac{c_2 N - a_2 M - (c_1 + c_2) M}{c_1 + c_2} - \epsilon > M$. Then by (6.13), $y_1 \geq (N - M) - y_2$. Therefore,

$$\begin{aligned} y_1 &\geq (N - M) - y_2 \\ &\geq (N - M) - \frac{a_2 M + c_1 N}{c_1 + c_2} - \epsilon \\ &= \frac{c_2 N - a_2 M - (c_1 + c_2) M}{c_1 + c_2} - \epsilon \\ &> M \geq x_1. \end{aligned} \tag{6.19}$$

So $y_1(t) > x_1(t)$, $y_4(t) \leq x_2(t)$, for any $t > T$, then by a similar argument the solution of (6.11) converges. □

Both condition (I) and (II) in Proposition 6.3.2 require N to be sufficiently larger than M, to guarantee that y_1 is larger than x_1. Since our PEPA model is symmetric, Proposition 6.3.2 has a corresponding symmetric version.

Proposition 6.3.3. *Suppose $y_1(0) + y_2(0) \leq x_1(0)$ and $y_3(0) + y_4(0) > x_2(0)$. If either (I). $N > \left(2 + \dfrac{a_2 + c_3}{c_4}\right) M$, or (II). $N > \dfrac{2(c_3 + c_4) + a_1}{c_1} M$, where $N > M > 0$ are the populations of Y and X respectively, then the solution of (6.11) converges.*

The proof of Proposition 6.3.3 is omitted here. We should point out that in our model the shared activity $action1$ (respectively, $action2$) has the same local rate a_1 (respectively, a_2). We have taken advantage of this in the above proofs. If the local rates of shared activities are not set to be the same, analogous conclusions can still hold but the discussion will be more complicated. However, the structural property of invariance can still play an important role.

The above three propositions have illustrated the convergence for all situations in terms of the

starting state, except for the case of $y_1(0) + y_2(0) > x_1(0), y_3(0) + y_4(0) > x_2(0)$. See a summary in Table 6.1. If $y_1(0) + y_2(0) > x_1(0)$, $y_3(0) + y_4(0) > x_2(0)$, then the dynamic behaviour of the system is rather complex. A numerical study for this case will be presented in the next subsection.

Starting state condition	Additional condition	Conclusion
$y_1(0) + y_2(0) \leq x_1(0), y_3(0) + y_4(0) \leq x_2(0)$	No	Proposition 6.3.1
$y_1(0) + y_2(0) > x_1(0), y_3(0) + y_4(0) \leq x_2(0)$	$N > k_1 M$	Proposition 6.3.2
$y_1(0) + y_2(0) \leq x_1(0), y_3(0) + y_4(0) > x_2(0)$	$N > k_2 M$	Proposition 6.3.3
$y_1(0) + y_2(0) > x_1(0), y_3(0) + y_4(0) > x_2(0)$	None identified	Explored numerically

Table 6.1: *A summary for the convergence of Model 3*

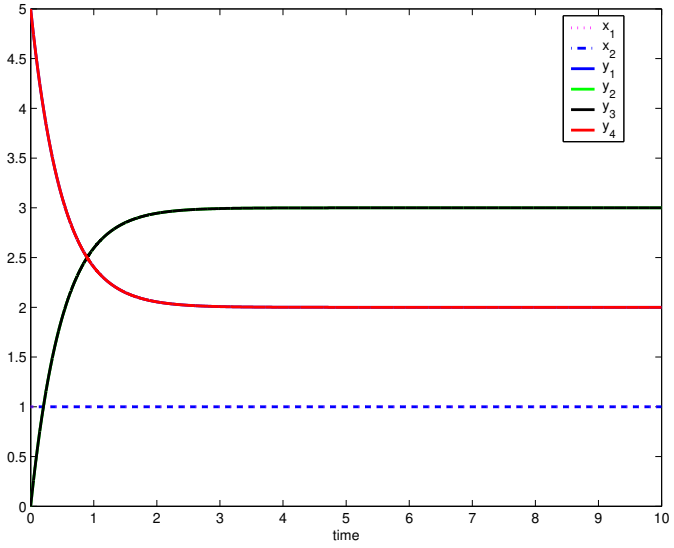


Figure 6.2: *Numerical study for Model 3: rates* $(1, 1, 1, 1, 1, 1)$*; equilibrium point* $(1, 1, 2, 3, 3, 2)$*. (Note: the curves of* x_1 *and* x_2*, the curves of* y_1 *and* y_4*, as well as those of* y_2 *and* y_3 *respectively, completely overlap.)*

6.3.2 Numerical study for convergence

This subsection presents a numerical study at different action rate conditions. The starting state in this subsection is always assumed as $(1, 1, 5, 0, 0, 5)$, which satisfies the condition of $y_1(0) + y_2(0) > x_1(0)$ and $y_3(0) + y_4(0) > x_2(0)$.

If all the action rates in the model are set to one, i.e. $(a_1, a_2, c_1, c_2, c_3, c_4) = (1, 1, 1, 1, 1, 1)$, then the equilibrium point of the ODEs is $(x_1^*, x_2^*, y_1^*, y_2^*, y_3^*, y_4^*) = (1, 1, 2, 3, 3, 2)$, as the

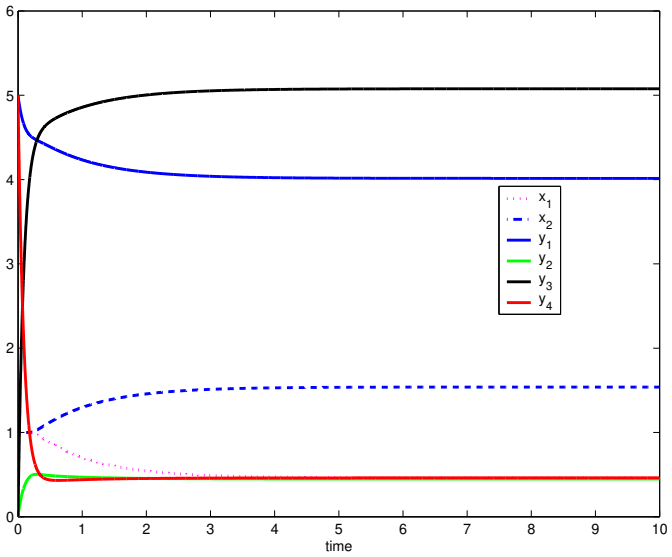


Figure 6.3: *Numerical study for Model 3: rates* $(1, 1, 1, 10, 1, 10)$*; equilibrium point* $(0.4616, 1.5384, 4.0140, 0.4476, 5.0769, 0.4615)$

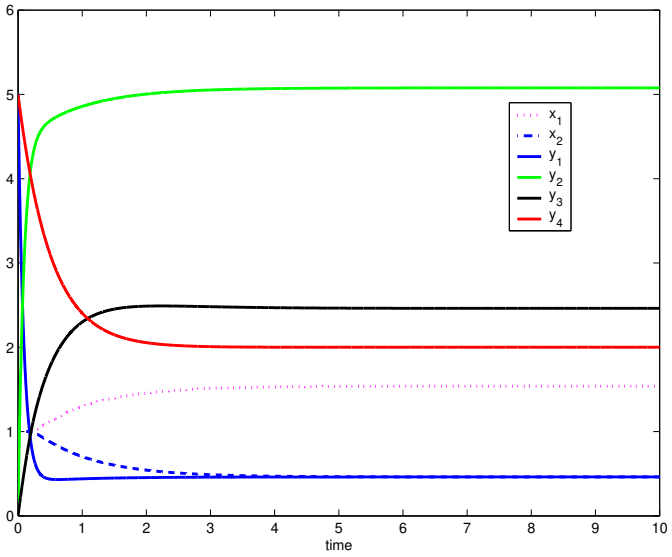


Figure 6.4: *Numerical study for Model 3: rates* $(1, 1, 10, 1, 1, 1)$*; equilibrium point* $(1.5384, 0.4616, 0.4615, 5.0769, 2.4616, 2.0000)$

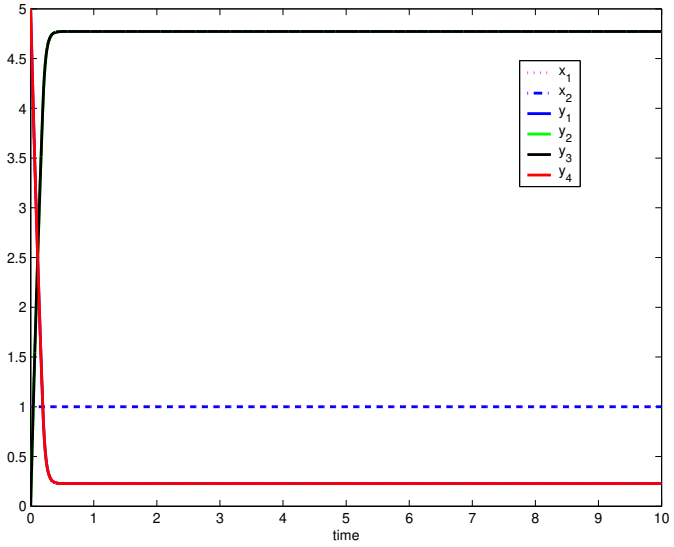


Figure 6.5: *Numerical study for Model 3: rates* $(20, 20, 1, 1, 1, 1)$*; equilibrium point* $(1, 1, 0.2273, 4.7727, 4.7727, 0.2273)$*. (Note: the curves of* x_1 *and* x_2*, the curves of* y_1 *and* y_4*, as well as those of* y_2 *and* y_3 *respectively, completely overlap.)*

numerical solution of the ODEs illustrates. In this case, the matrix Q_1 finally dominates the system. See Figure 6.2. Notice that in this figure, the curves of x_1 and x_2 completely overlap, as well as the curves of y_1 and y_4, and those of y_2 and y_3.

In other situations, for example, if $(a_1, a_2, c_1, c_2, c_3, c_4) = (1, 1, 1, 10, 1, 10)$ then the equilibrium point is $(0.4616, 1.5384, 4.0140, 0.4476, 5.0769, 0.4615)$, and the matrix Q_2 eventually dominates the system. See Figure 6.3. Moreover, the matrices Q_3 and Q_4 can also finally dominate the system as long as the action rates are appropriately specified. See Figure 6.4 and Figure 6.5. We should point out that similarly to Figure 6.2, in Figure 6.5 the curves of x_1 and x_2, the curves of y_1 and y_4, as well as those of y_2 and y_3 respectively, completely overlap.

In short, the system dynamics is rather complex in the situation of $y_1(0) + y_2(0) > x_1(0)$ and $y_3(0) + y_4(0) > x_2(0)$. A summary of these numerical studies is organised in Table 6.2.

Rates: $(a_1, a_2, c_1, c_2, c_3, c_4)$	Equilibrium points: $(x_1^*, x_2^*, y_1^*, y_2^*, y_3^*, y_4^*)$	Dominator	Figure
$(1, 1, 1, 1, 1, 1)$	$(1, 1, 2, 3, 3, 2)$	Q_1	Figure 6.2
$(1, 1, 1, 10, 1, 10)$	$(0.4616, 1.5384, 4.0140, 0.4476, 5.0769, 0.4615)$	Q_2	Figure 6.3
$(1, 1, 10, 1, 1, 1)$	$(1.5384, 0.4616, 0.4615, 5.0769, 2.4616, 2.0000)$	Q_3	Figure 6.4
$(20, 20, 1, 1, 1, 1)$	$(1, 1, 0.2273, 4.7727, 4.7727, 0.2273)$	Q_4	Figure 6.5

Table 6.2: *Complex dynamical behaviour of Model 3: starting state (1,1,5,0,0,5)*

6.4 Convergence For Two Component Types and One Synchronisation (I): A Special Case

The problem of convergence for more general models without strict conditions, is rather complex and has not been completely solved. But for a particular class of PEPA model — a model composed of two types of component with one synchronisation between them, we can determine the convergence of the solutions of the derived ODEs.

As discussed in the previous section, the ODEs derived from PEPA are piecewise linear and may be dominated by different coefficient matrices alternately. For any PEPA model which has two component types and one synchronisation, the two corresponding coefficient matrices can be proved to have a good property: their eigenvalues are either zeros or have negative real parts. The remaining issue for convergence is to ascertain that these two matrices will not always alternately dominate the system. In fact, we will prove that under some mild conditions, there exists a time after which there is only one coefficient matrix dominating the system. This means the ODEs become linear after that time. Since the coefficient matrix of the linear ODEs satisfies the good eigenvalue property, then by Corollary D.2.3, the bounded solution will converge as time goes to infinity.

We first utilise an example in this section to show our approach to dealing with the convergence problem for this class of PEPA models. The proof for a general case in this class is presented in the next section.

6.4.1 A previous model and the derived ODE

Let us look at the following PEPA model, which is Model 1 presented in Chapter 1:

$$User_1 \stackrel{def}{=} (task_1, a).User_2$$

$$User_2 \stackrel{def}{=} (task_2, b).User_1$$

$$Provider_1 \stackrel{def}{=} (task_1, a).Provider_2$$

$$Provider_2 \stackrel{def}{=} (reset, d).Provider_1$$

$$(User_1[M]) \underset{\{task1\}}{\bowtie} (Provider_1[N]).$$

The derived ODEs are as follows:

$$\begin{cases} \dfrac{\mathrm{d}x_1}{\mathrm{d}t} = -a\min\{x_1, y_1\} + bx_2 \\ \dfrac{\mathrm{d}x_2}{\mathrm{d}t} = a\min\{x_1, y_1\} - bx_2 \\ \dfrac{\mathrm{d}y_1}{\mathrm{d}t} = -a\min\{x_1, y_1\} + dy_2 \\ \dfrac{\mathrm{d}y_2}{\mathrm{d}t} = a\min\{x_1, y_1\} - dy_2 \end{cases} \tag{6.20}$$

where x_i and y_i represent the populations of $User_i$ and $Provider_i$ respectively, $i = 1, 2$. Obviously, (6.20) is equivalent to

$$\begin{pmatrix} \frac{\mathrm{d}x_1}{\mathrm{d}t} \\ \frac{\mathrm{d}x_2}{\mathrm{d}t} \\ \frac{\mathrm{d}y_1}{\mathrm{d}t} \\ \frac{\mathrm{d}y_2}{\mathrm{d}t} \end{pmatrix} = I_{\{x_1 \le y_1\}} Q_1 \begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{pmatrix} + I_{\{x_1 > y_1\}} Q_2 \begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{pmatrix}, \tag{6.21}$$

where

$$Q_1 = \left(\begin{array}{cc|cc} -a & b & 0 & 0 \\ a & -b & 0 & 0 \\ \hline -a & 0 & 0 & d \\ a & 0 & 0 & -d \end{array}\right), \quad Q_2 = \left(\begin{array}{cc|cc} 0 & b & -a & 0 \\ 0 & -b & a & 0 \\ \hline 0 & 0 & -a & d \\ 0 & 0 & a & -d \end{array}\right). \tag{6.22}$$

Our interest is to see if the solution of (6.21) will converge as time goes to infinity. As we mentioned, this convergence problem can be divided into two subproblems, i.e. whether the nonlinear equations can finally become linear and whether the eigenvalues of the coefficient matrix are either zeros or have negative real parts. If answers to these two subproblems are both positive, then the convergence will hold.

The second subproblem can be easily dealt with. By calculations, the matrix Q_1 has eigenvalues 0 (two folds), $-d$, and $-(a+b)$. Similarly, Q_2 has eigenvalues 0 (two folds), $-b, -(a+d)$. Therefore, the eigenvalues of Q_1 and Q_2 other than zeros are negative. Moreover, for a general PEPA model which has two component types and one synchronisation, Theorem 6.5.4 in the next section shows that the corresponding coefficient matrices always have this property.

The remaining work to determine the convergence of the ODE solution, is to solve the first subproblem, i.e. to ascertain that after a time it is always the case that $x_1 > y_1$ or $x_1 \le y_1$. In

this model, there is no invariance relating the two different component types, so we cannot rely on invariants to investigate this subproblem. However, we have a new way to cope with this problem.

6.4.2 Outline of proof

In order to illustrate our approach to deal with those nonlinear ODEs, we will first introduce the techniques used to tackle a linear case in the first part of this subsection. This introduction will help the readers to understand the discussions presented in the second part of this subsection, which will require more complex technical skills to cope with the nonlinearity in the proof of the convergence for PEPA.

6.4.2.1 Discussion on linear ODEs: an illustrated example

Now we use an example to show that the asymptotic behaviour of the solutions of linear ODEs strongly relates to the eigenvalues of the coefficients. The techniques which will be introduced can be generalised to deal with the nonlinear ODEs derived from PEPA models.

Consider a matrix

$$A = \begin{pmatrix} -4 & 2 & 4 \\ 2 & -2 & 2 \\ 2 & 0 & -6 \end{pmatrix}.$$

Obviously, the transpose of A, i.e. A^T, is an infinitesimal generator of a CTMC. This CTMC has three states, namely S_1, S_2 and S_3. The transition rates from S_1 to S_2 and S_3 are both 2, from S_3 to S_1 and S_2 are 4 and 2 respectively. The rate of the transition from S_2 to S_1 is 2 while to S_3 is zero. Obviously, this CTMC has a steady-state probability distribution $\pi = (\pi_1, \pi_2, \pi_3)^T = \left(\frac{3}{8}, \frac{4}{8}, \frac{1}{8}\right)^T$, which is obtained by solving $A\pi = 0$ with $\pi_1 + \pi_2 + \pi_3 = 1$.

Given an initial probability distribution $\mathbf{x}(0)$ for this CTMC, then its transient probability distribution $\mathbf{x}(t)$ at time t is determined by the following ODEs:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = A\mathbf{x}. \tag{6.23}$$

According to the theory of linear ODEs, the solution of (6.23) is $\mathbf{x}(t) = e^{At}\mathbf{x}(0)$, hereafter the exponential of a matrix Q is defined by the series $e^Q = \sum_{k=0}^{\infty} \frac{Q^k}{k!}$ (see [Per91], page 12).

The coefficient matrix A has three distinct eigenvalues: $0, -4$, and -8, so we can diagonalise A:

$$A = U \begin{pmatrix} 0 & 0 & 0 \\ 0 & -4 & 0 \\ 0 & 0 & -8 \end{pmatrix} U^{-1},$$

where

$$U = \begin{pmatrix} 3 & 1 & 1 \\ 4 & -2 & 0 \\ 1 & 1 & -1 \end{pmatrix}, \quad U^{-1} = \begin{pmatrix} \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} \\ -\frac{3}{8} & -\frac{1}{8} & -\frac{5}{8} \end{pmatrix}.$$

In the above, the columns of U correspond to A's eigenvalues $0, -4, -8$ respectively. In particular, the first column of U, i.e. $(3, 4, 1)^T$, is the eigenvector corresponds to the eigenvalue 0. Clearly, the steady-state probability distribution associated this infinitesimal generator, i.e. $\pi = \left(\frac{3}{8}, \frac{4}{8}, \frac{1}{8}\right)^T$, is the normalized eigenvector corresponding the zero eigenvalue.

Denote

$$D = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -4 & 0 \\ 0 & 0 & -8 \end{pmatrix},$$

then $A = UDU^{-1}$, and for a scalar t and an integer k,

$$(tA)^k = \underbrace{tA \cdot tA \cdots tA}_{k \text{ times}} = \underbrace{(U(tD)U^{-1}) \cdot (U(tD)U^{-1}) \cdots (U(tD)U^{-1})}_{k \text{ times}} = U(tD)^k U^{-1}.$$

Notice

$$(tD)^k = \begin{pmatrix} 0^k & 0 & 0 \\ 0 & (-4t)^k & 0 \\ 0 & 0 & (-8t)^k \end{pmatrix},$$

then

$$(tA)^k = U \begin{pmatrix} 0^k & 0 & 0 \\ 0 & (-4t)^k & 0 \\ 0 & 0 & (-8t)^k \end{pmatrix} U^{-1}.$$

Noticing $0! = 0^0 = 1$, so

$$\sum_{k=0}^{\infty} \frac{1}{k!} \begin{pmatrix} 0^k & 0 & 0 \\ 0 & (-4t)^k & 0 \\ 0 & 0 & (-8t)^k \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \sum_{k=0}^{\infty} \frac{(-4t)^k}{k!} & 0 \\ 0 & 0 & \sum_{k=0}^{\infty} \frac{(-8t)^k}{k!} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{-4t} & 0 \\ 0 & 0 & e^{-8t} \end{pmatrix}$$

where the last equality holds is due to the Taylor expansion $e^y = \sum_{k=0}^{\infty} \frac{y^k}{k!}$ for any y. Therefore, we have

$$\begin{aligned} e^{tA} &= \sum_{k=0}^{\infty} \frac{(tA)^k}{k!} \\ &= \sum_{k=0}^{\infty} \frac{1}{k!} U \begin{pmatrix} 0^k & 0 & 0 \\ 0 & (-4t)^k & 0 \\ 0 & 0 & (-8t)^k \end{pmatrix} U^{-1} \\ &= U \sum_{k=0}^{\infty} \frac{1}{k!} \begin{pmatrix} 0^k & 0 & 0 \\ 0 & (-4t)^k & 0 \\ 0 & 0 & (-8t)^k \end{pmatrix} U^{-1} \\ &= U \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{-4t} & 0 \\ 0 & 0 & e^{-8t} \end{pmatrix} U^{-1}. \end{aligned}$$

So, given an initial value $\mathbf{x}(0)$, the solution of (6.23) is

$$\mathbf{x}(t) = e^{At}\mathbf{x}(0) = U \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{-4t} & 0 \\ 0 & 0 & e^{-8t} \end{pmatrix} U^{-1}\mathbf{x}(0). \tag{6.24}$$

As time goes to infinity, both e^{-4t} and e^{-8t} converge to zeros, thus $\mathbf{x}(t)$ converges to a constant $\mathbf{x}^*$, where

$$\mathbf{x}^* = U \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} U^{-1}\mathbf{x}(0). \tag{6.25}$$

With some calculation,

$$\begin{aligned}
\mathbf{x}^* &= U \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} U^{-1}\mathbf{x}(0) \\
&= U \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} U^{-1}\mathbf{x}(0) \\
&= \begin{pmatrix} 3 \\ 4 \\ 1 \end{pmatrix} \begin{pmatrix} \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \end{pmatrix} \begin{pmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \end{pmatrix} \\
&= \begin{pmatrix} \frac{3(x_1(0)+x_2(0)+x_3(0))}{8} \\ \frac{4(x_1(0)+x_2(0)+x_3(0))}{8} \\ \frac{x_1(0)+x_2(0)+x_3(0)}{8} \end{pmatrix}.
\end{aligned} \tag{6.26}$$

Since $\mathbf{x}(0) = (x_1(0), x_2(0), x_3(0))^T$ is an initial probability distribution, so $\sum_{i=1}^{3} x_i(0) = 1$. Thus, $\mathbf{x}^* = \left(\frac{3}{8}, \frac{4}{8}, \frac{1}{8}\right)$, which coincides with the steady-state distribution π. From (6.26), we can see that $\mathbf{x}^*$ is determined by the first column of U and the first row of U^{-1}. Since the first row of U^{-1} only takes the role of normalisation, so the the steady-state distribution is ultimately determined by the first column of U, i.e. the eigenvector corresponds to the eigenvalue zero. In fact, from the equation $A\pi = 0 = 0\pi$, we also know that π is in fact an eigenvector corresponding to the eigenvalue 0.

Moreover, the speed of the convergence to the steady-state distribution depends on the nonzero eigenvalues of the generator. For example, it is easy to see that the convergence speed in our case depends on e^{-4t} and e^{-8t}. Because $e^{-8t} \leq e^{-4t}$, i.e. e^{-8t} converges to zero more quickly than e^{-4t}, so the convergence speed mainly depends on e^{-4t}, or the largest nonzero eigenvalue $\lambda = -4$. The estimation of the largest nonzero eigenvalue (called the *principle eigenvalue* in some literature) is a hot topic in the fields of Markov theory and functional analysis [Che05, Wan05], since it has lots of applications, including determining the speed of converging to equilibria. In our case, a mathematical statement for the dependence of the convergence speed on the principle eigenvalue $\lambda = -4$ is

$$\|\mathbf{x}(t) - \mathbf{x}^*\| \leq Ce^{-4t}. \tag{6.27}$$

where C is a constant.

Not all coefficient matrices can be diagonalised and not all their eigenvalues are real numbers. However, as long as the eigenvalues are either zeros or have negative real parts then the convergence for bounded solutions can be guaranteed by Corollary D.2.3 in Appendix D.2. Furthermore, the largest real part of the eigenvalues determines the convergence speed. That is, if

$$\Lambda = \inf\{-\Re(\lambda) \mid \lambda \text{ is a nonzero eigenvalue of } A\} \tag{6.28}$$

whereafter $\Re(\cdot)$ represents the real part of a complex number, then by Theorem D.2.2 in Appendix D.2, the converge speed depends on Λ, that is,

$$\|\mathbf{x}(t) - \mathbf{x}^*\| \leq C(t)e^{-\Lambda t}. \tag{6.29}$$

Here $C(t)$ is a polynomial function. We should point out that as time goes to infinity, the asymptotic behaviour of the product of a polynomial and an exponential function can mainly be determined by the exponential function. In particular, as in our case, if $C(t) = C$ is a constant then (6.29) becomes

$$\|\mathbf{x}(t) - \mathbf{x}^*\| \leq Ce^{-\Lambda t}. \tag{6.30}$$

Of course, both (6.29) and (6.30) can imply

$$\lim_{t\to\infty} \|\mathbf{x}(t) - \mathbf{x}^*\| = 0. \tag{6.31}$$

Moreover, in the situation that the eigenvalues are unknown, but as long as the corresponding Λ is proven to be positive, then we can claim that $\mathbf{x}$ converges according to (6.29) and (6.30).

6.4.2.2 Dealing with nonlinearity in PEPA

As we have mentioned in the previous subsection, all remaining work to determine the convergence for our ODEs (6.20) or (6.21) is to ensure that the ODEs will finally become linear. For

convenience, the ODEs (6.20) are presented again, see the following

$$\begin{cases} \dfrac{\mathrm{d}x_1}{\mathrm{d}t} = -a\min\{x_1, y_1\} + bx_2 \\ \dfrac{\mathrm{d}x_2}{\mathrm{d}t} = a\min\{x_1, y_1\} - bx_2 \\ \dfrac{\mathrm{d}y_1}{\mathrm{d}t} = -a\min\{x_1, y_1\} + dy_2 \\ \dfrac{\mathrm{d}y_2}{\mathrm{d}t} = a\min\{x_1, y_1\} - dy_2 \end{cases} \tag{6.32}$$

Notice $y_1(t) \leq N$ by the boundedness of solutions. If we can prove that after time T, $x_1(t) \geq cM$, where $c > 0$ is independent of M, we will get, provided $cM > N$,

$$x_1(t) \geq cM > N \geq y_1(t),\ t \geq T.$$

Therefore, the ODEs (6.32) will become linear after time T. We hope to use the techniques discussed previously to prove this conclusion. Because the ODEs considered previously are linear, the previous techniques have to be improved to deal with the nonlinearity in our case.

Let

$$\alpha(t) = \begin{cases} \frac{\min\{x_1(t), y_1(t)\}}{x_1(t)}, & x_1(t) \neq 0, \\ 1, & x_1(t) = 0, \end{cases}$$

then $0 \leq \alpha(t) \leq 1$. The ODEs associated with component type X can be rewritten as

$$\begin{cases} \dfrac{\mathrm{d}x_1}{\mathrm{d}t} = -a\alpha(t)x_1 + bx_2, \\ \dfrac{\mathrm{d}x_2}{\mathrm{d}t} = a\alpha(t)x_1 - bx_2. \end{cases} \tag{6.33}$$

Let

$$A(t) = \begin{pmatrix} -a\alpha(t) & b \\ a\alpha(t) & -b \end{pmatrix}, \quad X(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}.$$

Then (6.33) can be written as

$$\frac{dX(t)}{dt} = A(t)X(t). \tag{6.34}$$

The solution of (6.34) is

$$X(t) = e^{\int_0^t A(s)ds}X(0). \tag{6.35}$$

Notice, if $A(t)$ is a constant matrix A, then the solution $e^{\int_0^t A(s)ds}X(0) = e^{At}X(0)$ coincides

with (6.24), which is in the linear situation.

Let $B(t) = \frac{1}{t}\int_0^t A(s)ds$, then

$$X(t) = e^{\int_0^t A(s)ds}X(0) = e^{tB(t)}X(0), \tag{6.36}$$

and

$$B(t) = \frac{1}{t}\begin{pmatrix} -a\int_0^t \alpha(s)ds & bt \\ a\int_0^t \alpha(s)ds & -bt \end{pmatrix} = \begin{pmatrix} -a\beta(t) & b \\ a\beta(t) & -b \end{pmatrix}, \tag{6.37}$$

where $\beta(t) = \dfrac{\int_0^t \alpha(s)ds}{t}$. Obviously, $0 \leq \beta(t) \leq 1$ because $0 \leq \alpha(s) \leq 1$.

Similar to the diagonalisation of A in Section 6.4.2.1, the matrix $B(t)$ can be diagonalised as

$$\begin{aligned} B(t) &= \begin{pmatrix} -a\beta(t) & b \\ a\beta(t) & -b \end{pmatrix} \\ &= \begin{pmatrix} \frac{b}{a\beta(t)+b} & 1 \\ \frac{a\beta(t)}{a\beta(t)+b} & -1 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & -(a\beta(t)+b) \end{pmatrix} \begin{pmatrix} 1 & 1 \\ \frac{a\beta(t)}{a\beta(t)+b} & -\frac{b}{a\beta(t)+b} \end{pmatrix} \\ &= U(t)\begin{pmatrix} 0 & 0 \\ 0 & -(a\beta(t)+b) \end{pmatrix} U^{-1}(t), \end{aligned} \tag{6.38}$$

where

$$U(t) = \begin{pmatrix} \frac{b}{a\beta(t)+b} & 1 \\ \frac{a\beta(t)}{a\beta(t)+b} & -1 \end{pmatrix}, \quad U^{-1}(t) = \begin{pmatrix} 1 & 1 \\ \frac{a\beta(t)}{a\beta(t)+b} & -\frac{b}{a\beta(t)+b} \end{pmatrix}. \tag{6.39}$$

In the formula (6.38), both 0 and $-(a\beta(t)+b)$ are $B(t)$'s eigenvalues, while the columns of $U(t)$ are the corresponding eigenvectors. In particular, $\left(\dfrac{b}{a\beta(t)+b}, \dfrac{a\beta(t)}{a\beta(t)+b}\right)^T$, i.e. the first column of $U(t)$, is the eigenvector corresponding to the eigenvalue zero.

Analogously to (6.25), we define a function $\hat{X}(t)$ by

$$\hat{X}(t) = U(t)\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} U^{-1}(t)X(0). \tag{6.40}$$

By simple calculation,

$$\hat{X}(t) = (\hat{x}_1(t), \hat{x}_2(t))^T = \left(\frac{bM}{a\beta(t) + b}, \frac{a\beta(t)M}{a\beta(t) + b} \right)^T. \tag{6.41}$$

Obviously, $\hat{X}(t)$ is the normalised eigenvector corresponding the zero eigenvalue (at the time t), analogous to the equilibrium $\mathbf{x}^*$ in Section 6.4.2.1. Here the normalisaton is in terms of the total population M of the component type X. We hope to see that $\hat{X}(t)$ embodies some equilibrium meaning. In fact, we can analogously prove a conclusion: for $X(t)$ and $\hat{X}(t)$ defined as above, we have

$$\lim_{t \to \infty} \|X(t) - \hat{X}(t)\| = 0. \tag{6.42}$$

A proof of (6.42), which relies on the explicit expression of $U(t)$, is given in Appendix D.5. This conclusion is also included in Proposition 6.4.2, which will be introduced later.

Now we discuss the benefit brought by this formula. In fact, by (6.42) the first entry of $X(t)$ approximates the first entry of $\hat{X}(t)$, i.e. $x_1(t)$ approximates $\hat{x}_1(t) = \dfrac{bM}{a\beta(t) + b}$. Thus, for any $\epsilon > 0$, there exists $T > 0$ such that for any $t \geq T$,

$$x_1(t) > \hat{x}_1(t) - \epsilon = \frac{bM}{a\beta(t) + b} - \epsilon.$$

Since $\dfrac{bM}{a+b} \leq \dfrac{bM}{a\beta(t) + b} \leq M$ because $0 \leq \beta(t) \leq 1$, so $x_1(t) > \dfrac{bM}{a+b} - \epsilon$. Therefore, if $\dfrac{bM}{a+b} > N$, then by the boundedness of $y_1(t)$, i.e. $y_1(t) \leq N$, we have

$$x_1(t) > \frac{bM}{a+b} - \epsilon > N \geq y_1(t),$$

as long as ϵ is small enough. This means that Q_2 will dominate the system after time T. So we have

Proposition 6.4.1. *If $\dfrac{bM}{a+b} > N$, then the solution of the ODEs (6.21) converges as time tends to infinity.*

Since the model is symmetric, this proposition has a symmetric version: if $\dfrac{dN}{a+d} > M$, then the solution of the ODEs also converges.

As we discussed, there are two key steps in the proof of Proposition 6.4.1. The first step is to establish the approximation of $x_1(t)$ to $\hat{x}_1(t)$, i.e. $x_1(t) \approx \hat{x}_1(t)$. The second one is to give

an estimation $\hat{x}_1(t) \geq cM$. According to these two conclusions, we have $x_1(t) \geq c'M$ where $c' < c$, and therefore can conclude that $x_1(t) \geq c'M > N > y_1(t)$ provided the condition $c'M > N$. This is the main philosophy behind our proof for $x_1(t) > y_1(t)$.

For the sake of generality, the proofs of these two conclusions should not rely on the explicit expressions of the eigenvalues and eigenvectors of $B(t)$. This is because for general PEPA models with two component types and one synchronisation, these explicit expressions are not always available. The following subsection will present our discussions about these steps, and the proofs for the conclusions which do not rely on these explicit expressions.

6.4.3 Proofs not relying on explicit expressions

This subsection will divide into two parts. In the first part, we will give a lower bound for the eigenvalues of the coefficient matrix $B(t)$, based on which a proof of the approximation of $X(t)$ to $\hat{X}(t)$ is given. The second part will establish the estimation $\hat{x}_1(t) \geq cM$. All proofs in this subsection do not require knowledge of the explicit expressions of the eigenvalues and eigenvectors of $B(t)$.

For convenience, in this subsection we define

$$f(\beta) = \begin{pmatrix} -a\beta & b \\ a\beta & -b \end{pmatrix}, \tag{6.43}$$

where f is a matrix-valued function defined on $\mathbb{R}$. Then the matrix

$$B(t) = \begin{pmatrix} -a\beta(t) & b \\ a\beta(t) & -b \end{pmatrix}$$

is in fact a composition of the functions of $f(\beta)$ and $\beta = \beta(t)$. That is, $B(t) = f(\beta(t))$, where $t \in [0, \infty)$. The diagonalisation of $f(\beta)$ can be written as

$$f(\beta) = g(\beta) \begin{pmatrix} 0 & 0 \\ 0 & \lambda(\beta) \end{pmatrix} g^{-1}(\beta),$$

where $\lambda(\beta)$ is $f(\beta)$'s nonzero eigenvalue, and $g(\beta)$ is a matrix whose columns are the eigenvectors of $f(\beta)$. Here $g^{-1}(\beta)$ is the inverse of the matrix $g(\beta)$. Notice that $\lambda(\beta)$ is real, because if $\lambda(\beta)$ is complex then its conjugation must be an eigenvalue, which is contradicted by the fact

that $f(\beta)$ only has two eigenvalues, 0 and $\lambda(\beta)$. The following discussions in this subsection do not rely on the explicit expressions of $\lambda(\beta)$, $g(\beta)$ and $g^{-1}(\beta)$, although it is easy to see that $\lambda(\beta) = -a\beta + b$ and

$$g(\beta) = \begin{pmatrix} \frac{b}{a\beta+b} & 1 \\ \frac{a\beta}{a\beta+b} & -1 \end{pmatrix}, \quad g^{-1}(\beta) = \begin{pmatrix} 1 & 1 \\ \frac{a\beta}{a\beta+b} & -\frac{b}{a\beta+b} \end{pmatrix}.$$

6.4.3.1 $X(t)$ approximates $\hat{X}(t)$

In the following, we will give a lower bound for the nonzero eigenvalue of $f(\beta)$, i.e. $\lambda(\beta)$, and based on this prove the approximation of $A(t)$ to $\hat{X}(t)$ as time tends to infinity.

If $\beta > 0$, then the transpose of $f(\beta)$, i.e. $f(\beta)^T$, is an infinitesimal generator, and thus the nonzero eigenvalue $\lambda(\beta)$ has negative real part, i.e. $\Re(\lambda(\beta)) < 0$. If $\beta = 0$, then $f(\beta)$ is independent of β and becomes a nonnegative matrix, i.e. each entry of it is nonnegative. Based on the Perron-Frobenious theorem which is presented in the next section, we can still have $\Re(\lambda(0)) < 0$. Therefore, for any β, $f(\beta)$'s eigenvalue other than zero has negative real part. This conclusion is stated in the following lemma.

Lemma 6.4.1. *For any $\beta \in [0,1]$, $\Re(\lambda(\beta)) < 0$, where $\lambda(\beta)$ is a nonzero eigenvalue of $f(\beta)$.*

The proof of Lemma 6.4.1 is presented in Appendix D.6. Lemma 6.4.1 can further lead to the following

Lemma 6.4.2. *Let*

$$\Lambda_1 = \inf_{\beta\in[0,1]} \{-\Re(\lambda(\beta)) \mid \lambda(\beta) \textit{ is } f(\beta)\textit{'s non-zero eigenvalue}\}, \tag{6.44}$$

then $\Lambda_1 > 0$.

Proof. By Lemma 6.4.1, $-\Re(\lambda(\beta)) > 0$ for any $\beta \in [0,1]$, so $\Lambda_1 \geq 0$. Suppose $\Lambda_1 = 0$. Because the eigenvalue $\lambda(\beta)$ is a continuous function of the matrix $f(\beta)$, where $f(\beta)$ is also continuous on $[0,1]$ with respect to β, so $\lambda(\beta)$ is a continuous function of β on $[0,1]$. This is due to the fact that a composition of continuous functions is still continuous. Noticing $\Re(\cdot)$ is also a continuous function, so $-\Re(\lambda(\beta))$ is continuous with respect to $\lambda(\beta)$, and thus with respect to β on $[0,1]$. Since a continuous function on a closed interval can achieve its minimum (see Theorem D.1.3 in Appendix D.1), there exists $\beta_0 \in [0,1]$ such that $-\Re(\lambda(\beta_0))$ achieves

the minimum Λ_1, i.e. $-\Re(\lambda(\beta_0)) = \Lambda_1 = 0$. This is contradicted to Lemma 6.4.1. Therefore, $\Lambda_1 > 0$. □

For any $t \in [0, \infty)$, $\beta(t) \in [0, 1]$. Thus $\{\beta(t) \mid t \in [0, \infty)\} \subseteq [0, 1]$. Noticing $B(t) = f(\beta(t))$, therefore

$$\begin{aligned} &\{\lambda \mid \lambda \text{ is } B(t)\text{'s non-zero eigenvalue}, t > 0\} \\ =&\{\lambda \mid \lambda \text{ is } f(\beta(t))\text{'s non-zero eigenvalue}, t > 0\} \\ \subseteq&\{\lambda \mid \lambda \text{ is } f(\beta)\text{'s non-zero eigenvalue}; \beta \in [0, 1]\}. \end{aligned} \tag{6.45}$$

Thus,

$$\begin{aligned} \Lambda &\triangleq \inf\{-\Re(\lambda) \mid \lambda \text{ is } B(t)\text{'s nonzero eigenvalue}, t > 0\} \\ &\geq \inf\{-\Re(\lambda) \mid \lambda \text{ is } f(\beta)\text{'s non-zero eigenvalue}; \beta \in [0, 1]\} = \Lambda_1. \end{aligned} \tag{6.46}$$

Because $\Lambda_1 > 0$ by Lemma 6.4.2, so $\Lambda > 0$. That is,

Corollary 6.4.3. *Let*

$$\Lambda = \inf_{t \geq 0}\{-\Re(\lambda(t)) \mid \lambda(t) \textit{ is } B(t)\textit{'s non-zero eigenvalue}\},$$

then $\Lambda > 0$.

Based on this corollary, we can prove the approximation of $X(t)$ to $\hat{X}(t)$.

Proposition 6.4.2. *Let* $X(t) = (x_1(t), x_2(t))^T = e^{tB(t)}X(0)$*, i.e. the solution of (6.33). Let* $\hat{X}(t)$ *be defined by (6.40), i.e.*

$$\hat{X}(t) = \begin{pmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \end{pmatrix} = U(t) \begin{pmatrix} 1 & \\ & 0 \end{pmatrix} U(t)^{-1} \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix}. \tag{6.47}$$

Then $\lim_{t \to \infty} \|X(t) - \hat{X}(t)\| = 0$.

Proof. Notice that eigenvectors of a matrix are continuous functions of the matrix. Since $g(\beta)$ is composed of the eigenvectors of the matrix $f(\beta)$ and $f(\beta)$ is continuous on $[0, 1]$ with respect to β, therefore $g(\beta)$ is continuous on $[0, 1]$ with respect to β. Because the inverse of a matrix is a continuous mapping, so $g^{-1}(\beta)$, i.e. the inverse of $g(\beta)$, is continuous with respect to $g(\beta)$, and

therefore is continuous on $[0,1]$ with respect to β since $g(\beta)$ is continuous on $[0,1]$. Since any continuous function is bounded on a compact set $[0,1]$ (see Theorem D.1.3 in Appendix D.1), both $g(\beta)$ and $g^{-1}(\beta)$ are bounded on $[0,1]$. That is, there exists $K > 0$ such that $\|g(\beta)\| \leq K$ and $\|g^{-1}(\beta)\| \leq K$ for all $\beta \in [0,1]$. Because

$$\{U(t) \mid t \in [0,\infty)\} = \{g(\beta(t)) \mid t \in [0,\infty)\} \subseteq \{g(\beta) \mid \beta \in [0,1]\},$$

we have

$$\sup_{t\geq 0} \|U(t)\| \leq \sup_{\beta\in[0,1]} \|g(\beta)\| \leq K.$$

Similarly, $\sup_{t\geq 0} \left\|U^{-1}(t)\right\| \leq K$. Notice

$$\begin{aligned}
X(t) - \hat{X}(t) =& e^{tB(t)}X(0) - U(t)\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} U(t)^{-1}X(0) \\
=& \left[U(t)\begin{pmatrix} 1 & 0 \\ 0 & e^{t\lambda(t)} \end{pmatrix} U(t)^{-1} - U(t)\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} U(t)^{-1}\right] X(0) \\
=& U(t)\begin{pmatrix} 0 & 0 \\ 0 & e^{t\lambda(t)} \end{pmatrix} U(t)^{-1}X(0),
\end{aligned}$$

where $\lambda(t)$ is $B(t)$'s nonzero eigenvalue. By a similar argument to $\lambda(\beta)$, $\lambda(t)$ is also real. Therefore,

$$-\lambda(t) = \Re(-\lambda(t)) = -\Re(\lambda(t)) \geq \Lambda > 0$$

or $\lambda(t) \leq -\Lambda < 0$, where Λ is defined in Corollary 6.4.3. Then

$$\begin{aligned}
\|X(t) - \hat{X}(t)\| =& \left\|U(t)\begin{pmatrix} 0 & 0 \\ 0 & e^{t\lambda(t)} \end{pmatrix} U(t)^{-1}X(0)\right\| \\
\leq& \|U(t)\| \left\|\begin{pmatrix} 0 & 0 \\ 0 & e^{t\lambda(t)} \end{pmatrix}\right\| \|U(t)^{-1}\|\|X(0)\| \\
\leq& K^2\|X(0)\|e^{t\lambda(t)} \\
\leq& K^2\|X(0)\|e^{-t\Lambda}.
\end{aligned}$$

Here we have used the norm property: $\|AB\| \leq \|A\|\|B\|$. Since $\Lambda > 0$, we have $\lim_{t\to\infty} \|X(t) - \hat{X}(t)\| = 0$. □

6.4.3.2 An lower-bound estimation on population in local derivatives

In the following, we will prove that there exists T, such that $\hat{x}_1(t) \geq cM$ for any $t > T$.

We first define a function

$$h(\beta) = (h_1(\beta), h_2(\beta))^T = g(\beta) \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} g^{-1}(\beta) \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix}.$$

Clearly, we have $\hat{X}(t) = h(\beta(t))$ and $\hat{x}_1(t) = h_1(\beta(t))$. Since $\beta(t) \in [0,1]$ for all t, the following proposition can imply $\hat{x}_1(t) \geq cM$.

Proposition 6.4.3. *There exists $c > 0$ such that*

$$\inf_{\beta \in [0,1]} h_1(\beta) \geq cM.$$

where $M = x_1(0) + x_2(0)$, c is independent of M.

Proof. Without loss of generality, we assume $M = 1$. We will show $\inf_{\beta \in [0,1]} h_1(\beta) = c > 0$. Since $h_1(\beta)$ is a continuous function of β which is due to the continuity of $g(\beta)$ and $g^{-1}(\beta)$, by Theorem D.1.3 in Appendix D.1, $h_1(\beta)$ can achieve its minimum on $[0,1]$. That is, there exists $\beta_0 \in [0,1]$, such that

$$h_1(\beta_0) = \inf_{\beta \in [0,1]} h_1(\beta) = c.$$

Consider the matrix

$$f(\beta_0) = \begin{pmatrix} -a\beta_0 & b \\ a\beta_0 & -b \end{pmatrix}$$

and a set of linear ODEs

$$\begin{pmatrix} \frac{\mathrm{d}z_1}{\mathrm{d}t} \\ \frac{\mathrm{d}z_2}{\mathrm{d}t} \end{pmatrix} = f(\beta_0) \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}. \tag{6.48}$$

The solution of (6.48), given an initial value $Z(0) = X(0) = (x_1(0), x_2(0))^T$, is $Z(t) = e^{tf(\beta_0)} X(0)$.

According to (6.38), $f(\beta_0)$ can be diagonalised as

$$f(\beta_0) = g(\beta_0) \begin{pmatrix} 0 & 0 \\ 0 & \lambda(\beta_0) \end{pmatrix} g^{-1}(\beta_0).$$

where $\lambda(\beta_0)$ is the nonzero and real eigenvalue of $f(\beta_0)$. Similarly to (6.24) and the discussions in Section 6.4.2.1,

$$Z(t) = e^{tf(\beta_0)}X(0) = g(\beta_0)\begin{pmatrix} 1 & 0 \\ 0 & e^{t\lambda(\beta_0)} \end{pmatrix} g(\beta_0)^{-1}\begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix}. \tag{6.49}$$

Because $\lambda(\beta_0) < 0$ by Lemma 6.4.1, so as time goes to infinity,

$$\begin{aligned} Z(t) &= g(\beta_0)\begin{pmatrix} 1 & 0 \\ 0 & e^{t\lambda(\beta_0)} \end{pmatrix} g(\beta_0)^{-1}\begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} \\ &\longrightarrow g(\beta_0)\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} g(\beta_0)^{-1}\begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = h(\beta_0). \end{aligned} \tag{6.50}$$

That is, $\lim_{t\to\infty} Z(t) = h(\beta_0)$. In the following, we discuss two possible cases: $\beta_0 > 0$ and $\beta_0 = 0$.

If $\beta_0 > 0$, then the transpose of the matrix $f(\beta_0)$, i.e. $f(\beta_0)^T$, is an infinitesimal generator of an irreducible CTMC, which has two states and the transition rates between these two states are $a\beta_0$ and b respectively. Moreover, the transient distribution of this CTMC, denoted by $Z(t) = (z_1(t), z_2(t))^T$, satisfies the ODEs (6.48). As time goes to infinity, the transient distribution $Z(t)$ converges to the unique steady-state probability distribution. Since $\lim_{t\to\infty} Z(t) = h(\beta_0)$, therefore $h(\beta_0) = (h_1(\beta_0), h_2(\beta_0))^T$ is the steady-state probability distribution and thus $h_1(\beta_0) > 0$. So $\inf_{\beta\in[0,1]} h_1(\beta) = h_1(\beta_0) > 0$.

If $\beta_0 = 0$, then

$$f(\beta_0) = \begin{pmatrix} 0 & b \\ 0 & -b \end{pmatrix}.$$

Since $\lim_{t\to\infty} Z(t) = h(\beta_0)$, therefore $\left(\frac{\mathrm{d}z_1}{\mathrm{d}t}, \frac{\mathrm{d}z_2}{\mathrm{d}t}\right)^T$ converges to zero. Letting time go to infinity on the both sides of (6.48), we obtain the following equilibrium equations,

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} = f(\beta_0)\begin{pmatrix} h_1(\beta_0) \\ h_2(\beta_0) \end{pmatrix}. \tag{6.51}$$

By the conservation law, $(h_1(\beta_0)+h_2(\beta_0))^T = M = 1$. Therefore, $(h_1(\beta_0), h_2(\beta_0))^T$ satisfies

$$\begin{cases} f(\beta_0)(h_1(\beta_0), h_2(\beta_0))^T = 0, \\ h_1(\beta_0) + h_2(\beta_0) = M = 1. \end{cases} \tag{6.52}$$

Solving (6.52), we obtain the unique solution $(h_1(\beta_0), h_2(\beta_0))^T = (1, 0)^T$. Therefore, $h_1(\beta_0)$ is one, and thus $\inf_{\beta\in[0,1]} h_1(\beta) = h_1(\beta_0) > 0$. □

Remark 6.4.1. *As β tends to 0,*

$$f(\beta) = \begin{pmatrix} -a\beta & b \\ a\beta & -b \end{pmatrix} \longrightarrow f(0) = \begin{pmatrix} 0 & b \\ 0 & -b \end{pmatrix}.$$

Correspondingly, for the equilibrium $h(\beta) = (h_1(\beta), h_2(\beta))^T = \left(\frac{bM}{a\beta+b}, \frac{a\beta M}{a\beta+b}\right)^T$ satisfying $f(\beta)h(\beta) = 0$ and $h_1(\beta) + h_2(\beta) = M$, we have $\left(\frac{bM}{a\beta+b}, \frac{a\beta M}{a\beta+b}\right)^T \to (M, 0)^T$ as β tends to zero. From the explicit expression, i.e. $h_1(\beta) = \frac{bM}{a\beta+b}$, the minimum and maximum of $h_1(\beta)$ are $\frac{bM}{a+b}$ and M respectively, which correspond to the matrices $f(1)$ and $f(0)$ respectively. In the context of the PEPA model, $f(1)$ corresponds to a free subsystem and there is no synchronisation effect on it, i.e. the subsystem of component type X is independent of Y. The matrix $f(0)$ reflects that the subsystem of X has been influenced by the subsystem of Y, i.e. the rates of shared activities are determined by Y, that is, the term $a\min\{x_1, y_1\}$ has been replaced by ay_1. Therefore the exit rates from the local derivative X_1 correspondingly become smaller since now $ay_1 < ax_1$. In order to balance the flux, which is described by the equilibrium equation, the population of X_1 must increase. That is why the equilibrium $h_1(\beta)$ increases as β decreases. In short, synchronisations can increase the populations in syncrhonised local derivatives in the steady state.

As an application of the above facts, if $h_1(\beta_0) > 0$ for some $\beta_0 > 0$, then we can claim that $h_1(0) > 0$ because $h_1(0) \geq h_1(\beta_0) > 0$.

Obviously, Proposition 6.4.3 has a corollary:

Corollary 6.4.4. *There exists $c > 0$ such that for any $t \in [0, \infty)$, $\hat{x}_1(t) \geq cM$.*

Proposition 6.4.2 and Proposition 6.4.3 can lead to the following lemma.

Lemma 6.4.5. *There exists $c > 0, T > 0$, such that $x_1(t) \geq cM$ for all $t \geq T$.*

Proof. By Proposition 6.4.3 or Corollary 6.4.4, there exists $c_1, T_1 > 0$ such that $\hat{x}_1(t) \geq c_1 M$ for any $t > T_1$. By Proposition 6.4.2, $\lim_{t\to\infty} |x_1(t) - \hat{x}_1(t)| = 0$, which implies that for any ϵ, there exists $T_2 > 0$ such that for any $t > T_2$, $x_1(t) > \hat{x}_1(t) - \epsilon$. Choose $T_2 > T_1$, then we have

$$x_1(t) > \hat{x}_1(t) - \epsilon \geq c_1 M - \epsilon.$$

Therefore, there exist $c, T > 0$ such that such that $x_1(t) \geq cM$ for all $t > T$. □

Because $x_1(t) \geq cM$, provided $cM > N$ we have $x_1(t) \geq cM > N \geq y_1(t)$, i.e., the system will finally become linear. In the following we will show how to apply our method to more general cases.

6.5 Convergence For Two Component Types and One Synchronisation (II): General Case

This section deals with such an arbitrary PEPA model which has two component types and one synchronisation. The local action rates of the shared activity are not assumed to be the same. The main result of this section is a convergence theorem: as long as the population of one component type is sufficiently larger than the population of the other, then the solution of the derived ODEs converges as time tends to infinity.

6.5.1 Features of coefficient matrix

We assume the component types to be X and Y. The component type X is assumed to have local derivatives $X_1, X_2, \cdots, X_m$, while Y has local derivatives $Y_1, Y_2, \cdots, Y_n$. We use $x_i(t)$ to denote the population of X in X_i $(i = 1, \cdots, m)$ at time t. Similarly, $y_j(t)$ denotes the population of Y in Y_j $(j = 1, \cdots, n)$ at time t. Without loss of generality, we assume the synchronisation is associated with the local derivatives X_1 and Y_1, i.e. the nonlinear term in the derived ODEs is $\min\{rx_1(t), sy_1(t)\}$ where r and s are some constants. In fact, if the synchronisation is associated with X_i and Y_j, by appropriately permuting their suffixes, i.e. $i \to 1,\ i+1 \to 2,\ \cdots, i-1 \to m, j \to 1,\ j+1 \to 2,\ \cdots, j-1 \to n$, the synchronisation will be associated with X_1 and Y_1. According to the mapping semantics presented in the

previous chapter, the derived ODEs from this class of PEPA model are

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \sum_{l} l f(\mathbf{x}, l) \tag{6.53}$$

where $\mathbf{x} = (x_1(t), \cdots, x_m(t), y_1(t), \cdots, y_n(t))^T$. For convenience, we denote

$$X(t) = (x_1(t), x_2(t), \cdots, x_m(t))^T,$$

$$Y(t) = (y_1(t), y_2(t), \cdots, y_n(t))^T.$$

In (6.53) all terms are linear except for those containing "$\min\{rx_1(t), sy_1(t)\}$". Notice

$$\min\{rx_1(t), sy_1(t)\} = I_{\{rx_1(t) \leq sy_1(t)\}} rx_1(t) + I_{\{rx_1(t) > sy_1(t)\}} sy_1(t).$$

When $rx_1(t) \leq sy_1(t)$, which is indicated by $I_{\{rx_1(t) \leq sy_1(t)\}} = 1$ and $I_{\{rx_1(t) > sy_1(t)\}} = 0$, we can replace $\min\{rx_1(t), sy_1(t)\}$ by $rx_1(t)$ in (6.53). Then (6.53) becomes linear since all nonlinear terms are replaced by linear terms $rx_1(t)$, so the ODEs have the following form,

$$\begin{pmatrix} \frac{\mathrm{d}X}{\mathrm{d}t} \\ \frac{\mathrm{d}Y}{\mathrm{d}t} \end{pmatrix} = Q_1 \begin{pmatrix} X \\ Y \end{pmatrix}, \tag{6.54}$$

where Q_1 is a coefficient matrix. Similarly, if $rx_1(t) > sy_1(t)$, $\min\{rx_1(t), sy_1(t)\}$ can be replaced by $sy_1(t)$ in (6.53). Then (6.53) can become

$$\begin{pmatrix} \frac{\mathrm{d}X}{\mathrm{d}t} \\ \frac{\mathrm{d}Y}{\mathrm{d}t} \end{pmatrix} = Q_2 \begin{pmatrix} X \\ Y \end{pmatrix}, \tag{6.55}$$

where Q_2 is another coefficient matrix corresponding to the case of $rx_1(t) > sy_1(t)$.

In short, the derived ODEs (6.53) are just the following

$$\begin{pmatrix} \frac{\mathrm{d}X}{\mathrm{d}t} \\ \frac{\mathrm{d}Y}{\mathrm{d}t} \end{pmatrix} = I_{\{rx_1 \leq sy_1\}} Q_1 \begin{pmatrix} X \\ Y \end{pmatrix} + I_{\{rx_1 > sy_1\}} Q_2 \begin{pmatrix} X \\ Y \end{pmatrix}. \tag{6.56}$$

The case discussed in the previous section is a special case of this kind of form. If the conditions $rx_1(t) \leq sy_1(t)$ and $rx_1(t) > sy_1(t)$ occur alternately, then the matrices Q_1 and Q_2 will correspondingly alternately dominate the system, as Figure 6.6 illustrates.

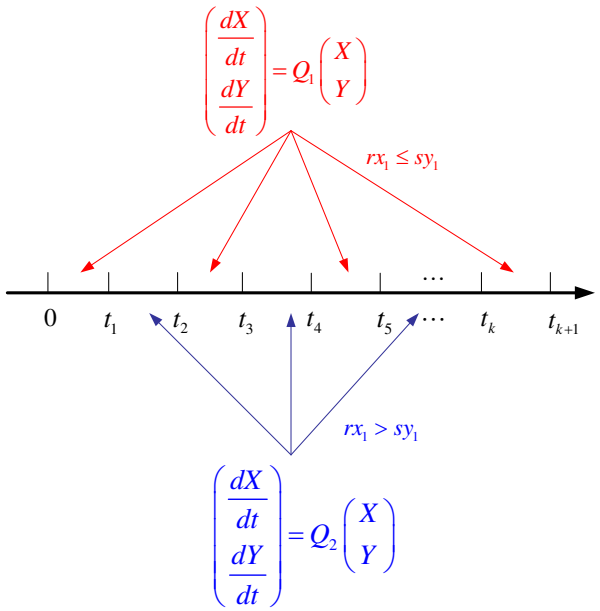


Figure 6.6: *Illustration of derived ODEs with component types and one synchronisation*

Similar to the cases discussed in the previous two sections, the convergence problem of (6.56) can be divided into two subproblems, i.e. to examine whether the following two properties hold:

1. There exists a time T, such that either $x_1 \le y_1,\ \forall t > T$ or $x_1 \le y_1,\ \forall t > T$.

2. The eigenvalues of Q_1 and Q_2 other than zeros have negative real parts.

The first item can guarantee (6.56) to eventually have a constant linear form, while the second item ensures the convergence of the bounded solution of the linear ODEs. If the answers to these two problems are both positive, then the convergence of the solution of (6.56) will hold. The study of these two problems are discussed in the next two subsections. In the remainder of this subsection, we first investigate the structure property of the coefficient matrices Q_1 and Q_2 in (6.56).

The structure of the coefficient matrices Q_1 and Q_2 is determined by the following two propositions, which indicate that they are either block lower-triangular or block upper-triangular.

Proposition 6.5.1. *Q_1 in (6.56) can be written as*

$$Q_1 = \begin{pmatrix} \hat{Q}_1 & 0 \\ W & V_{n\times n} \end{pmatrix}_{(m+n)\times(m+n)}, \tag{6.57}$$

where $\hat{Q}_1^T$ is an infinitesimal generator matrix with the dimension $m \times m$, and

$$W_{n\times m} = \begin{pmatrix} w_{11} & 0 & \cdots & 0 \\ w_{21} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ w_{n1} & 0 & \cdots & 0 \end{pmatrix}, \tag{6.58}$$

where $w_{11} < 0$, $w_{j1}(j = 2, \cdots, n) \geq 0$ and $\sum_{j=1}^{n} w_{j1} = 0$. Here V and W satisfy that if we let

$$P = (W_1 + V_1, V_2, \cdots, V_n), \tag{6.59}$$

i.e. P's first column is the sum of V's first column and W's first column, while P's other columns are the same to V's other columns, then P^T is also an infinitesimal generator matrix.

Proof. Let

$$Q_1 = \begin{pmatrix} \hat{Q}_1 & U \\ W & V \end{pmatrix}, \tag{6.60}$$

where $\hat{Q}_1$ and V are $m \times m$ and $n \times n$ matrices respectively. Suppose $rx_1(t) \leq sy_1(t)$, then

$$\begin{pmatrix} \frac{\mathrm{d}X}{\mathrm{d}t} \\ \frac{\mathrm{d}Y}{\mathrm{d}t} \end{pmatrix} = Q_1 \begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} \hat{Q}_1 & U \\ W & V \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix}. \tag{6.61}$$

So we have

$$\frac{\mathrm{d}X}{\mathrm{d}t} = \hat{Q}_1 X + UY. \tag{6.62}$$

The condition $rx_1(t) \leq sy_1(t)$ implies that all nonlinear terms $\min\{rx_1(t), sy_1(t)\}$ can be replaced by $x_1(t)$. This means that the behaviour of the component type X in (6.61) and (6.62) is independent of the component type Y. Thus in (6.61) U must be a zero matrix, i.e.

$$Q_1 = \begin{pmatrix} \hat{Q}_1 & 0 \\ W & V \end{pmatrix}.$$

Moreover, (6.62) becomes

$$\frac{\mathrm{d}X}{\mathrm{d}t} = \hat{Q}_1 X, \tag{6.63}$$

that is, there is no synchronisation in the ODEs corresponding to the component type X given $rx_1(t) \leq sy_1(t)$. Then by Proposition 5.3.1 in Chapter 4, $\hat{Q}_1^T$ is an infinitesimal generator.

According to (6.61),

$$\begin{aligned}\frac{\mathrm{d}Y}{\mathrm{d}t} &= WX + VY \\ &= (W_1, W_2, \cdots, W_m)(x_1, x_2, \cdots, x_m)^T + VY \\ &= x_1 W_1 + VY + \sum_{i=2}^{m} x_i W_i,\end{aligned} \tag{6.64}$$

where $W = (W_1, W_2, \cdots, W_m)$. Notice that the component type Y is synchronised with the component type X only through the term $\min\{rx_1(t), sy_1(t)\} = x_1(t)$. In other words, in (6.64) Y is directly dependent on only x_1 other than x_i $(i \geq 2)$. This implies $W_1 \neq 0$ while $W_i = 0$ $(i = 2, 3, \cdots, m)$. Therefore,

$$\begin{aligned}\frac{\mathrm{d}Y}{\mathrm{d}t} &= x_1 W_1 + VY \\ &= x_1 W_1 + \sum_{j=1}^{n} y_j V_j,\end{aligned} \tag{6.65}$$

where V_j $(j = 1, 2, \cdots, n)$ are the columns of V. Denote $W_1 = (w_{11}, w_{21}, \cdots, w_{n1})^T$. Notice that Y_1 is a pre local derivative of the shared activity, and $x_1 w_{11}$ represents the exit rates of the shared activity from Y_1. Therefore, $w_{11} < 0$. Moreover, $x_1 w_{j1}$ $(j = 2, \cdots, n)$ are the synchronised entry rates for the local derivatives Y_j $(j = 2, \cdots, n)$ respectively, so $w_{j1} \geq 0$ $(j = 2, \cdots, n)$. By the conservation law, the total synchronised exit rates are equal to the total synchronised entry rates, i.e. $x_1 \sum_{j=1}^{n} w_{j1} = 0$ or $\sum_{j=1}^{n} w_{j1} = 0$.

We have known that x_1 in (6.65) derives from the synchronised term $\min\{rx_1, sy_1\}$. If the effect of the synchronisation on the behaviour of Y is removed, i.e. recover y_1 by replacing x_1,

then (6.65) will become

$$\begin{aligned}\frac{\mathrm{d}Y}{\mathrm{d}t} &= y_1 W_1 + VY \\ &= y_1 W_1 + \sum_{j=1}^{n} y_j V_j \\ &= PY,\end{aligned} \tag{6.66}$$

where $P = (W_1 + V_1, V_2, \cdots, V_n)$. Since there is no synchronisation contained in the subsystem of the component type Y, according to Proposition 5.3.1 in Chapter 4, P^T is the infinitesimal generator. □

Similarly, we can prove

Proposition 6.5.2. *Q_2 in (6.56) can be written as*

$$Q_2 = \begin{pmatrix} E_{m\times m} & F \\ 0 & \hat{Q}_2 \end{pmatrix}_{(m+n)\times(m+n)}, \tag{6.67}$$

where $\hat{Q}_2^T$ is an infinitesimal generator matrix with the dimension $n \times n$, and

$$F_{m\times n} = \begin{pmatrix} f_{11} & 0 & \cdots & 0 \\ f_{21} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ f_{m1} & 0 & \cdots & 0 \end{pmatrix}, \tag{6.68}$$

where $f_{11} < 0$, $f_{j1}(j = 2, \cdots, m) \geq 0$ and $\sum_{j=1}^{m} f_{j1} = 0$. Here F and E satisfy that if we let

$$R = (F_1 + E_1, E_2, \cdots, E_n), \tag{6.69}$$

then R^T is also an infinitesimal generator matrix.

6.5.2 Eigenvalues of Q_1 and Q_2

In this subsection, we will determine the eigenvalue property of Q_1 and Q_2. First, the Perron-Frobenius theorem gives an estimation of eigenvalues for nonnegative matrices.

Theorem 6.5.1. ***(Perron-Frobenius)**. Let $A = (a_{ij})$ be a real $n \times n$ matrix with nonnegative*

entries $a_{ij} \geq 0$. Then the following statements hold:

1. *There is a real eigenvalue r of A such that any other eigenvalue λ satisfies $|\lambda| \leq r$.*

2. *r satisfies $\min_i \sum_j a_{ij} \leq r \leq \max_i \sum_j a_{ij}$.*

Remark 6.5.1. *We should point out that in the second property, exchanging i and j in a_{ij} in the formula, we still have $\min_i \sum_j a_{ji} \leq r \leq \max_i \sum_j a_{ji}$. In fact, A^T is also a real matrix with non-negative entries. Since A^T and A share the same eigenvalues, so r is one of the eigenvalues of A^T, such that any other eigenvalue λ of A^T satisfies $|\lambda| \leq r$. Notice $(A^T)_{ij} = A_{ji}$, By applying the Perron-Frobenius theorem to A^T, we have*

$$\min_i \sum_j a_{ji} \leq r \leq \max_i \sum_j a_{ji}.$$

We cannot directly apply this theorem to our coefficient matrices Q_1 and Q_2, since both of them have negative elements, not only on the diagonal but also in other entries. However, we use some well-known techniques in linear algebra, i.e. the following Lemma 6.5.2 and 6.5.3 (which can be easily found in linear algebra textbooks), to cope with this problem, and thus derive estimates of their eigenvalues.

Lemma 6.5.2. *If $E_{m\times m}$ and $F_{n\times n}$ have eigenvalues λ_i $(i = 1, 2, \cdots, m)$ and δ_j $(j = 1, 2, \cdots, n)$ respectively, then*

$$H = \begin{pmatrix} E & 0 \\ G & F \end{pmatrix}$$

has eigenvalues λ_i $(i = 1, 2, \cdots, m)$ and δ_j $(j = 1, 2, \cdots, n)$.

Proof. Since E and F are square matrices, then by classical linear algebra theory,

$$\begin{aligned} |\lambda I_{(m+n)\times(m+n)} - H| &= \left| \lambda I_{(m+n)\times(m+n)} - \begin{pmatrix} E & 0 \\ G & F \end{pmatrix} \right| \\ &= \left| \begin{pmatrix} \lambda I_{m\times m} - E & 0 \\ -G & \lambda I_{n\times n} - F \end{pmatrix} \right| \\ &= |\lambda I_{m\times m} - E| \times |\lambda I_{n\times n} - F|, \end{aligned}$$

which implies

$$|\lambda I_{(m+n)\times(m+n)} - H| = 0 \Longleftrightarrow |\lambda I_{m\times m} - E| = 0 \text{ or } |\lambda I_{n\times n} - F| = 0.$$

In other words, E's and F's eigenvalues are H's eigenvalues, each eigenvalue of H is an eigenvalue of either E or F. □

Remark 6.5.2. *This lemma is also valid for a block upper-triangular matrix,*

$$H = \begin{pmatrix} E & G \\ 0 & F \end{pmatrix}.$$

The benefit of the lemma, in terms of eigenvalues, is to avoid caring about G, the lower-left or upper-right block of H.

Lemma 6.5.2 shows that the eigenvalues of a block lower-triangular or block upper-triangular matrix are only determined by the diagonal blocks. For example, the following two matrices

$$\begin{pmatrix} -a & b \\ a & -b \end{pmatrix}, \qquad \begin{pmatrix} 0 & d \\ 0 & -d \end{pmatrix}$$

have eigenvalues $0, -(a+b)$ and $0, -d$ respectively. Notice

$$Q_1 = \left(\begin{array}{cc|cc} -a & b & 0 & 0 \\ a & -b & 0 & 0 \\ \hline -a & 0 & 0 & d \\ a & 0 & 0 & -d \end{array}\right)$$

has eigenvalues 0 (two folds), $-d, -(a+b)$. This is consistent with Lemma 6.5.2.

Lemma 6.5.3. *If λ is an eigenvalue of V, then $\lambda + r$ is an eigenvalue of $V + rI$, where r is a scalar.*

Proof. Let $\mathbf{x}$ be the eigenvector corresponding to λ, then

$$(V + rI)\mathbf{x} = V\mathbf{x} + r\mathbf{x} = \lambda\mathbf{x} + r\mathbf{x} = (\lambda + r)\mathbf{x}.$$

So $\lambda + r$ is an eigenvalue of $V + rI$. □

Remark 6.5.3. *If some diagonal elements of V are negative, by adding such a matrix rI where r is large enough, all the diagonal elements can become positive or nonnegative. Meanwhile, all eigenvalues simply have an r-shift.*

Theorem 6.5.4. *The eigenvalues of both Q_1 and Q_2 are either zeros or have negative real parts.*

Proof. We only give the proof for Q_1's case. By Proposition 6.5.1,

$$Q_1 = \begin{pmatrix} \hat{Q}_1 & 0 \\ W & V_{n\times n} \end{pmatrix}_{(m+n)\times(m+n)}. \tag{6.70}$$

According to Lemma 6.5.2, if all eigenvalues of $\hat{Q}_1$ and V are determined, then the eigenvalues of Q_1 will be determined. Let us consider V first.

Notice that only diagonal elements of V are possibly negative (which can be deduced from Proposition 6.5.1). Let $r = \sup_i |V_{ii}| > 0$, then all the entries of $V + rI$ are nonnegative. Let λ be an arbitrary eigenvalue of V, then by Lemma 6.5.3, $\lambda + r$ is an eigenvalue of $V + rI$.

Notice the sum of the elements of any column of V is zero (because the sum of entry rates equals to the sum of exit rates), so V has a zero eigenvalue with the corresponding eigenvector $\mathbf{1}$, i.e. $V\mathbf{1} = \mathbf{0}$. Thus $r = 0 + r$ is an eigenvalue of $V + rI$. Moreover,

$$\min_i \sum_j (V + rI)_{ji} = r = \max_i \sum_j (V + rI)_{ji}.$$

Applying the Perron-Frobenius theorem (Theorem 6.5.1) and Remark 6.5.1 to $V + rI$, so

$$|\lambda + r| \leq r. \tag{6.71}$$

Let $\lambda = a + bi$, then (6.71) implies that $a \leq 0$, and if $a = 0$ then $b = 0$. In other words, V's eigenvalues are either zeros or have negative real parts.

Similarly, $\hat{Q}_1$'s eigenvalues other than zeros have negative real parts. By Lemma 6.5.2, the eigenvalues of Q_1 other than zeros have negative real parts. The proof is complete. □

6.5.3 Convergence theorem

Now we deal with another subproblem: whether or not after a long time, we always have $rx_1 \geq sy_1$ (or $rx_1 < sy_1$). If the population of X is significantly larger than the population of Y, intuitively, there will finally be a greater number of X in the local derivative X_1, than the number of Y in Y_1. This will lead to $rx_1 > sy_1$.

Lemma 6.5.5. *Under the assumptions in Section 6.4.1, for the following ODEs*

$$\begin{pmatrix} \frac{\mathrm{d}X}{\mathrm{d}t} \\ \frac{\mathrm{d}Y}{\mathrm{d}t} \end{pmatrix} = I_{\{rx_1 \leq sy_1\}} Q_1 \begin{pmatrix} X \\ Y \end{pmatrix} + I_{\{rx_1 > sy_1\}} Q_2 \begin{pmatrix} X \\ Y \end{pmatrix},$$

there exists $c_1 > 0, c_2 > 0, T > 0$, such that $x_1(t) \geq c_1 M$, $y_1(t) \geq c_2 N$ for any $t \geq T$, where c_1 and c_2 are independent of M and N.

Proof. The proof is essentially the same as the proof of Lemma 6.4.5. We only give the sketch of the proof for $x_1(t) \geq c_1 M$. By introducing new two functions $\alpha(t)$ and $\beta(t)$,

$$\alpha(t) = \begin{cases} \frac{\min\{rx_1(t), sy_1(t)\}}{rx_1(t)}, & x_1(t) \neq 0, \\ 0, & x_1(t) = 0. \end{cases},$$

$$\beta(t) = \frac{1}{t} \int_0^t \alpha(s) ds,$$

the nonlinear term $\min\{rx_1(t), sy_1(t)\}$ equals $r\alpha(t)x_1(t)$, and thus the ODEs associated with the subsystem X can be written as

$$\frac{\mathrm{d}X}{\mathrm{d}t} = A(t)X \tag{6.72}$$

where $A(t)$ is related to $\alpha(t)$. The solution of (6.72) is $X(t) = e^{tB(t)}X(0)$, where $B(t)$ is defined by $B(t) = \frac{1}{t}\int_0^t A(s)ds$, and thus $B(t)$ is related to $\beta(t)$.

Notice that according to Theorem 6.5.4 and its proof, the eigenvalues of $B(t)$ other than zeros have negative real parts for any $t > 0$. By a similar proof to Corollary 6.4.3, we have

$$\Lambda = \inf_{t \geq 0} \{-\Re(\lambda) \mid \lambda \text{ is } B(t)\text{'s non-zero eigenvalue}\} > 0. \tag{6.73}$$

This fact will lead to the conclusion that $X(t)$ can be approximated by $\hat{X}(t)$, where $\hat{X}(t)$ is constructed similarly to the one in Proposition 6.4.2. Because a general $B(t)$ considered here may not be diagonalisable, so the construction of $\hat{X}(t)$ is a little bit more complicated. We

detail the construction as well as the proof of the following result in Appendix D.2.2:

$$\lim_{t\to\infty} \|X(t) - \hat{X}(t)\| = 0.$$

Then, by similar arguments to Proposition 6.4.3 and Corollary 6.4.4, we can prove that $\inf_{t>T} \hat{x}_1(t) \geq cM$, where $\hat{x}_1(t)$ is the first entry of $\hat{X}(t)$. Then, by a similar proof to the proof of Lemma 6.4.5, we can conclude that there exists a number c_1 such that $x_1(t) \geq c_1 M$ after a time T. □

Lemma 6.5.6. *Under the assumptions of Lemma 6.5.5, if $M > K_1 N$ or $N > K_2 M$, where constants $K_1 > 0$ and $K_2 > 0$ are sufficiently large, then there exists T such that $rx_1(t) \geq sy_1(t),\ \forall t \geq T$ or $rx_1(t) \leq sy_1(t),\ \forall t \geq T$ respectively.*

Proof. By the boundedness of solutions, $0 \leq x_1(t) \leq M$ and $0 \leq y_1(t) \leq N$ for any t. Suppose $M > K_1 N$. By Lemma 6.5.5, there exists $c, T > 0$, $x_1(t) \geq cM,\ \forall t \geq T$. Since K_1 is assumed to be large enough such that $K_1 \geq \frac{s}{rc}$, then $rcM > sN$. So for any $t > T$, we have

$$rx_1(t) \geq rcM \geq sN \geq sy_1(t).$$

If $N > K_2 M$, the proof is similar and omitted here. □

Now we state our convergence theorem.

Theorem 6.5.7. *If $M > K_1 N$ or $N > K_2 M$, where constants $K_1, K_2 > 0$ are sufficiently large, then the solution of the derived ODEs (6.56), i.e.*

$$\begin{pmatrix} \frac{\mathrm{d}X}{\mathrm{d}t} \\ \frac{\mathrm{d}Y}{\mathrm{d}t} \end{pmatrix} = I_{\{rx_1 \leq sy_1\}} Q_1 \begin{pmatrix} X \\ Y \end{pmatrix} + I_{\{rx_1 > sy_1\}} Q_2 \begin{pmatrix} X \\ Y \end{pmatrix},$$

converges to a finite limit as time goes to infinity.

Proof. Suppose $M > K_1 N$, then by Lemma 6.5.6, there exists a time $T > 0$, such that after time T, $rx_1(t) \geq sy_1(t)$, so (6.56) becomes

$$\begin{pmatrix} \frac{\mathrm{d}X}{\mathrm{d}t} \\ \frac{\mathrm{d}Y}{\mathrm{d}t} \end{pmatrix} = Q_2 \begin{pmatrix} X \\ Y \end{pmatrix}. \tag{6.74}$$

Since Q_2's eigenvalues other than zeros have strict negative real parts according to Theorem 6.5.4, and the solution of the above equation is bounded, then by Corollary D.2.3, the solution converges to a finite limit as time goes to infinity. □

Remark 6.5.4. *We should point out that for a general PEPA model with two component types and one synchronisation, the limit of the solution of the derived ODEs is determined by the populations of these two component types rather than the particular starting state. That is to say, whatever the initial state is, as long as the total populations of the components are the same, the limit that the solution converges to will always be the same. We do not plan to discuss this topic in detail in this thesis.*

6.6 Summary

In this chapter, we have presented our investigations in the fluid approximation for PEPA, from an analytical perspective. The analytical proofs for the convergence of the solutions of the ODEs derived from some PEPA models, have been demonstrated. In particular, the case study of an interesting model, i.e. Model 3, has shown that the structural property of invariance can be used to prove the convergence. Moreover, for the class of models with two component types and one synchronisation, the converge has been proved under some mild conditions. In addition, this chapter has also presented an analytical proof for the boundedness of the solutions for any general PEPA model.

Chapter 7

Deriving Performance Measures for Large Scale Content Adaptation Systems

7.1 Introduction

In the previous chapters, we have presented the techniques and theoretical results developed for the PEPA language. This chapter will show how to derive performance measures from large scale PEPA models based on the previous developments, as well as demonstrate the application of these developments to large scale content adaptation systems proposed by the Mobile VCE.

In this chapter, numerical experiments are used to show the properties of the fluid approximation of the content adaptation PEPA model, with an emphasis on the convergence and consistence characteristics. Then, we will discuss what kind of performance measures can be derived through this fluid approximation approach. In order to obtain the performance metrics that cannot be derived through this approach, we propose a Gillespie stochastic simulation algorithm based on our numerical representation scheme of PEPA. Elaborated comparisons of the different approaches for deriving performance measures from content adaptation systems will be subsequently presented.

Finally, for the purpose of demonstrating the performance derivation and evaluation for large scale PEPA models through the fluid approximation approach, this chapter will detail the numerical solutions of the ODEs derived from the content adaptation model, assessing the sensitivity of the framework to the performance of individual components, and the scalability of the framework under increasing loads and different resource conditions. Structural analysis of a content adaptation system, in terms of invariance analysis and deadlock-checking for a subsystem, as an application of the techniques developed in Chapter 4, is also briefly discussed in this chapter.

7.2 Fluid Approximation of the PEPA Model of Content Adaptation Systems

In Chapter 2, we have presented the framework of content adaptation proposed by the Mobile VCE. A working cycle based on the logical architecture has been modelled using the PEPA language. Some performance evaluation based on the model at small scale has been conducted and demonstrated. This section will introduce the fluid approximation of the PEPA model of the content adaptation system and demonstrate its use for performance analysis. Moreover, numerical experiments will be used to intuitively illustrate the characteristics of fluid approximations.

7.2.1 ODEs derived from the PEPA model of content adaptation systems

In Chapter 2, the PEPA model of the content adaptation system based on the framework put forward by the Mobile VCE, has already been constructed. For convenience, we show the model again:

PDE:

$$\begin{aligned}
PDE_1 &\stackrel{def}{=} (pde_ext_cont_req, r_{pde_ext_cont_req}).PDE_2 \\
PDE_2 &\stackrel{def}{=} (pde_int_cont_req, r_{pde_int_cont_req}).PDE_3 \\
PDE_3 &\stackrel{def}{=} (csp_to_pde, \top).PDE_4 \\
&\quad + \quad (ca_to_pde, \top).PDE_4 \\
PDE_4 &\stackrel{def}{=} (pde_user_interface, r_{pde_user_interface}).PDE_1
\end{aligned}$$

AM:

$$\begin{aligned}
AM_1 &\stackrel{def}{=} (pde_int_cont_req, \top).AM_2 \\
AM_2 &\stackrel{def}{=} (csp_cc_req, r_{csp_cc_req}).AM_3 \\
AM_3 &\stackrel{def}{=} (csp_cc_res, \top).AM_4 \\
AM_4 &\stackrel{def}{=} (am_assimilation, \tfrac{1}{2}r_{am_assimilation}).AM_5 \\
&\quad + \quad (am_assimilation, \tfrac{1}{2}r_{am_assimilation}).AM_9 \\
AM_5 &\stackrel{def}{=} (ca_states_req, r_{ca_states_req}).AM_6 \\
AM_6 &\stackrel{def}{=} (ca_states_res, \top).AM_7 \\
AM_7 &\stackrel{def}{=} (am_decision, r_{am_decision}).AM_8 \\
AM_8 &\stackrel{def}{=} (am_adapt_plan, r_{am_adapt_plan}).AM_1 \\
AM_9 &\stackrel{def}{=} (am_cont_req, r_{am_cont_req}).AM_1
\end{aligned}$$

CA:

$$\begin{aligned}
CA_1 &\stackrel{def}{=} (ca_states_req, \top).CA_2 \\
CA_2 &\stackrel{def}{=} (ca_states_res, r_{ca_states_res}).CA_3 \\
CA_3 &\stackrel{def}{=} (csp_call_ca_adapt, \top).CA_4 \\
CA_4 &\stackrel{def}{=} (ca_adaptation, r_{ca_adaptation}).CA_5 \\
CA_5 &\stackrel{def}{=} (ca_to_pde, r_{ca_to_pde}).CA_1
\end{aligned}$$

C/S Provider:

$$\begin{aligned}
CSP_1 &\stackrel{def}{=} (csp_cc_req, \top).CSP_2 \\
CSP_2 &\stackrel{def}{=} (csp_cc_res, r_{csp_cc_res}).CSP_3 \\
CSP_3 &\stackrel{def}{=} (am_cont_req, \top).CSP_4 \\
&\quad + \ (am_adapt_plan, \top).CSP_5 \\
CSP_4 &\stackrel{def}{=} (csp_to_pde, r_{csp_to_pde}).CSP_1 \\
CSP_5 &\stackrel{def}{=} (csp_call_ca_adapt, r_{csp_call_ca_adapt}).CSP_1
\end{aligned}$$

System Equation:

$$PDE_1[M] \underset{L_1}{\bowtie} \left(\left(AM_1[N] \underset{L_2}{\bowtie} CA_1[P] \right) \underset{L_3}{\bowtie} CSP_1[Q] \right),$$

where

$$\begin{aligned}
L_1 &= \{pde_int_cont_req, ca_to_pde, csp_to_pde\}, \\
L_2 &= \{ca_states_req, ca_states_res\}, \\
L_3 &= \{csp_cc_req, csp_cc_res, am_cont_req, am_adapt_plan, csp_call_ca_adapt\}.
\end{aligned}$$

Notice that in this chapter, the numbers of independent copies of all entities in the system, which are represented by M, N, P, and Q respectively, are variables of some of our experiments. The parameter settings have already been presented in Table 2.2 in Chapter 2, so we omit them here.

According to the mapping semantics presented in Chapter 3, the ODEs derived from our PEPA model are as follows.

PDE:

$$
\begin{aligned}
\frac{\mathrm{d}PDE_1}{\mathrm{d}t} &= -r_{pde\_ext\_cont\_req}PDE_1 + r_{pde\_user\_interface}PDE_4 \\
\frac{\mathrm{d}PDE_2}{\mathrm{d}t} &= -\min\{r_{pde\_int\_cont\_req}PDE_2, AM_1\top\} + r_{pde\_ext\_cont\_req}PDE_1 \\
\frac{\mathrm{d}PDE_3}{\mathrm{d}t} &= -\min\{PDE_3\top, r_{ca\_to\_pde}CA_5\} - \min\{PDE_3\top, r_{csp\_to\_pde}CSP_4\} \\
&\quad + \min\{r_{pde\_int\_cont\_req}PDE_2, AM_1\top\} \\
\frac{\mathrm{d}PDE_4}{\mathrm{d}t} &= \min\{PDE_3\top, r_{ca\_to\_pde}CA_5\} + \min\{PDE_3\top, r_{csp\_to\_pde}CSP_4\} \\
&\quad - r_{pde\_user\_interface}PDE_4.
\end{aligned}
$$

AM:

$$
\begin{aligned}
\frac{\mathrm{d}AM_1}{\mathrm{d}t} &= -\min\{r_{pde\_int\_cont\_req}PDE_2, AM_1\top\} \\
&\quad + \min\{r_{am\_adaptation\_plan}AM_8, CSP_3\top\} + \min\{r_{am\_cont\_req}AM_9, CSP_3\top\} \\
\frac{\mathrm{d}AM_2}{\mathrm{d}t} &= \min\{r_{pde\_int\_cont\_req}PDE_2, AM_1\top\} - \min\{r_{csp\_cc\_req}AM_2, CSP_1\top\} \\
\frac{\mathrm{d}AM_3}{\mathrm{d}t} &= -\min\{r_{csp\_cc\_res}CSP_2, AM_3\top\} + \min\{r_{csp\_cc\_req}AM_2, CSP_1\top\} \\
\frac{\mathrm{d}AM_4}{\mathrm{d}t} &= \min\{r_{csp\_cc\_res}CSP_2, AM_3\top\} - r_{am\_assimilation}AM_4 \\
\frac{\mathrm{d}AM_5}{\mathrm{d}t} &= -\min\{r_{ca\_states\_req}AM_5, CA_1\top\} + \frac{1}{2}r_{am\_assimilation}AM_4 \\
\frac{\mathrm{d}AM_6}{\mathrm{d}t} &= \min\{r_{ca\_states\_req}AM_5, CA_1\top\} - \min\{r_{ca\_states\_res}CA_2, AM_6\top\} \\
\frac{\mathrm{d}AM_7}{\mathrm{d}t} &= -r_{am\_decision}AM_7 + \min\{r_{ca\_states\_res}CA_2, AM_6\top\} \\
\frac{\mathrm{d}AM_8}{\mathrm{d}t} &= r_{am\_decision}AM_7 - \min\{r_{am\_adaptation\_plan}AM_8, CSP_3\top\} \\
\frac{\mathrm{d}AM_9}{\mathrm{d}t} &= \frac{1}{2}r_{am\_assimilation}AM_4 - \min\{r_{am\_cont\_req}AM_9, CSP_3\top\}
\end{aligned}
$$

CA:

$$\begin{aligned}
\frac{\mathrm{d}CA_1}{\mathrm{d}t} &= -\min\{r_{ca_states_req}AM_5, CA_1\top\} + \min\{PDE_3\top, r_{ca_to_pde}CA_5\} \\
\frac{\mathrm{d}CA_2}{\mathrm{d}t} &= \min\{r_{ca_states_req}AM_5, CA_1\top\} - \min\{r_{ca_states_res}CA_2, AM_6\top\} \\
\frac{dCA_3}{\mathrm{d}t} &= \min\{r_{ca_states_res}CA_2, AM_6\top\} - \min\{r_{csp_call_ca_adapt}CSP_5, CA_3\top\} \\
\frac{\mathrm{d}CA_4}{\mathrm{d}t} &= \min\{r_{csp_call_ca_adapt}CSP_5, CA_3\top\} - r_{ca_adaptation}CA_4 \\
\frac{\mathrm{d}CA_5}{\mathrm{d}t} &= r_{ca_adaptation}CA_4 - \min\{PDE_3\top, r_{ca_to_pde}CA_5\}
\end{aligned}$$

C/S Provider:

$$\begin{aligned}
\frac{\mathrm{d}CSP_1}{\mathrm{d}t} &= -\min\{r_{csp_cc_req}AM_2, CSP_1\top\} \\
&\quad + \min\{r_{csp_call_ca_adapt}CSP_5, CA_3\top\} + \min\{PDE_3\top, r_{csp_to_pde}CSP_4\} \\
\frac{\mathrm{d}CSP_2}{\mathrm{d}t} &= \min\{r_{csp_cc_req}AM_2, CSP_1\top\} - \min\{r_{csp_cc_res}CSP_2, AM_3\top\} \\
\frac{\mathrm{d}CSP_3}{\mathrm{d}t} &= \min\{r_{csp_cc_res}CSP_2, AM_3\top\} \\
&\quad - \min\{r_{am_adaptation_plan}AM_8, CSP_3\top\} - \min\{r_{am_cont_req}AM_9, CSP_3\top\} \\
\frac{\mathrm{d}CSP_4}{\mathrm{d}t} &= -\min\{PDE3\top, r_{csp_to_pde}CSP_4\} + \min\{r_{am_cont_req}AM_9, CSP_3\top\} \\
\frac{\mathrm{d}CSP_5}{\mathrm{d}t} &= -\min\{r_{csp_call_ca_adapt}CSP_5, CA_3\top\} + \min\{r_{am_adaptation_plan}AM_8, CSP_3\top\}
\end{aligned}$$

The initial condition of these ODEs is determined by the system equation, i.e., $PDE_1(0) = M$, $AM_1(0) = N$, $CA_1(0) = P$, $CSP_1(0) = Q$ and all other entries are zeros. As we have mentioned in Chapter 5, the terms such as "$\min\{A\top, rB\}$" in the above ODEs are interpreted as [BGH07]:

$$\min\{A\top, rB\} = \begin{cases} rB, & A > 0, \\ 0, & A = 0. \end{cases}$$

7.2.2 The properties of the solution of the derived ODEs

In this subsection, we will demonstrate numerical experiments based on the ODEs given in the previous subsection. As we have pointed out, the above definition of "$\min\{A\top, rB\}$" may result in jumps in the derivation functions on the right side of the ODEs. Then according to the theory of ordinary differential equations, these ODEs may have no analytic solutions. That is,

the existence and uniqueness theorem presented in Chapter 5 cannot apply to these ODEs. In this case, in order to guarantee the existence of solutions, these ODEs should be considered as difference rather than differential equations.

Some numerical solutions to these equations under different initial conditions, which are obtained by the Euler difference method, are presented in Figure 7.1, 7.2, 7.3, and 7.4. Here and in the following stochastic simulations the activity rates are specified as shown in Table 2.2 in Chapter 2, as we have mentioned in Section 7.2.1. As we can see from these figures (solid blue curve), the solutions are nonnegative, as well as bounded between zero and the total populations of the corresponding component types. This is consistent with the nonnegativeness and boundedness theorem, i.e. Theorem 6.2.1, in the previous chapter. Moreover, the curves of these solutions become flat after a finite time, e.g. time eight. So there is reason to believe that they tend to finite limits as time goes to infinity, which is consistent with the convergence theorem presented in Chapter 5, i.e. Theorem 5.6.3.

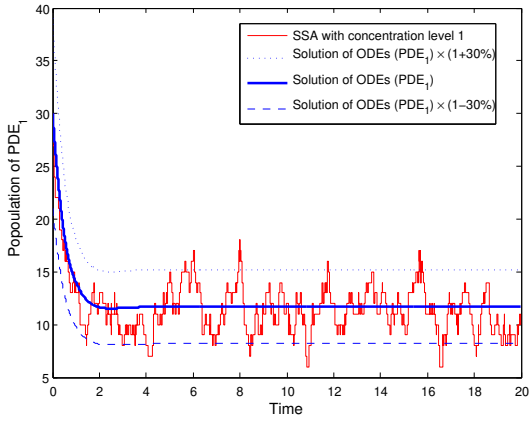
Now we use numerical experiments to illustrate another kind of convergence, which is with respect to the *concentration level* rather than time. Let $\mathbf{x}(t)$ be the solution of the ODEs given in the previous subsection, and $\mathbf{x}_0$ be the initial condition, i.e. the starting state of the PEPA model. Denote by $\{X_n(t)\}_n$ the density dependent CTMCs associated with $\mathbf{x}_0$ underlying the model. So $X_n(0) = n\mathbf{x}_0$ for any n. Let $\hat{X}_n(t) = \frac{X_n(t)}{n}$, i.e. the underlying *concentrated* density dependent CTMCs. Then for any concentration level n, $\hat{X}_n(0) = \mathbf{x}_0$. That is, whatever the concentration level is, the starting state of a family of concentrated density dependent CTMCs is always the same, and so is the population of each entity in the CTMCs. Moreover, for the CTMC $\hat{X}_n(t)$, the step of increment or decrement of the entries within the numerical vector states is $\frac{1}{n}$, while it is always one for $X_n(t)$. For convenience, unless otherwise mentioned, the CTMCs in the following refer to the concentrated density dependent CTMCs. According to Theorem 5.4.1 on page 124, for any $t > 0$,

$$\lim_{n\to\infty} \sup_{u\leq t} \|\hat{X}_n(u) - \mathbf{x}(u)\| = 0 \qquad a.s.$$

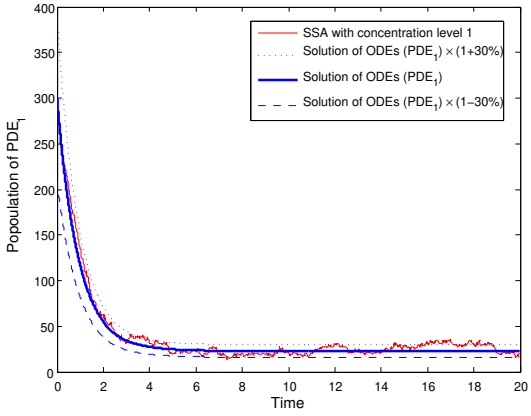
That is, any single path or trajectory of the CTMC $\hat{X}_n(t)$ converges to the solution of the ODEs as the concentration level n tends to infinity. This fact is also illustrated by Figure 7.1 – 7.4. For example, in the scenario of 30 PDEs and the population[1] of any other entity being

[1]This kind of scenario or population is determined by the starting state of the PEPA model, which is the same as the starting states of the corresponding concentrated density dependent CTMCs.

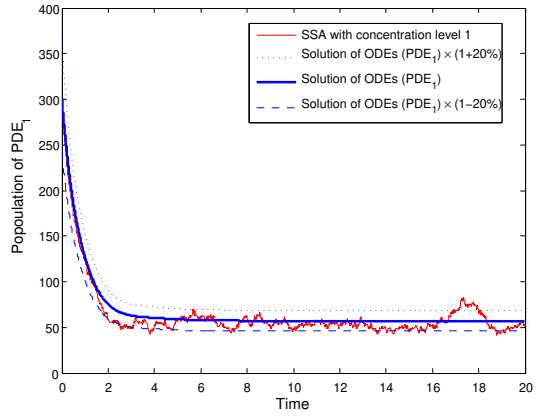
20, if the concentration level is under ten, there is substantial difference between the single paths of the CTMC and the solutions of the ODEs. In fact, the paths generally fluctuate by no less than 20% from the solution, see Figure 7.1 (a), Figure 7.2 (b), Figure 7.3 (a) and (b). For convenience, in this chapter this kind of deviation is called the *percentage error*, which measures the relative error between the CTMC and the ODEs and reflects the extent of the approximation between them. Each path (at any time) is in fact a vector, but for convenience, only the entry corresponding to PDE_1 is used for comparison. As the concentration level increases to 20, the percentage error decreases to 10%. When the concentration level reaches 50, the CTMC generally does not deviate more than 5% from the ODE solution, i.e., it fluctuates around the solution with the percentage error 5%. See Figure 7.3 (c) and (d).



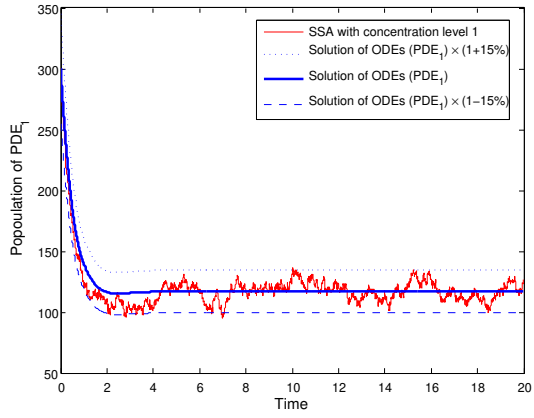
(a) $(M, N, P, Q) = (30, 20, 20, 20)$, deviation 30%



(b) $(M, N, P, Q) = (300, 20, 20, 20)$, deviation 30%



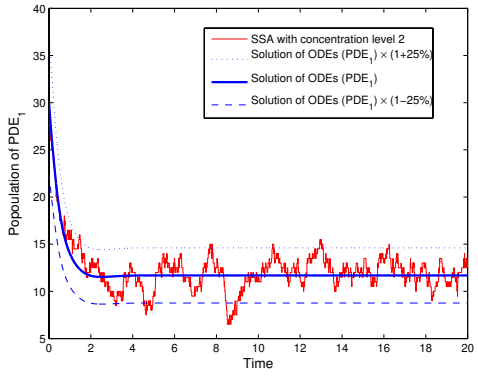
(c) $(M, N, P, Q) = (300, 50, 50, 50)$, deviation 20%



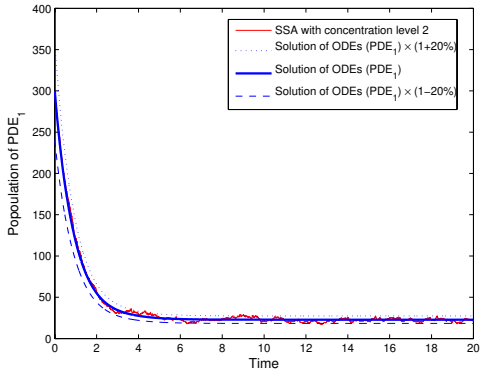
(d) $(M, N, P, Q) = (300, 200, 200, 200)$, deviation 15%

Figure 7.1: *Concentrated density dependent CTMCs (concentration level one) approximate the ODEs*

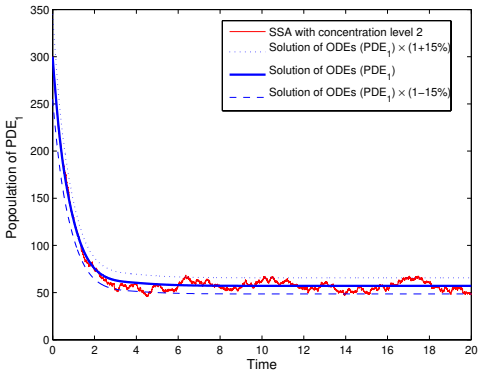
If the concentration level is fixed at one, i.e. $n = 1$, then the percentage errors between the



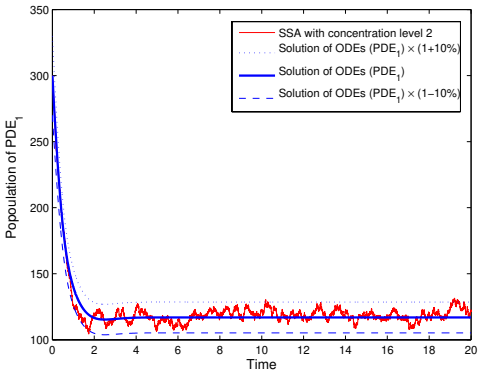
(a) $(M, N, P, Q) = (30, 20, 20, 20)$, deviation 25%



(b) $(M, N, P, Q) = (300, 20, 20, 20)$, deviation 20%



(c) $(M, N, P, Q) = (300, 50, 50, 50)$, deviation 15%

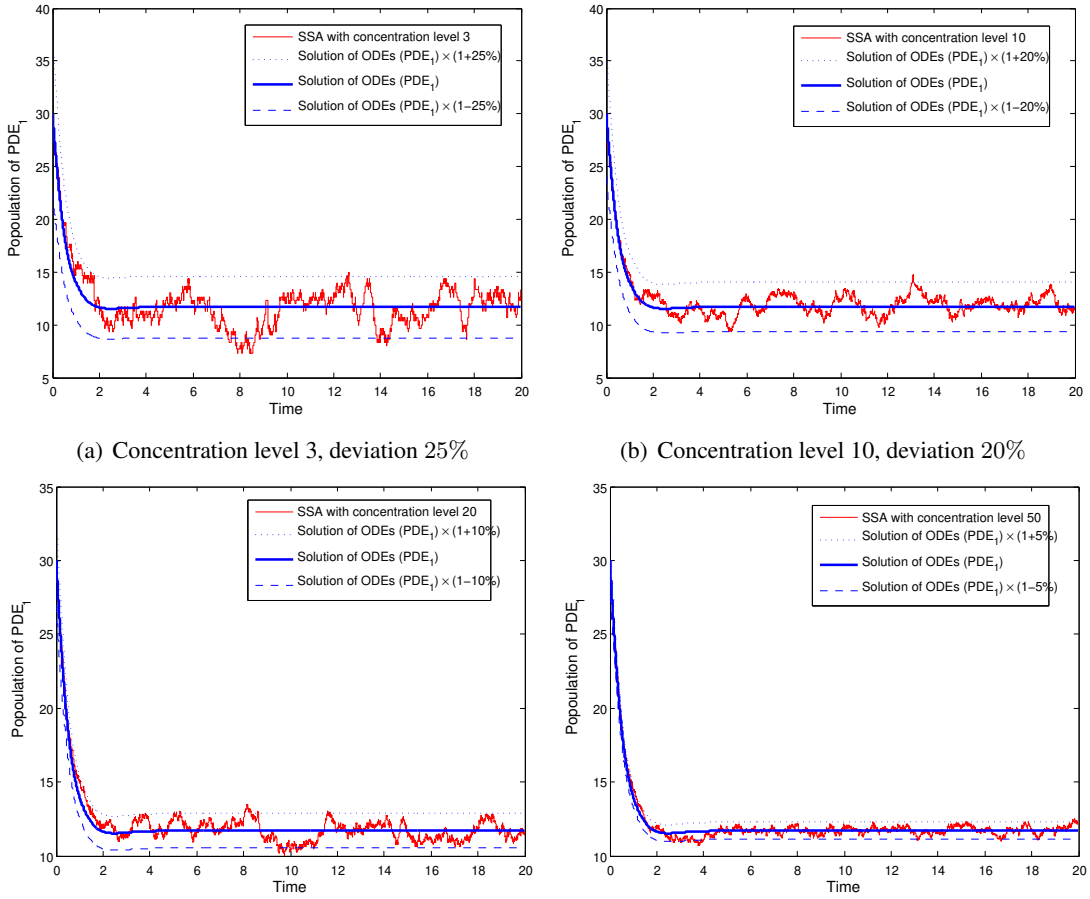


(d) $(M, N, P, Q) = (300, 200, 200, 200)$, deviation 10%

Figure 7.2: *Concentrated density dependent CTMCs (concentration level two) approximate the ODEs*

Level n	$(M, N, P, Q) =$ (30,20,20,20)	$(M, N, P, Q) =$ (300,20,20,20)	$(M, N, P, Q) =$ (300,50,50,50)	$(M, N, P, Q) =$ (300,200,200,200)
1	$\sim 30\%$	$\sim 30\%$	$\sim 20\%$	$\sim 15\%$
2	$\sim 25\%$	$\sim 20\%$	$\sim 15\%$	$\sim 10\%$
3	$\sim 25\%$	$\sim 20\%$	$\sim 15\%$	$\sim 10\%$
10	$\sim 15\%$	$\sim 10\%$	$\sim 5\%$	$< 5\%$
20	$\sim 10\%$	$\sim 5\%$	$< 5\%$	$< 5\%$
50	$\sim 5\%$	$< 5\%$	$< 5\%$	$< 5\%$

Table 7.1: *Percentage error between CTMCs and ODEs*



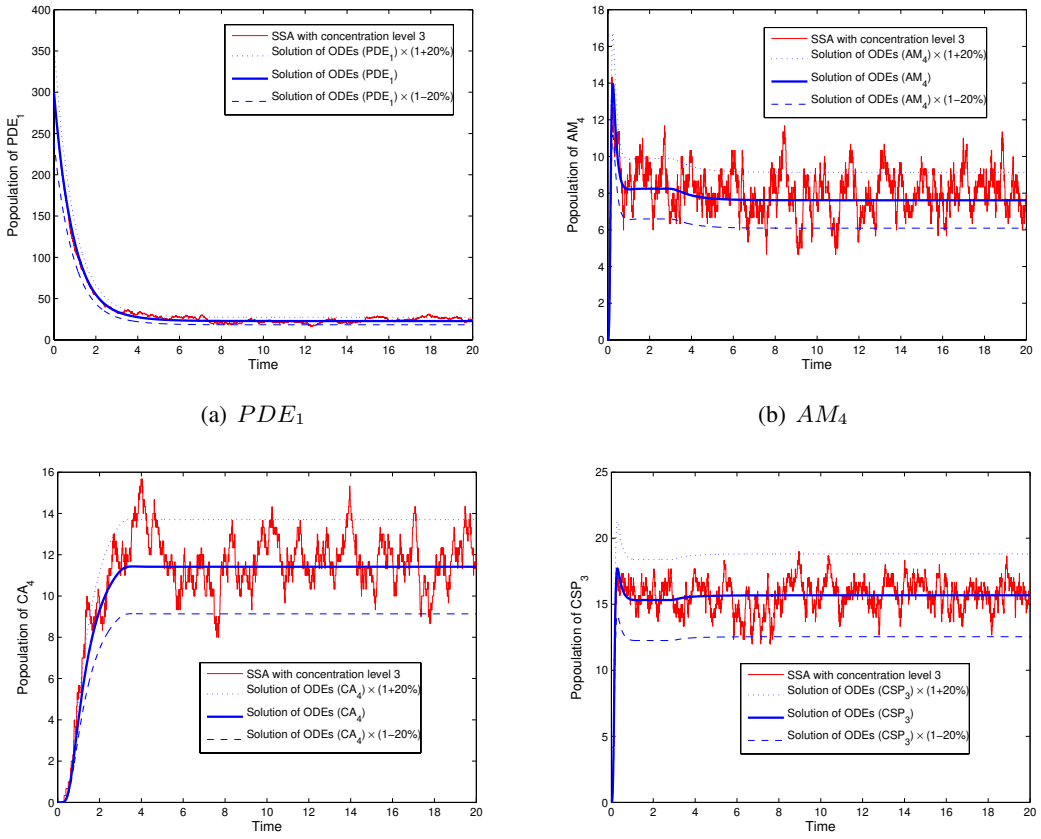
(a) Concentration level 3, deviation 25%

(b) Concentration level 10, deviation 20%

(c) Concentration level 20, deviation 10%

(d) Concentration level 50, deviation 5%

Figure 7.3: *Concentrated density dependent CTMCs approximate the ODEs* ($(M, N, P, Q) = (30, 20, 20, 20)$*)*



(a) PDE_1 (b) AM_4

(c) CA_4 (d) CSP_3

Figure 7.4: *Concentrated density dependent CTMCs (concentration level three) approximate the ODEs ($(M, N, P, Q) = (300, 20, 20, 20)$)*

paths and the solutions are about 30%, when there are 30 and 300 PDEs in the system while the population of each other entity is fixed at 20. See Figure 7.1 (a) and Figure 7.1 (b). That is, there is not much difference with respect to deviation between the cases of 30 PDEs and 300 PDEs in that scenario. But when $n = 2$, in the same scenario, the percentage errors are obviously different since these errors are 25% and 20% respectively, see Figure 7.2 (a) and Figure 7.2 (b). When the population of each other entity increases to 50 and further to 200 while the number of PDEs is fixed at 300, then the percentage errors significantly decrease regardless of whether $n = 1$ or $n = 2$, see Figure 7.1 (c) and (d), as well as Figure 7.2 (c) and (d). That is, the population size on the server side (N, P, Q) has a significant effect on the approximation accuracy. Moreover, these suggest that the smallest population of the entities has more effect on percentage errors than the maximum of the populations has. This observation can be supported by further experimental results that are presented in Table 7.1 (on page 188). In this table, M, N, P, Q represent the numbers of PDE_1, AM_1, CA_1 and CSP_1 in the starting state. As illustrated by this table, as the population increases (both in terms of the smallest and maximum of the populations), the level needed to achieve percentage error of approximately 5% or less will decrease.

The evolution of the populations of some other entities, such as AM_4, CA_4 and CSP_3, in the scenario $(M, N, P, Q) = (300, 20, 20, 20)$, has been demonstrated in Figure 7.4. These entities have a similar percentage error to the one of PDE_1.

We should point out that the percentage error discussed here is in terms of the difference between one-realisation (i.e. one run) of the CTMC and the solution of the ODEs. There is no doubt that, the "averaged path" in each figure should have much smaller deviations. This phenomenon can be theoretically explained. For example, let $\{\xi_i\}_{i=1}^n$ be a family of random variables and each of them has the same variance, namely σ^2. Then the average of $\{\xi_i\}_{i=1}^n$, i.e. $\frac{1}{n}\sum_{i=1}^n \xi_i$ has a reduced variance $\frac{\sigma^2}{n}$, provided that these random variables are independent. An averaged path can be considered as the average of multiple trajectories of the CTMC, which therefore should have a reduced variance. Thus, the concentration level needed to achieve a percentage error with respect to average path should be smaller than the error with respect to single path.

7.3 Deriving Quantitative Performance Measures through Different Approaches

In Section 2.4 of Chapter 2, we have discussed how to derive performance measures from small scale PEPA models, through the approach of solving global balance, linear algebraic equations. This approach is not feasible for large scale models due to the state-space explosion problem. However, we have alternative ways to obtain quantitative performance measures from a large scale PEPA model, which will be discussed in this section.

7.3.1 Deriving performance measures through fluid approximation approach

In this subsection, we will discuss what kind of performance measures can be derived through the fluid approximation approach and how these performance measures can be derived from PEPA models. As illustrated in Chapter 5, for a given PEPA model, there is a family of density dependent CTMCs, namely, $X_n(t)$, underlying this model. Let $\hat{X}_n(t)$ be the concentrated density dependent CTMCs, i.e. $\hat{X}_n(t) = \frac{X_n(t)}{n}$. Denote the expectation of $\hat{X}_n(t)$ by $\hat{M}_n(t)$, that is,

$$\hat{M}_n(t) = E[\hat{X}_n(t)] = \sum_{\mathbf{s} \in \hat{S}_n} \mathbf{s}\hat{\pi}_t^n(\mathbf{s}),$$

where $\hat{S}_n$ and $\hat{\pi}_t^n(\cdot)$ are the state space and the transient probability distribution of $\hat{X}_n(t)$, respectively. Let

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = F(\mathbf{x}) = \sum_{l \in \mathcal{A}_{\mathrm{label}}} l f(\mathbf{x}, l)$$

be the ODEs derived from this model. Then according to Lemma 5.5.1 in Chapter 5,

$$\lim_{n\to\infty} \hat{M}_n(t) = \lim_{n\to\infty} E[\hat{X}_n(t)] = \mathbf{x}(t). \tag{7.1}$$

Moreover, as Theorem 5.6.3 states, under the stated condition, the solution $\mathbf{x}(t)$ of the ODEs converges to a limit $\mathbf{x}^*$ as time tends to infinity, and satisfies

$$\lim_{t\to\infty} \lim_{n\to\infty} \hat{M}_n(t) = \lim_{t\to\infty} \mathbf{x}(t) = \mathbf{x}^*. \tag{7.2}$$

That is,

$$\lim_{t\to\infty} \lim_{n\to\infty} E[\hat{X}_n(t)] = \mathbf{x}^*. \tag{7.3}$$

According to (7.1) and (7.3), the fluid approximation of a PEPA model captures the information of the *first-order moments* of the underlying CTMCs. Correspondingly, from the perspective of performance measure, all performance metrics that only depend on the first-order moment information can be derived through this fluid approximation approach. These metrics include: the throughput of an acitivity, utilisation, average response time. However, if a performance metric needs or relates the information of higher-order moments, then this approach may not be sufficient to derive this performance. In the following, we will mathematically discuss this problem.

For the CTMC $\hat{X}_n(t)$ given above, some averaged transient performance measure $\xi(n,t)$ can be expressed by

$$\xi(n,t) = E[\rho(\hat{X}_n(t))] = \sum_{\mathbf{s}\in\hat{S}_n} \rho(\mathbf{s})\hat{\pi}_t^n(\mathbf{s}),$$

where ρ is a reward function. Some discussions about reward functions have been presented in Chapter 2. If ρ is a **linear** function, then this performance measure can be derived through the fluid approximation approach. See the following proposition.

Proposition 7.3.1. *For a given PEPA model, let $\{\hat{X}_n(t)\}_n$ be the family of concentrated density dependent CTMCs underlying this model. Let $\mathbf{x}(t)$ be the solution of the ODEs derived from this model. Then for $\xi(n,t) = E[\rho(\hat{X}_n(t))]$, where ρ is a linear function, we have*

$$\lim_{n\to\infty} E[\rho(\hat{X}_n(t))] = \rho(\mathbf{x}(t)). \tag{7.4}$$

Moreover, if $\lim_{t\to\infty} \mathbf{x}(t) = \mathbf{x}^*$, then

$$\lim_{t\to\infty}\lim_{n\to\infty} E[\rho(\hat{X}_n(t))] = \rho(\mathbf{x}^*). \tag{7.5}$$

Proof. Because ρ is linear, so we can exchange the notations of the expectation and the function ρ (Proposition 2.4, [BZ99]), that is $E[\rho(\cdot)] = \rho(E[\cdot])$. Therefore,

$$\lim_{n\to\infty} E[\rho(\hat{X}_n(t))] = \lim_{n\to\infty} \rho(E[\hat{X}_n(t)]).$$

Since ρ is linear and thus continuous, so $\lim_{n\to\infty} \rho(E[\hat{X}_n(t)]) = \rho(\lim_{n\to\infty} E[\hat{X}_n(t)])$. Then by (7.1), $\lim_{n\to\infty} E[\hat{X}_n(t)] = \mathbf{x}(t)$, so we have $\lim_{n\to\infty} E[\rho(\hat{X}_n(t))] = \rho(\mathbf{x}(t))$, which completes the first assertion.

Moreover, if $\lim_{t\to\infty} \mathbf{x}(t) = \mathbf{x}^*$, we have

$$\begin{aligned}
\lim_{t\to\infty}\lim_{n\to\infty} \xi(n,t) &= \lim_{t\to\infty}\lim_{n\to\infty} E[\rho(\hat{X}_n(t))] \\
&= \lim_{t\to\infty}\lim_{n\to\infty} \rho(E[\hat{X}_n(t)]) \\
&= \rho\left(\lim_{t\to\infty}\lim_{n\to\infty} E[\hat{X}_n(t)]\right) \\
&= \rho\left(\lim_{t\to\infty} \mathbf{x}(t)\right) \\
&= \rho(\mathbf{x}^*).
\end{aligned}$$

In above formula, the second "=" holds because $E[\rho(\cdot)] = \rho(E[\cdot])$ for a linear function ρ. The third "=" holds due to the continuity of ρ. □

We should point out that if ρ is not a linear function, then Proposition 7.3.1 may not hold. This is because, for a general function, we may not have the property $E[\rho(\cdot)] = \rho(E[\cdot])$ which is needed in the proof of Proposition 7.3.1. In fact, if ρ is a convex function, then $E[\rho(\cdot)] \geq \rho(E[\cdot])$ (Theorem 2.2 (Jensen's inequality) [BZ99]), so both (7.4) and (7.5) may not hold. However, in this case we can still benefit from the fluid approximation: the value calculated from the ODEs gives an upper bound on the performance measure. As we have mentioned, the convergence theorem, i.e. Theorem 5.6.3, captures the information of the first-order moments of the CTMCs. A linear mapping of the first-order is still first-order, and this first-order information can be provided by the ODEs. That is why a performance measure that can be defined through a linear reward function, can be derived through the fluid approximation.

As we have discussed in Chapter 2, many performance measures of interest can be represented through a linear reward function. For example, the average throughput of the activity $ca\_adaptation$ in the CTMC $\hat{X}_n(t)$ underlying the content adaptation model, denoted by $\mathrm{Thr}(ca\_adaptation, n, t)$, is

$$\mathrm{Thr}(ca\_adaptation, n, t) = \sum_{\mathbf{s}\in\hat{S}_n} (\mathbf{s}[CA_4] r_{ca\_adaptation})\, \hat{\pi}_t^n(\mathbf{s}), \tag{7.6}$$

where $\mathbf{s}[CA_4]$ indicates the population of CA_4 in the state $\mathbf{s}$. Let $\rho_1(\mathbf{s}) = \mathbf{s}[CA_4] r_{ca\_adaptation}$, then

$$\mathrm{Thr}(ca\_adaptation, n, t) = E[\rho_1(\hat{X}_n(t))]. \tag{7.7}$$

Since $\rho_1(\mathbf{s})$ is linear with respect to $\mathbf{s}$, we can conclude that

$$\lim_{t\to\infty}\lim_{n\to\infty}\mathrm{Thr}(ca\_adaptation,n,t)=\mathbf{x}^*[CA_1]r_{ca\_adaptation}\stackrel{def}{=}\mathrm{Thr}^*(ca\_adaptation). \tag{7.8}$$

Here $\mathrm{Thr}^*(ca\_adaptation)$ reflects the throughput of $ca\_adaptation$ in the equilibrium state, i.e. the limit of the transient throughput as time as well as the concentration level tends to infinity.

In addition to the throughput of activities, utilisation can also be derived through the fluid approximation approach. For example, the utilisation of the idle state CA_1 is defined as

$$\mathrm{Util}(CA_1,n,t)=\sum_{\mathbf{s}\in\hat{S}_n}\left(\frac{\mathbf{s}[CA_1]}{N_{CA}}\right)\hat{\pi}_t^n(\mathbf{s}), \tag{7.9}$$

where N_{CA} is the total population of the component CA. Similarly, if we let $\rho_2(\mathbf{s})=\frac{\mathbf{s}[CA_1]}{N_{CA}}$, then ρ_2 is linear and

$$\mathrm{Util}(CA_1,n,t)=E[\rho_2(\hat{X}_n(t))]. \tag{7.10}$$

According to Proposition 7.3.1,

$$\lim_{t\to\infty}\lim_{n\to\infty}\mathrm{Util}(CA_1,n,t)=\frac{\mathbf{x}^*[CA_1]}{N_{CA}}\stackrel{def}{=}\mathrm{Util}^*(CA_1). \tag{7.11}$$

Unfortunately, this approach cannot be used to derive the measure of "variation of the utilisation of CA_1", which is defined by

$$\sum_{\mathbf{s}\in S}\left(\frac{\mathbf{s}[CA_1]}{N_{CA}}-\frac{\mathbf{x}^*[CA_1]}{N_{CA}}\right)^2\pi(\mathbf{s}).$$

This is because information about the second-order moment such as $\sum_{\mathbf{s}\in S}\left(\mathbf{s}[CA_1]\right)^2\pi(\mathbf{s})$ is needed in this metric but the ODEs cannot provide this information. That is to say, not all rewards associated with the CTMCs underlying PEPA models can be derived through the fluid approximation approach, only those that depend on the first-order moments of the CTMCs. We should point out that some work on the fluid approximation of higher-order moments of the component counting stochastic processes for a PEPA model has been presented in [HB08], but there is no theoretical justification. In [HB09], the application of fluid-generated higher moments to passage-time approximations has been demonstrated.

To conclude, if a performance metric is population-based, or can be represented by a linear reward function, then this measure can be derived through the fluid approximation approach. Otherwise, it is not possible to derive this metric using this approach. However, in this situation, we still have methods to get this measure from PEPA models, which will be presented later in this chapter. In the following, we first discuss how to use Little's law to derive average response time.

7.3.2 Deriving average response time via Little's Law

Response time is a very important performance metric which has already been discussed based on a small scale system in Chapter 2. For large scale systems, a good method to get average response time through fluid approximation models has been demonstrated in [CDGH08]. In that paper, expected passage response times for large PEPA models are calculated using Little's law.

Little's Law [Lit61] is usually phrased in terms of the jobs in a system and relates the average number of jobs in the system N to the residence time W, the average time they spend in the system. Let X be the throughput, i.e. the number of jobs completed during a unit time. Then for the system in its steady-state, Little's law states that:

$$N = XW$$

The residence time, of course, is the response time if we take the system to be that which occurs between a request and response action within a model. The only requirement for using Little's law is the existence of a stationary state.

For the content adaptation model, what we want to measure is the duration of the service flows, i.e. the time from the starting activity $pde\_int\_cont\_req$ to the ending activity csp_to_pde or ca_to_pde. Then using Little's Law we obtain:

$$W = \frac{PDE_2^* + PDE_3^*}{PDE_1^* \times r_{pde\_ext\_cont\_req}}, \tag{7.12}$$

where PDE_i^* represents the population of the local derivative PDE_i in the steady state $(i = 1, 2, 3)$. In (7.12), the sum of PDE_2^* and PDE_3^* represents the number of jobs engaged in the service in the system and $PDE_1^* \times r_{pde\_ext\_cont\_req}$ is the corresponding throughput. Since

in the steady state the rates of flux in and out are equal, so $PDE_1^* \times r_{pde\_ext\_cont\_req}$ in (7.12) could equally be replaced with $PDE_4^* \times r_{pde\_user\_interface}$, where PDE_4^* is the population of PDE_4 in the steady state.

To get the average response time W, the only remaining work is to determine the values of PDE_i^* $(i = 1, 2, 3)$. Clearly, PDE_i^* $(i = 1, 2, 3)$ are three entries of the vector of the equilibrium solution $\mathbf{x}^*$ of the ODEs derived from the same model. Therefore, the average response time is available as long as the solution is available. It should be pointed out that the steady solution $\mathbf{x}^*$ can be directly derived from the equilibrium equation:

$$F(\mathbf{x}^*) = \sum_{l \in \mathcal{A}_{\text{label}}} lf(\mathbf{x}^*, l) = 0. \tag{7.13}$$

In fact, since $\lim_{t\to\infty} \mathbf{x}(t) = \mathbf{x}^*$, then

$$0 = \lim_{t\to\infty} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \lim_{t\to\infty} F(\mathbf{x}) = F(\lim_{t\to\infty} \mathbf{x}) = F(\mathbf{x}^*).$$

Therefore, the average response time is "approximately" governed by the equilibrium equation. Thus in the sense of approximation, some capacity planning and optimisation for large scale content adaptation systems can be simply and directly carried out based on a set of nonlinear algebra equations.

7.3.3 Deriving performance measures through stochastic simulation approach

As discussed in Section 7.3.1, if a reward function associated with the CTMC underlying a PEPA model is not linear, then we cannot derive the corresponding reward through the fluid approximation approach. An alternative widely-used way to obtained performance is stochastic simulation. In fact, Gillespie's stochastic simulation algorithm (SSA) [Gil76] has already been implemented in the PEPA Eclipse plug-in [TDG09], a tool supporting PEPA.

The Gillespie algorithm has been widely applied to model and simulate biochemical reactions [Kie02]. This method exploits the fact that the duration from one transition (or reaction) to the next satisfies an exponential distribution with the reciprocal of total transition rate as the mean, and assumes that the transition rate is dependent on the state [UHCW06]. In the context of PEPA, this algorithm can be directly constructed based on our numerical represen-

tation scheme that is defined in Chapter 3, see Algorithm 3. In Algorithm 3, the states of a PEPA model are represented as numerical vector forms, and the rates between those states are specified by the transition rate functions which only depend on the transition type (i.e. labelled activity) and the current state. In this algorithm, the generated time τ in each iteration can be regarded as having been drawn from an exponential distribution with the mean $\frac{1}{f(\mathbf{x})}$, which has already been pointed out in [Gil76]. Therefore, Line 9 in Algorithm 3 is in fact expressing:

"generate τ from an exponential distribution with the mean $\frac{1}{f(\mathbf{x})}$".

Thus, this algorithm is essentially to simulate the CTMC underlying a PEPA model.

We have several reasons to put forward this algorithm in this thesis. Firstly, as one benefit of our numerical representation scheme, it provides a good platform for directly and conveniently exploiting the simulation algorithm for PEPA. Secondly, we emphasise that any general performance measure can be derived from PEPA models through this kind of simulation, particularly those that cannot be obtained by the fluid approximation approach. In addition, the simulated results can be used for comparisons with those obtained by solving ODEs. Finally, simulations offer a new insight into the dynamics of the CTMCs underlying PEPA models from the stochastic perspective, which is in contrast to the deterministic fluid approximation approach discussed intensively in the previous chapters.

In Algorithm 3, the stop condition could be: $t > T$, where T is a given large time. In addition, since the output performance converges as time goes to infinity as the following Proposition 7.3.2 states, other choices for stopping the algorithm include: the absolute error of two continued iterations is small enough:

$$\delta = \left| \frac{\mathrm{PerMeasure}_n}{t_n} - \frac{\mathrm{PerMeasure}_{n-1}}{t_{n-1}} \right| < \epsilon,$$

or the relative error is sufficient small:

$$\frac{\delta}{\mathrm{PerMeasure}_n / t_n} < \epsilon,$$

where ϵ is a given small number, t_{n-1}, t_n are the accumulated time up to the $n-1$-th and n-th iterations respectively, while $\mathrm{PerMeasure}_{n-1}$ and $\mathrm{PerMeasure}_n$ are the accumulated performance up to the $n-1$-th and n-th iterations respectively.

[2] In practise, in order to decrease the computational cost, we should not calculate the performance until after a warm up period so that the effects of the initial state bias can be considered to be negligible.

Algorithm 3 Gillespie simulation algorithm for deriving general performance measures from PEPA model

1: //Initialisation
2: starting state $\mathbf{x}$; labelled activity set $\mathcal{A}_{\text{label}} = \{l_1, l_2, \cdots, l_m\}$; activity matrix; transition rate function f
3: reward function ρ; $\text{PerMeasure} = 0$
4: **while** stop condition not satisfied **do**
5: //Sampling
6: compute the transition rate function $f(\mathbf{x}, l_j)$, $j = 1, 2, \cdots, m$
7: $f(\mathbf{x}) = \sum_{j=1}^{m} f(\mathbf{x}, l_j)$
8: generate uniform random numbers r_1, r_2 on $[0, 1]$
9: compute $\tau = -(1/f(\mathbf{x})) \ln r_1$
10: find μ such that $\sum_{j=1}^{\mu-1} f(\mathbf{x}, l_j) \leq r_2 f(\mathbf{x}) < \sum_{j=1}^{\mu} f(\mathbf{x}, l_j)$
11: //Updating
12: $\text{PerMeasure} = \text{PerMeasure} + \rho(\mathbf{x}) \times \tau$ // Accumulate performance measure[2]
13: $t = t + \tau$ //Accumulate time
14: $\mathbf{x} = \mathbf{x} + l_\mu$ // Update state vector of system
15: **end while**
16: Output performance: $\text{PerMeasure} = \frac{\text{PerMeasure}}{t}$

Now we prove that performance calculated using this algorithm converges. In order to prove this conclusion, we need the following theorem.

Theorem 7.3.1. *(Theorem 3.8.1, [Nor98]) If $X(t)$ is an irreducible and positive recurrent CTMC with the state space S and the unique invariant distribution π, then*

$$\Pr\left(\frac{1}{t}\int_0^t 1_{\{X_z=\mathbf{s}\}}\mathrm{d}z \to \pi(\mathbf{s}) \text{ as } t \to \infty\right) = 1. \tag{7.14}$$

Moreover, for any bounded function $\rho : S \to \mathbb{R}$, we have

$$\Pr\left(\frac{1}{t}\int_0^t \rho(X_z)\mathrm{d}z \to E[\rho(X)] \text{ as } t \to \infty\right) = 1. \tag{7.15}$$

where $E[\rho(X)] = \sum_{\mathbf{s}\in S} \rho(\mathbf{s})\pi(\mathbf{s})$.

Here is our conclusion:

Proposition 7.3.2. *The performance measure calculated according to Algorithm 3 converges as time goes to infinity.*

Proof. Assume that $n-1$ iterations have been finished and the time has accumulated to t_{n-1}. Suppose the current one is the n-th iteration and τ is the generated time in this iteration. After the n-th iteration is finished, the accumulated time will be updated to $t_n = t_{n-1} + \tau$. During the τ time interval, the simulated CTMC stays in the state $\mathbf{x}$, that is, $X_z = \mathbf{x}, z \in [t_{n-1}, t_n)$. Therefore,

$$\rho(\mathbf{x}) \times \tau = \rho(\mathbf{x}) \int_{t_{n-1}}^{t_n} \mathrm{d}z = \int_{t_{n-1}}^{t_n} \rho(\mathbf{x}) \mathrm{d}z = \int_{t_{n-1}}^{t_n} \rho(X_z) \mathrm{d}z.$$

That is, the performance calculated in this n-th iteration is $\displaystyle\int_{t_{n-1}}^{t_n} \rho(X_z)\mathrm{d}z$. Thus, the performance "$\mathrm{PerMeasure}$" accumulated in the previous $n-1$ iterations is

$$\int_{t_0}^{t_1} \rho(X_z)\mathrm{d}z + \int_{t_1}^{t_2} \rho(X_z)\mathrm{d}z + \cdots + \int_{t_{n-2}}^{t_{n-1}} \rho(X_z)\mathrm{d}z = \int_{0}^{t_{n-1}} \rho(X_z)\mathrm{d}z.$$

After updating in this n-th iteration, the performance $\mathrm{PerMeasure}$ will be accumulated to

$$\int_{0}^{t_{n-1}} \rho(X_z)\mathrm{d}z + \int_{t_{n-1}}^{t_n} \rho(X_z)\mathrm{d}z = \int_{0}^{t_n} \rho(X_z)\mathrm{d}z.$$

Therefore,

$$\frac{\mathrm{PerMeasure}}{t_n} = \frac{1}{t_n} \int_0^{t_n} \rho(X_z)\mathrm{d}z.$$

According to Theorem 7.3.1, $\displaystyle\frac{1}{t_n}\int_0^{t_n} \rho(X_z)\mathrm{d}z \to E[\rho(X)]$ as t_n tends to infinity. So the performance obtained through Algorithm 3 converges as the simulation time goes to infinity. □

Performance measure	Reward function	Alternative way
averaged population $\mathbf{s}^*$	$\rho(\mathbf{s}) = \mathbf{s}$	
total variation of $\mathbf{s}^*$	$\rho(\mathbf{s}) = (\mathbf{s} - \mathbf{x}^*)^T(\mathbf{s} - \mathbf{s}^*)$	
utilisation of CA_4	$\rho(\mathbf{s}) = \frac{\mathbf{s}[CA_4]}{N_{CA_0}}$	$\frac{\mathbf{s}^*[CA_4]}{N_{CA_0}}$
throughput of $ca\_adaptation$	$\rho(\mathbf{s}) = \mathbf{s}[CA_4]r_{ca\_adaptation}$	$\mathbf{s}^*[CA_4]r_{ca\_adaptation}$
response time		$\frac{\mathbf{s}^*(PDE_2) + \mathbf{s}^*(PDE_3)}{\mathbf{s}^*(PDE_1) \times r_{pde\_ext\_cont\_req}}$

Table 7.2: *Deriving performance measures through stochastic simulation*

Performance metrics, such as throughput of an activity and utilisation of a local derivative, can be derived through this algorithm by choosing appropriate reward functions. For example, for the content adaptation model, some computational methods for common performance measures have been given in Table 7.2. Particularly in this table, the metric of the total variation of the

averaged population cannot be obtained through the fluid approximation approach. When using Algorithm 3 to derive the throughput of a labelled activity α, we have another choice: the line 12 and line 16 of this algorithm can be respectively replaced by

$$\mathrm{Thr}(\alpha) = \mathrm{Thr}(\alpha) + 1_{\{l_\mu = \alpha\}},$$

$$\mathrm{Thr}(\alpha) = \frac{\mathrm{Thr}(\alpha)}{t}.$$

These two new lines account the number of occurrences of α in the total t time. This method is equivalent to the way of calculating throughput which is presented in Table 7.2.

7.3.4 Comparison of performance measures through different approaches

Until now we have discussed three approaches to deriving performance measures from a PEPA model. The first way that has been presented in Chapter 2 is to utilise the steady-state probability distribution of the CTMC, which can be obtained by solving a matrix equation associated with the corresponding infinitesimal generator. As we have pointed out, it is not feasible to get the steady-state distribution when the state space become very large. The other two approaches are the fluid approximation and stochastic simulation based on our numerical representation scheme, which have been discussed in the previous subsections. In this subsection, we will present a comparison between these approaches, in terms of both accuracy and computational cost, to derive the average response time for the content adaptation system.

7.3.4.1 Comparison in small scale case

According to Little's law and as the formula (7.12) shows, the average response time can be calculated based on the averaged populations of PDE_1, PDE_2, PDE_3. We have three different approaches to obtain these populations. The first way, named "solving generator" for convenience, is to get the expected populations based on the steady-state probability distribution, which is obtained by solving the global balance equation relating to the infinitesimal generator. This process can be automatically carried out by the PEPA Eclipse plug-in.

The second method is to numerically solve the corresponding ODEs (using the Euler difference algorithm) to derive the equilibrium, from which the populations of PDE_i $(i = 1, 2, 3)$ are available. The last approach is to employ Algorithm 3 to simulate the CTMC and get the

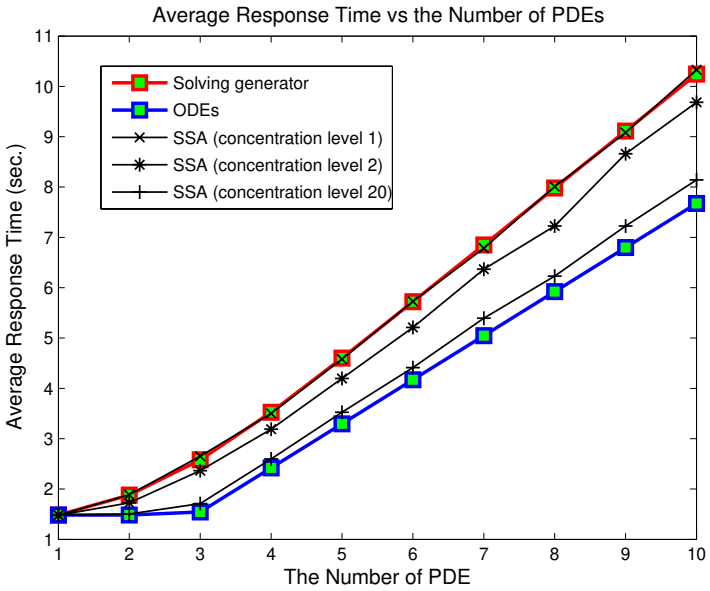
averaged populations. We should point out that Algorithm 3 is designed for simulating the underlying CTMC in the usual sense, which is also the first CTMC in the density-dependent family CTMCs underlying the same model, i.e. the CTMC $\hat{X}_1(t)$ with the concentration level one. When the algorithm is utilised to simulate the CTMC $\hat{X}_n(t)$ of concentration level n, according to the definition of density dependent in Chapter 5, it should be modified as follows: in the algorithm each transition vector l_j must be replaced by l_j/n and each transition rate function $f(\mathbf{x}, l_j)$ be replaced by $f(\mathbf{x}, l_j/n)$.

Suppose there is one AM, one CA and one C/S Provider in the content adaptation system, i.e. $(N, P, Q) = (1, 1, 1)$. The number of PDEs varies from one to ten. The experimental results obtained through the three approaches are presented in Figure 7.5 (a). Of course, the approach of "solving generator" provides the exact performance of the CTMC, i.e. the CTMC with the concentration level one, see the red curve. The simulated results with the concentration level one is coincident with the red curve, reflecting the consistence between the approach of "solving generator" and the approach of simulation to derive the performance from the same CTMC. As the level increases, the simulated performance approximates the blue curve which represents the results obtained by the fluid approximation approach. This is consistent with the conclusion that the ODEs are the limit of a family of density-dependent CTMCs as the concentration level tends to infinity. As Figure 7.5 (a) shows, there is a gap between the red and blue curves, indicating that for a small scale system the performance derived using fluid approximation may differ from the exact performance of the system. This difference results from the fact that the fluid approximation only captures the information about the CTMC with concentration level infinity rather than with level one. However, as we will see later, as the system scale increases, this difference will decrease to an acceptable level.

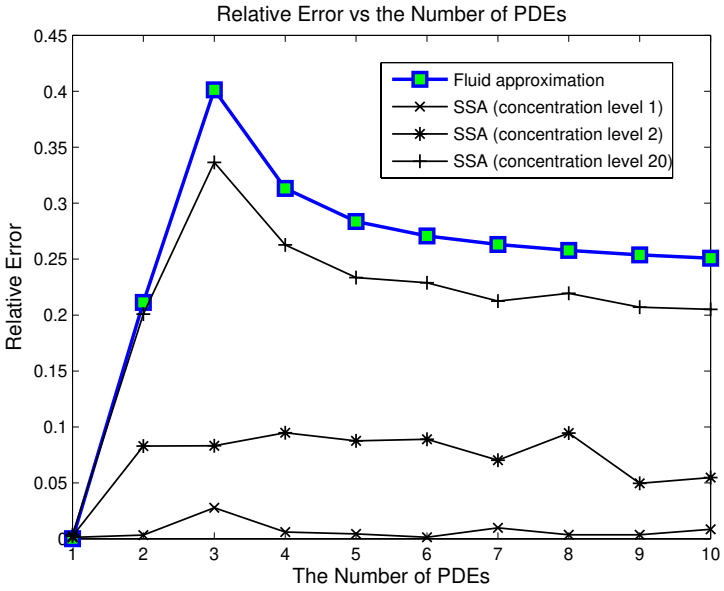
We would like to use *relative errors* to measure this kind of difference. The relative error of a result obtained by a method is defined as the relative difference between this result and the result derived by the simulation with the concentration level one, that is

$$\text{relative error} = \frac{|\text{ result by the other method} - \text{result by SSA with concentration level one})\,|}{\text{result by SSA with concentration level one}}.$$

In the case of small scale modelling, the reference for calculating relative error should be replaced by the numerical solution of the CTMC, as long as it is available. The information of the relative error about the results presented in Figure 7.5 (a) is given in Figure 7.5 (b). The relative errors between the exact performance and those obtained through the fluid approximation way



(a)



(b) Relative error

Figure 7.5: *Comparison between three approaches to derive response time*

are generally greater than 25%, while they reach a peak at the critical point three, which is also a turning point of the blue curve. As the concentration level decreases, the relative errors of the simulated results decrease. As we can see, when the concentration level decreases from twenty to two, the relative errors decrease from more than 20% to less than 10%. The relative errors are almost zero when the concentration level is one.

The computational cost of these experiments, mainly in terms of running times, is presented in Table 7.3. These experiments, except for those obtained through the "solving generator" approach, are carried out using Matlab 7.8.0 (R2009a) on a 2.66GHz Xeon CPU with 4Gb RAM runing Scientific Linux 5. As seen from Figure 7.1-7.4 in Section 7.2.2, the solutions of the ODEs will achieve the equilibria before time 20, so the stop time for the fluid approximation is set as $T = 20$. The stop time for the simulation using Algorithm 3 is set as $T = 5000$, since it can finally make the relative errors of the accumulated performance in two consecutive iterations less than 5%.

(M, N, P, Q)	Solving generator	ODEs: $T = 20$ step length 10^{-5}	SSA: $T = 5000$ $n = 1$	SSA $T = 5000$ $n = 2$	SSA $T = 5000$ $n = 20$
$(i, 1, 1, 1)$ $i = 1, \cdots, 10$	< 1.3	1.5885	3.0512	10.2321	378.4077

Table 7.3: *Running times (sec.) of small scale experiments*

There is no doubt that for small scale modelling in which the steady probability distribution can be obtained, the approach "solving generator" is promising, because it gives exact performance values at a low running time. The simulation approach can also provide a similar accuracy, but the computational cost is higher. The cost of the approach of fluid approximation is low but the accuracy is also low.

7.3.4.2 Comparison in large scale case

When there are 20 AMs, 20 CAs, 20 C/S Providers in the system, calculating the steady-state probability distribution by the PEPA Eclipse plug-in becomes infeasible. We have to rely on the approaches of fluid approximation and stochastic simulation to get performance measures. As we can see from Figure 7.6 (a) and (b), the fluid approximation can achieve good accuracy. The relative errors of the results obtained using the fluid approximation are about 6%. Here the reference for calculating relative errors is the simulated results with the concentration level one,

since they can represent the exact performance of the system considered. These two graphs suggest that the fluid approximation approach is reliable and accurate to derive performance measures from large scale systems.

As the concentration level increases, the percentage error in terms of population will decrease, which has been demonstrated in Section 7.2.2. That is, the path of the CTMC will approximate the solution of the ODEs more closely. Notice that the average response time is calculated based on averaged populations (or the first-order moment) according to Little's law. Therefore, as the concentration level increases, the response time will deviate from that corresponding to concentration level one and tends to those corresponding to concentration level infinity, i.e. the ODEs. Thus, the accuracy of the response time will correspondingly decrease, i.e. the relative errors will increase. That is why in Figure 7.6 (b), (d) and (f), the simulated results with concentration two have smaller relative errors compared to the results derived through the fluid approximation approach.

Moreover, as also indicated in Section 7.2.2, the percentage errors of averaged populations should be much less than those of one-realisation of populations. That is why in the same scenario, for example, $(M, N, P, Q) = (300, 20, 20, 20)$ and the concentration level $n = 2$, the relative error (about 3%) of the performance which is based on the average population is much less than the percentage error in terms of a one-realisation path (20%).

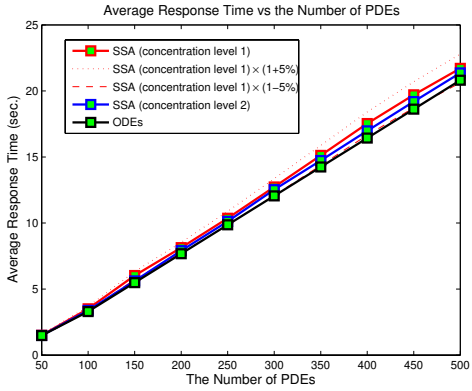
	path (percentage error)	performance (relative error)
$n \nearrow$	$\searrow$	$\nearrow$
$\max\{M, N, P, Q\} \nearrow$	weak $\searrow$	little effect
$\min\{M, N, P, Q\} \nearrow$	$\searrow$	$\searrow$

Table 7.4: *Factors and effects on paths and performance*

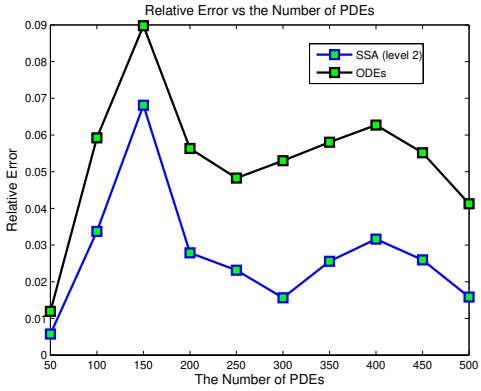
(M, N, P, Q)	ODEs: $T = 20$ step length 10^{-5}	SSA: $T = 5000$ $n = 1$	SSA $T = 5000$ $n = 2$
$(i, 20, 20, 20),\ i = 50 : 50 : 500$	1.6451 sec	5 min $\sim$ 6 min	12 min $\sim$ 13 min
$(i, 30, 30, 30),\ i = 50 : 50 : 500$	1.8148 sec	8 min $\sim$ 9 min	19 min $\sim$ 20 min
$(i, 50, 50, 50),\ i = 50 : 50 : 500$	1.8448 sec	14 min $\sim$ 15 min	31 min $\sim$ 32 min

Table 7.5: *Running time of large scale experiments*

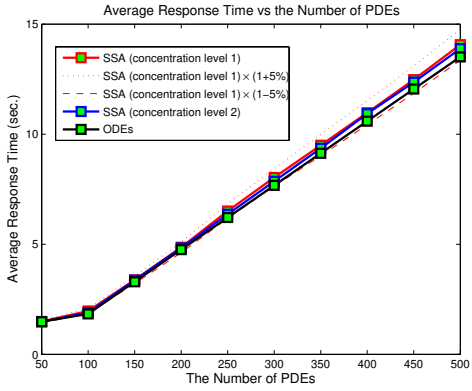
As Figure 7.6 shows, the smallest population of entities has more effect than the maximum of the populations on the improvement of performance. In fact, when the smallest population



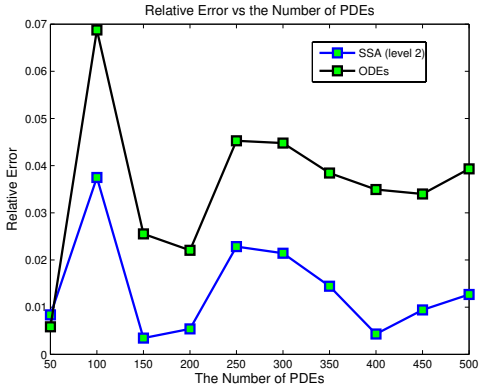
(a) $(N, P, Q) = (20, 20, 20)$



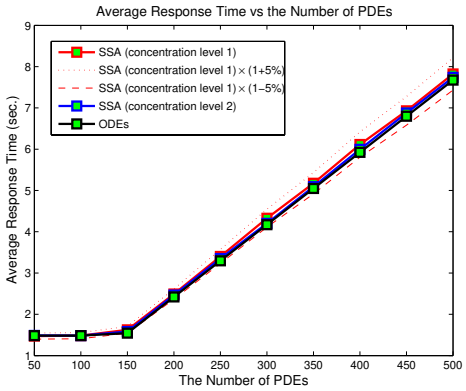
(b) $(N, P, Q) = (20, 20, 20)$



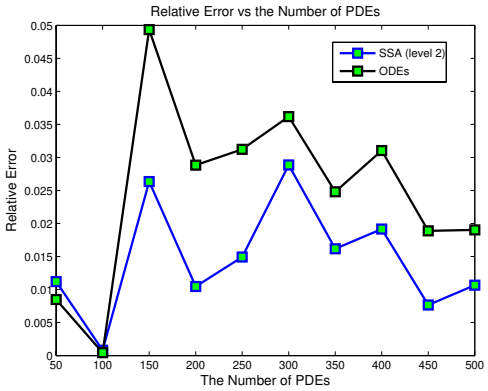
(c) $(N, P, Q) = (30, 30, 30)$



(d) $(N, P, Q) = (30, 30, 30)$



(e) $(N, P, Q) = (50, 50, 50)$



(f) $(N, P, Q) = (50, 50, 50)$

Figure 7.6: *Comparison between fluid approximation and stochastic simulation:*

increases from 20 to 30 and further to 50, the relative error of the fluid approximation will correspondingly decrease from around 6% to around 4% and further to 3%. However, in these graphs, when the number of PDEs increases from 50 to 500 while the numbers of other entities are fixed, there is no evident decrease in the relative errors. See Figure 7.6 (b), (d) and (f). These experimental results are summurised in Table 7.4. In addition, another interesting phenomenon is that in each of Figure 7.6 (b), (d) and (f), there is an obvious peak of relative errors of the simulated results with the concentration level one.

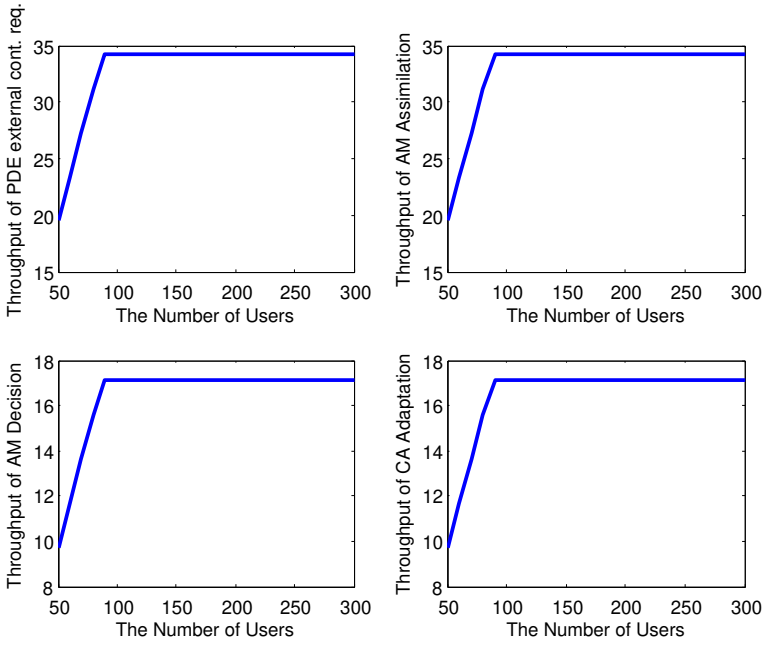
The running times of these experiments has been reported in Table 7.5. For the fluid approximation approach, the computational cost appears to be independent of the populations of the entities, while that of the simulation approach strongly depends on these populations. Therefore, for large scale performance modelling, the fluid approximation approach is a promising way to achieve a high accuracy at a low computational cost.

7.4 Performance Analysis for Large Scale Content Adaptation Systems

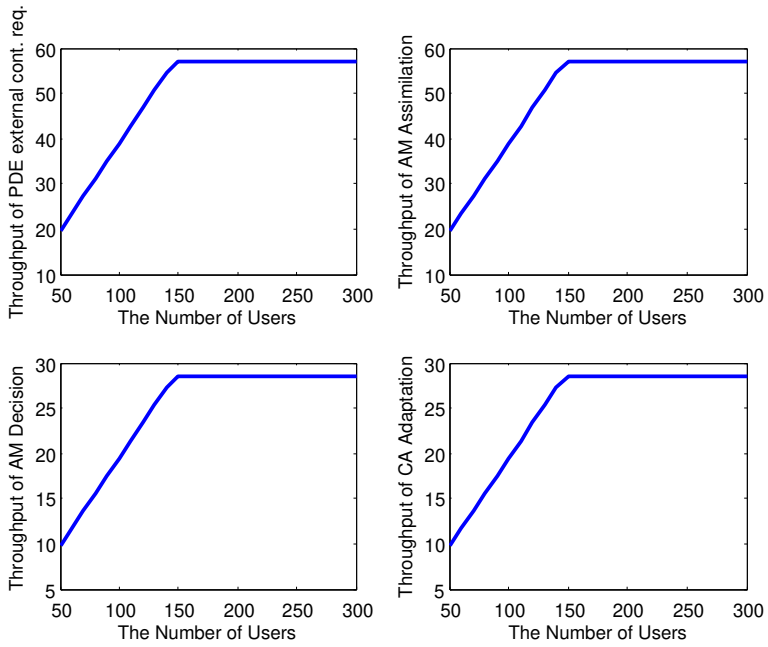
As we know, in the sense of approximation, the averaged performance measures such as response time are controlled by the equilibrium equation:

$$F(\mathbf{x}^*) = \sum_{l \in \mathcal{A}_{\text{label}}} l f(\mathbf{x}^*, l) = 0.$$

Generally, there are two types of factors affecting this equation and thus affecting the equilibrium $\mathbf{x}^*$. The first type are the coefficients of the equation, which are determined by the activity rates and reflect the operation speed of individual entities. The second one is the populations of entities, which are given in the initial condition, i.e. the starting state of the system, indicating the loading and resource conditions of the system. In this section, we will show how these two kinds of factors impact the response time of the system. This analysis is based on the fluid approximation method, i.e. all results shown in this section are obtained through the numerical solutions of the derived ODEs in Section 7.2.1.



(a) $(N, P, Q) = (30, 30, 30)$



(b) $(N, P, Q) = (50, 50, 50)$

Figure 7.7: *Throughput vs the number of PDEs*

7.4.1 Scalability analysis

One of the most severe problems that the content adaptation management system has to deal with, is the scalability issue: an adaptation service may have acceptable performance at present, but how is this performance likely to change as greater numbers of users are added.

Since the limited system resources have to be shared by the users, it is not surprising to see that the more users are in the system requiring service, the longer the waiting time of each user will be. As shown in Figure 7.6, the average response time increases almost linearly as the number of users increases. Notice in this figure, when many resources are available, say, 50 AMs, 50 CAs and 50 C/S Providers contained in the system, the average response time remains unchanged when the number of users varies up to 150. In other words, in this scenario there is some capacity left for extra users. After this critical point of 150 users, the waiting time increases linearly with the number of users, which indicates that the resources are approaching being fully utilised. The slopes of the "plots" in Figure 7.6 vary from case to case, reflecting the performance difference resulting from the different capacity planning decisions represented by the scenarios.

Let us consider this problem from the perspective of throughput. We know that the throughput of the CA's "Adaptation" reflects how fast the system runs the adaptation. Similarly, the throughput of the AM's "Assimilation" and "Decision" reveal the speed of the system doing the jobs of assimilation and decision-making respectively. The throughput of the PDE's "external content request" indicates how many external requests can be dealt with by the system during one second.

The variations of these four activities' throughput are shown in Figure 7.7(a) and Figure 7.7(b). In both figures, all the throughput increases with the number of users until they reach their respective peaks. This phenomenon is due to the fact that more users generate more content requests, making the system busier and leading to the performance improvement until the system fulfils its potential. After the respective critical points in terms of the number of users, all of the curves of throughput remain flat, reflecting no improvement of performance after the system resources are fully utilised. Corresponding to Figure 7.6, the critical point of users in Figure 7.7(b) is also 150. So the two different performance measures, i.e. the response time and throughput, are consistent.

In both Figure 7.7(a) and Figure 7.7(b), the throughput of the PDE's external content request

and the AM's assimilation is two times that of the throughput of the AM's decision and the CA's adaptation. This is due to decision making and content adaptation being not always required since not all requested content needs to be adapted. In fact, the activities of external content request and assimilation appear in both service flow 1 and 2 in Section 2.4.3 of Chapter 2, but there is no decision or adaptation in service flow 2.

7.4.2 Capacity planning

The previous subsection discussed the variation of performance under conditions of increasing loading while the resource conditions are fixed. However, for a fixed number of users, different resource conditions may have different impacts on the system performance. As shown in Figure 7.8, adding more C/S Providers into the system, which makes more resource available for 300 users, can decrease the response time. For example, an improvement of more than 6.5s for an average user's waiting time can be achieved if 30 providers are added into the system which already has 50 AMs, 50 CAs and 20 C/S Providers (see the corresponding curve marked by squares in the figure).

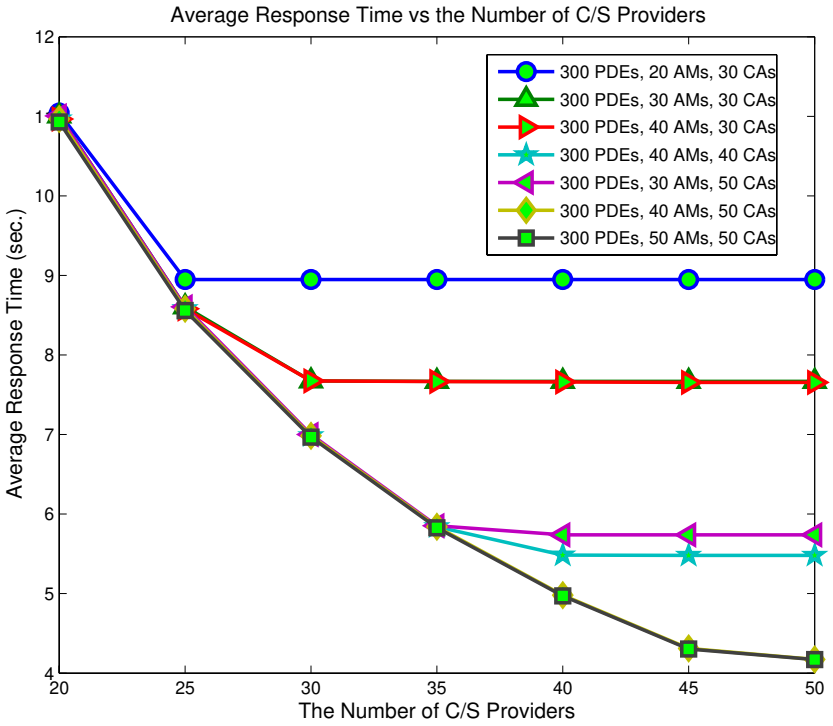


Figure 7.8: *Impact of the number of C/S Providers on performance*

If a system has 30 CAs, whenever the number of the C/S Providers varies there is not much difference in performance between the case of 30 AMs and the one of 40 AMs. This suggests no substantial improvement in performance can be gained by adding more AMs into the system in the scenario of 30 AMs and 30 CAs. But a significant reduction of about 2s will occur between the case of 30 AMs, 30 CAs and the case of 20 AMs, 30 CAs. Similarly, if the system has 50 CAs, the curves associated with 40 AMs and 50 AMs are very close all the time, while finally there is a gap of more than 1.5s between the curves associated with 40 AMs and 30 AMs.

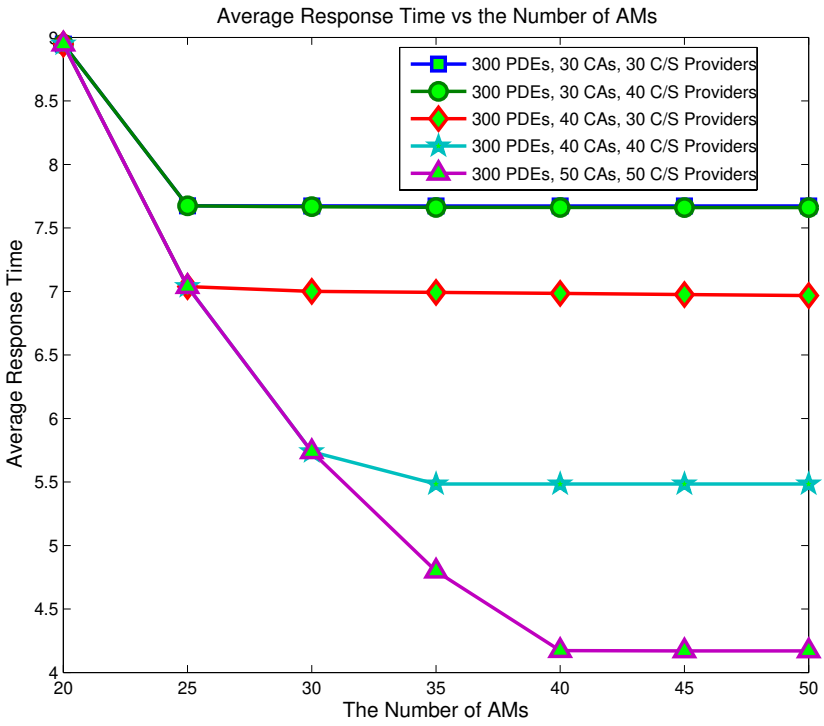


Figure 7.9: *Impact of the number of AMs on performance*

If there are 40 AMs in the system, as the number of C/S Providers increases, there will be significant gaps between the cases of 30, 40 and 50 CAs. The smaller the number of CAs is, the larger the gap seems to be. Figure 7.8 also illustrates that all curves tend to be flat as the number of C/S Providers goes up. In other words, as more and more providers are added into the system, the lack of the AM and the CA for matching the increase of the providers will become the bottleneck, preventing further reduction of the response time. For example, if the system has only 20 AMs and 25 CAs, 25 C/S Providers are sufficient, since there is no more improvement of performance when more providers are added.

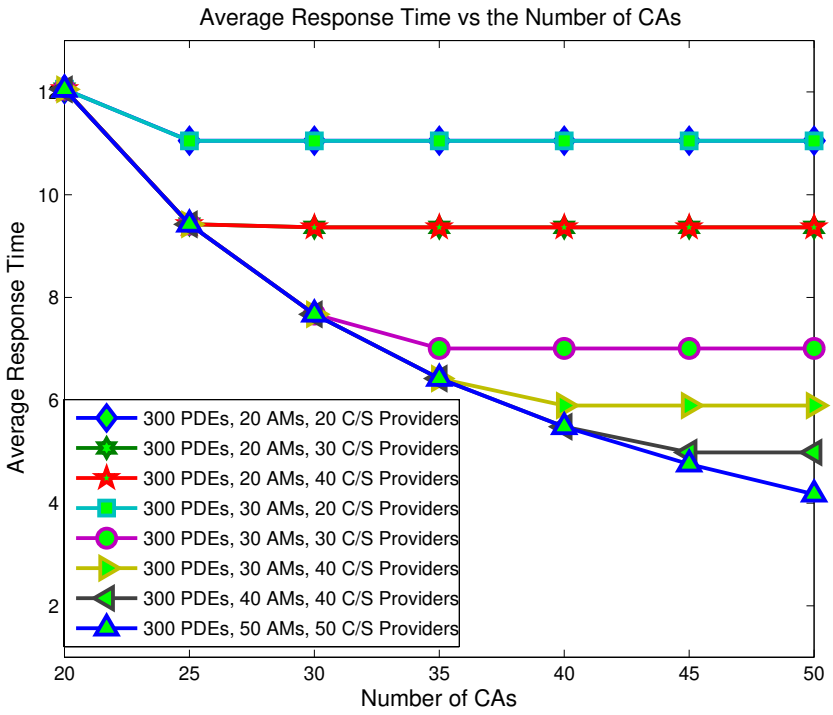


Figure 7.10: *Impact of the number of CAs on performance*

The system performance, of course, could be influenced by the number of the AMs and CAs, see Figure 7.9 and Figure 7.10 respectively. However, as Figure 7.10 demonstrates, the number of the CAs has little impact in the case of 20 C/S Providers, which is due to the bottleneck caused by the C/S Provider. Nevertheless, when the number of the AMs is fixed as 30, the gap between the case of 20 providers and the case of 40 providers will expand from less than 0.5s to more than 3s as the number of CAs increases from 20 to 50.

7.4.3 Sensitivity analysis

Our experiments show that the speed of the process of context assimilation has a significant role in improving users' satisfaction. In fact, as Figure 7.11 shows, increasing the rate of the AM's context assimilation from 0.5 to 5.5, i.e. correspondingly decreasing the average duration of this activity from about 2s to 0.182s, can dramatically reduce the average response time from 35s to 12s (see the curve marked by diamonds in Figure 7.11). This suggests that the context assimilation, as an indispensable element in the management process, should be suitably quick and efficient.

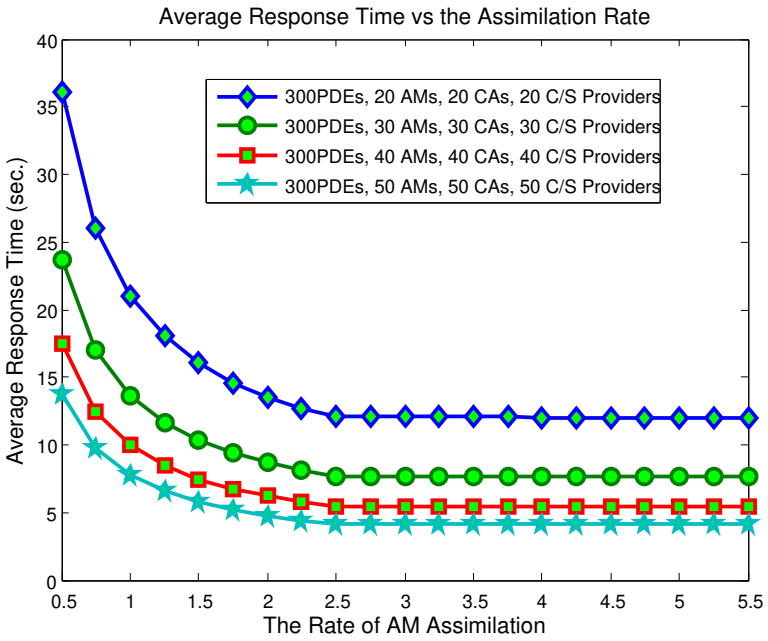


Figure 7.11: *Impact of assimilation rate on performance*

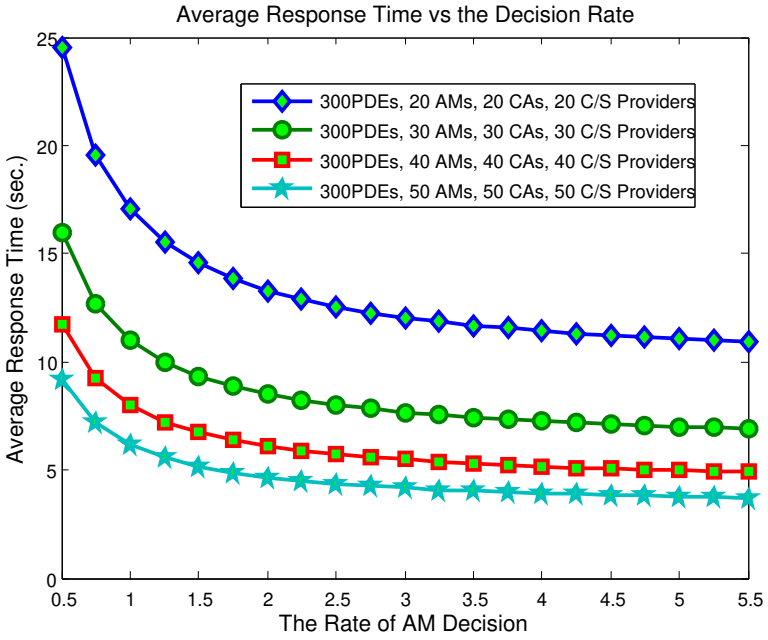


Figure 7.12: *Impact of decision rate on performance*

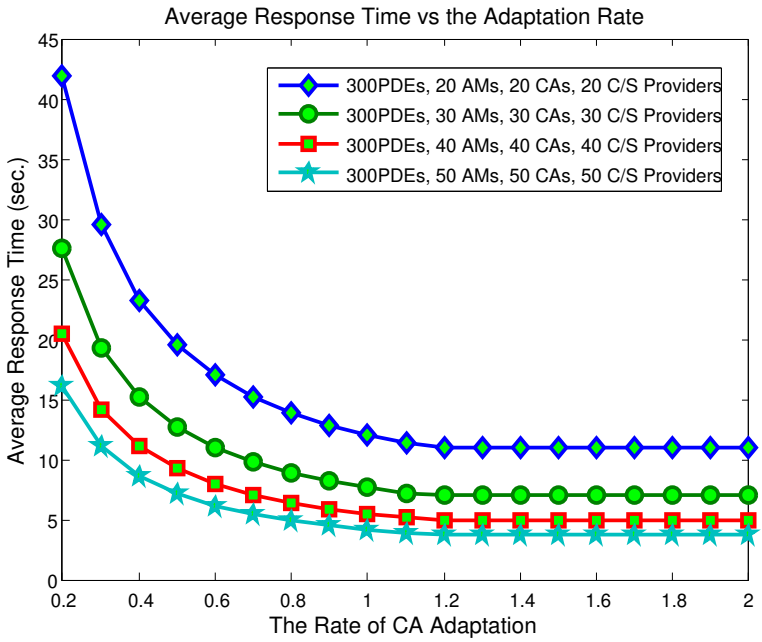


Figure 7.13: *Impact of adaptation rate on performance*

Decision making, aimed at creating an adaptation plan when needed, is another important activity of the AM, whose rate also impacts the response time, as shown in Figure 7.12. However, because the decision making is not always required since not every requested content needs to be adapted, the impact of the decision rate, shown in Figure 7.12 is less significant than the impact of the context assimilation rate shown in Figure 7.11. In both figures all curves are nearly flat when the relevant rate is above three, indicating that increasing the speed of assimilation or decision-making above this value does not help to improve the system performance.

Increasing the rate of CA's adaptation can also reduce the user's waiting time. For example, as the adaptation rate increases from 0.2 to 1.2, or correspondingly the adaptation duration decreases from 5s to about 0.83s, the response time reduces from about 42s to 11s (see the curve marked by diamonds in Figure 7.13).

As we can see from the above figures, the slight improvement of the components' performance can result in a significant change in the system's performance. This is because the improvement of an individual component's performance can not only decrease the processing time for a particular user, but can also decrease the number and the waiting time of the other users' requests in the buffer.

7.5 Structural Analysis of A Subsystem

In reality, content adaptation management may be rather complex, particularly in a highly loaded condition, due to the heterogeneity of devices and the diversity of requirements. The previous section has presented two ways to enhance the system performance: in addition to improving the performance of individual entities, the system manager could have an alternative way—capacity planning—to meet the users' requirements. In both ways, it is important to know information about the system such as the working states of those entities, for the purpose of efficiently controlling and optimising the management of the whole. Therefore, in this section we will deduce some functional properties of content adaptation systems by the techniques developed for PEPA that were presented in Chapter 4.

7.5.1 Adaptation management model

For convenience, the illustration of our technique is based on a subsystem of the content adaptation framework, which only consists of the three entities: the AM, the CA and the C/S Provider. This subsystem can be considered as an external environment to the users, which only conducts adaptation processes without communication with the users. See Figure 7.14, which illustrates the working cycle of the subsystem. We should point out that the users as well as the interactions with the users are not considered in this working mechanism. This allows us to place more emphasis on the management process and pay more attention to the interactions between the entities which can carry out the management and provide services. Another reason is that, a realistic system may differ significantly from the system considered in this chapter. A useful technique which can generally apply is thus more important than a result concluded from a specific model. Therefore, we emphasise our technique and just use the subsystem to illustrate the technique in this section.

The PEPA model of the operation of this subsystem can be similarly defined. See below.

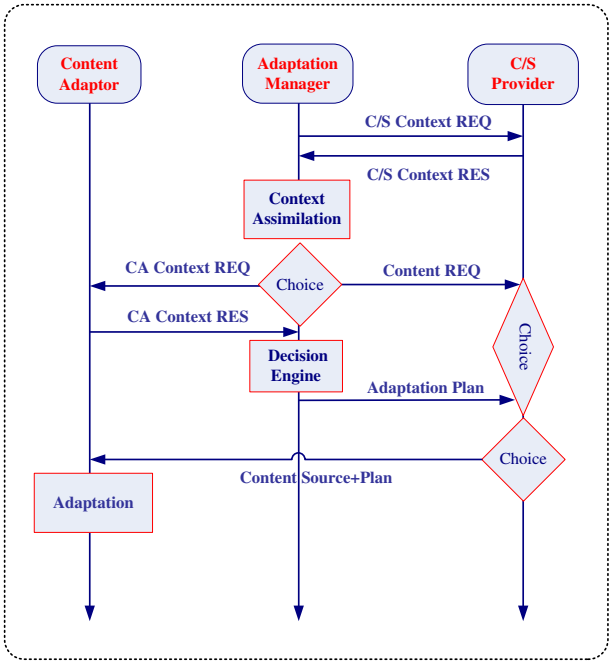


Figure 7.14: *Working cycle of content adaption model*

AM:

$$
\begin{aligned}
AM_2 &\stackrel{def}{=} (csp\_cc\_req, r_{csp\_cc\_req}).AM_3 \\
AM_3 &\stackrel{def}{=} (csp\_cc\_res, \top).AM_4 \\
AM_4 &\stackrel{def}{=} (am\_assimilation, \tfrac{1}{2} r_{am\_assimilation}).AM_5 \\
&\quad + (am\_assimilation, \tfrac{1}{2} r_{am\_assimilation}).AM_9 \\
AM_5 &\stackrel{def}{=} (ca\_states\_req, r_{ca\_states\_req}).AM_6 \\
AM_6 &\stackrel{def}{=} (ca\_states\_res, \top).AM_7 \\
AM_7 &\stackrel{def}{=} (am\_decision, r_{am\_decision}).AM_8 \\
AM_8 &\stackrel{def}{=} (am\_adapt\_plan, r_{am\_adapt\_plan}).AM_2 \\
AM_9 &\stackrel{def}{=} (am\_cont\_req, r_{am\_cont\_req}).AM_2
\end{aligned}
$$

CA:

$$
\begin{aligned}
CA_1 &\stackrel{def}{=} (ca\_states\_req, \top).CA_2 \\
CA_2 &\stackrel{def}{=} (ca\_states\_res, r_{ca\_states\_res}).CA_3 \\
CA_3 &\stackrel{def}{=} (csp\_call\_ca\_adapt, \top).CA_4 \\
CA_4 &\stackrel{def}{=} (ca\_adaptation, r_{ca\_adaptation}).CA_1
\end{aligned}
$$

C/S Provider:

$$\begin{aligned}
CSP_1 &\stackrel{def}{=} (csp_cc_req, \top).CSP_2 \\
CSP_2 &\stackrel{def}{=} (csp_cc_res, r_{csp_cc_res}).CSP_3 \\
CSP_3 &\stackrel{def}{=} (am_cont_req, \top).CSP_4 \\
&+ (am_adapt_plan, \top).CSP_5 \\
CSP_4 &\stackrel{def}{=} (csp_to_pde, r_{csp_to_pde}).CSP_1 \\
CSP_5 &\stackrel{def}{=} (csp_call_ca_adapt, r_{csp_call_ca_adapt}).CSP_1
\end{aligned}$$

The system equation of the model is

$$\Big(AM_2[N] \underset{H_1}{\bowtie} CA_1[P]\Big) \underset{H_2}{\bowtie} CSP_1[Q],$$

where

$$H_1 = \{ca\_states\_req, ca\_states\_res\},$$

$$H_2 = \{csp\_cc\_req, csp\_cc\_res, am\_cont\_req, am\_adapt\_plan, csp\_call\_ca\_adapt\}.$$

7.5.2 Invariants

By Algorithm 1 demonstrated in Chapter 3, the activity matrix $\mathbf{C}$ of the model of the subsystem is derived and presented in Table 7.6. In this table, $am\_assimilation1$ represents the labelled activity $am\_assimilation^{AM_4 \to AM_9}$, while $am\_assimilation2$ represents the labelled activity $am\_assimilation^{AM_4 \to AM_5}$. Since there is no confusion, the labels of other activities are omitted for convenience. All structural information of the system is captured in this matrix.

According to Lemma 4.3.1 in Chapter 4, any solution $\mathbf{y}$ of $\mathbf{C}^T\mathbf{y} = 0$ is an invariant. Here the activity matrix $\mathbf{C}$ is a 17×12 matrix with rank ten. Therefore, according to the theory of linear algebra, the rank of the following solution space

$$\{\mathbf{y} : \mathbf{C}^T\mathbf{y} = 0\}$$

is $17 - 10 = 7$. That is, there are seven invariants which can form the bases of the solution space. These are presented in Table 7.7. In this table, each row represents an invariant. For example, $\mathbf{y}_4$ in Table 7.7 demonstrates that $\mathbf{x}(CSP_2) - \mathbf{x}(AM_3)$ is a constant, that is, the difference between the number of CSP_2 and the number of AM_3 remains unchanged at any

time. One benefit brought by such an invariant is that some information about other entities can be derived and collected through the invariance of one particular entity. For example, if we know the situation in terms of the AM, then by the invariant $\mathbf{y}_4$, the number of C/S Providers in the local derivative CSP_2 can also be known. Similarly, $\mathbf{x}(CA_2) - \mathbf{x}(AM_6)$ is another constant which is illustrated by $\mathbf{y}_6$. These invariants can help the manager to improve the efficiency and performance of the AM as well as the whole system.

In addition, the first three rows of this table, i.e. the invariants $\mathbf{y}_1, \mathbf{y}_2$ and $\mathbf{y}_3$, reflect the conservation law satisfied by the AM, CA and C/S Provider respectively.

l	AM_2	AM_3	AM_4	AM_5	AM_6	AM_7	AM_8	AM_9	CA_1	CA_2	CA_3	CA_4	CSP_1	CSP_2	CSP_3	CSP_4	CSP_5
$am\_assimilation1$	0	0	−1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
$am\_assimilation2$	0	0	−1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
am_cont_req	1	0	0	0	0	0	0	−1	0	0	0	0	0	0	−1	1	0
$am\_decision$	0	0	0	0	0	−1	1	0	0	0	0	0	0	0	0	0	0
am_adapt_plan	1	0	0	0	0	0	−1	0	0	0	0	0	0	0	−1	0	1
ca_states_req	0	0	0	−1	0	0	0	0	−1	1	0	0	0	0	0	0	0
ca_states_res	0	0	0	0	−1	1	0	0	0	−1	1	0	0	0	0	0	0
$ca\_adaptation$	0	0	0	0	0	0	0	0	1	0	−1	0	0	0	0	0	0
csp_cc_req	−1	1	0	0	0	0	0	0	0	0	0	0	−1	1	0	0	0
csp_cc_res	0	−1	1	0	0	0	0	0	0	0	0	0	0	−1	1	0	0
$csp\_call\_ca\_adapt$	0	0	0	0	0	0	0	0	0	0	−1	1	1	0	0	0	−1
csp_to_pde	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	−1	0

Table 7.6: *Activity matrix* $\mathbf{C}$ *of the sub content management model*

	AM_2	AM_3	AM_4	AM_5	AM_6	AM_7	AM_8	AM_9	CA_1	CA_2	CA_3	CA_4	CSP_1	CSP_2	CSP_3	CSP_4	CSP_5
$\mathbf{y}_1$	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
$\mathbf{y}_2$	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
$\mathbf{y}_3$	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
$\mathbf{y}_4$	0	−1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
$\mathbf{y}_5$	0	0	0	0	0	1	1	0	0	0	−1	0	0	0	0	0	1
$\mathbf{y}_6$	0	0	0	0	−1	0	0	0	0	1	0	0	0	0	0	0	0
$\mathbf{y}_7$	0	0	1	1	0	0	0	1	0	1	1	0	0	0	−1	0	−1

Table 7.7: *Invariants of the sub content management model*

7.5.3 Deadlock-checking

One important issue in protocol design for content adaptation as well as other systems is to avoid deadlocks, which can make a system become stuck in a particular state. In Chapter 4, we have provided a structure-based algorithm, i.e. Algorithm 2, to check deadlocks for large scale systems.

We should point out that the model considered here is not equal conflict. In fact, we know that from the activity matrix, $\mathrm{pre}(am\_cont\_req) \cap \mathrm{pre}(am\_adapt\_plan) = \{CSP_3\}$ but they are not equal. So, according to Proposition 4.4.1 in Chapter 4, the model is not equal conflict. However, as discussed in Section 4.5.3, we can still apply this algorithm to the content adaptation system. That is, if there are no deadlocks in the generalised state space of the system which is checked by the Algorithm 2, then this system has no deadlocks. Fortunately, according to our analysis using Algorithm 2, in our model what ever the populations of N, P and Q are, there are no deadlocks in the linearised state space, so the protocol represented by the working cycle has no deadlocks.

Of course, we can do more qualitative analysis for content adaptation systems, as long as mature designs and concrete protocols are available.

7.6 Summary

This chapter has experimentally illustrated some fundamental characteristics of the fluid approximation of PEPA models, particularly with a focus on the convergence and consistence properties. In addition, this chapter has demonstrated the kind of performance measures supported by the fluid approximation approach and how they can be derived. In particular, this chapter revealed that the average performance is approximately governed by a set of corresponding nonlinear algebra equations, which can help to optimise a system in a simple and efficient way. For those metrics that cannot be obtained through this approach, we have proposed a numerical representation scheme-based stochastic simulation algorithm, along with a proof of the convergence of this algorithm. Detailed comparisons and analysis of performance derivation through different approaches have been presented.

This chapter has shown the performance analysis of the large scale content adaptation system. The numerical results from the evaluation were presented and analysed to determine the fac-

tors that affect the system performance. The analysis of the sensitivity and scalability of the response time has the potential to have great bearing on the continuing design of the content adaptation system and the capacity planning of future implementations. Moreover, this chapter has presented some qualitative analysis for a subsystem of content adaptation, in which invariance analysis as well as deadlock-checking was briefly discussed.

Chapter 8
Conclusions

8.1 Introduction

In this final chapter the main results of the thesis are summarised. Then several topics for further research are presented. This will conclude the main body of the thesis.

8.2 Summary

The work presented in this thesis addresses the technical and theoretical development for the formalism PEPA to overcome the state-space explosion problem and make it suitable to validate and evaluate large scale computer and communications systems. Our work embodies three levels of the process of performance modelling: model representation, computational approach and performance derivation.

As a high-level modelling language, the syntactic nature of PEPA makes the models easily understood by human beings. However, this advantage becomes a disadvantage for machines/computers (as well as for human beings) to directly carry out qualitative or quantitative analysis of PEPA. The numerical representation scheme for PEPA proposed in this thesis will make this analysis more direct and convenient. In this scheme, the correspondence between actions in PEPA and transitions in the underlying CTMC has been made to be one-to-one through the definition of labelled activities. Modified activity matrices have been defined based on labelled activities to capture the structural information of PEPA models, while transition rate functions were proposed to capture the timing information. Since all the information of a PEPA model is described and represented numerically by these definitions, based on them it is easy to extract and simulate the underlying CTMC and derive the fluid approximation. Moreover, this new presentation has lead to the finding: there is a P/T structure underlying each PEPA model, which reveals the strong connection between stochastic process algebras and stochastic Petri nets.

The scheme has been proved consistent with the original semantics of PEPA. An algorithm for automatically deriving the numerical representation scheme, and thus the underlying P/T structure and the fluid approximation, from any given PEPA model has been proposed. In addition, some investigations of the representation scheme were carried out. In particular, we have proved that using numerical vector forms to represent system states can significantly reduce the size of the state space of the system, i.e., the exponential increase of the size of the state space with the number of components can be reduced to at most polynomial increase.

Two important issues regarding structural and fluid-flow analysis were investigated in this thesis. These studies which were facilitated by the new representation scheme, have verified the associated computational approaches with PEPA.

The first issue was to develop approaches and techniques of the qualitative analysis for large scale PEPA models. Based on the underlying P/T structure and the theories developed for Petri nets, we have established powerful methods to derive and store the state space for a class of PEPA models without suffering the state-space explosion problem. Invariants in the context of PEPA were defined, and the method of how to find them was given. A conservation law was discussed, which was a particular kind of invariance, stating that the population of each component type is constant. The other kind of invariance expresses the coupling between different component types, i.e., the cooperations ensure that the numbers of local derivatives in the synchronised components always change together. Invariants could be used for qualitative reasoning about systems and to prove the convergence of the solution of the derived ODEs. As an important part of the structural analysis for PEPA, a new deadlock-checking approach has been proposed to avoid the state-space explosion problem, which can not only efficiently carry out checking for a particular system but can tell when and how a system structure leads to deadlocks.

The second issue was the theoretical development for the fluid approximation of PEPA models. In this thesis, based on the numerical representation scheme, an improved mapping from PEPA to ODEs has been proposed, which extended the current mapping semantics of fluid approximations. The derived ODEs could be considered as the limit of a family of density dependent CTMCs underlying the given PEPA model. We have established the fundamental characteristics of the derived ODEs, including the existence, uniqueness, boundedness and nonnegativeness of the solution. The convergence of the solution as time tends to infinity for general PEPA models, has been proved under a particular condition. This particular condition relates some

famous constants of Markov chains such as the spectral gap and the Log-Sobolev constant. For a class of PEPA models with two component types and one synchronisation, the convergence under some mild conditions that can be easily checked, as has been proved through a purely analytical approach. The coefficient matrices of the derived ODEs were studied: their eigenvalues are either zeros or have negative real parts. The structural property of invariance has been shown to play an important role in the proof of convergence for some PEPA models.

This thesis has established the consistency between the fluid approximation and the underlying CTMC for PEPA. We have proved that for a PEPA model without synchronisations, the solution of the derived ODEs converges and the limit coincides with the steady-state probability distribution of the underlying CTMC. For a model with synchronisations, the limit of the solution is consistent with the steady-state probability distribution corresponding to a family of underlying density dependent CTMCs underlying this model. This consistency has also been illustrated by the numerical comparisons between the fluid approximation and the simulations of the CTMCs underlying the content adaptation system.

Another important issue, deriving performance measures from large scale PEPA models, was also addressed in this thesis. We have shown that only the performance metrics that depend on the averaged population or that can be represented by a Markov reward function, can be derived from the fluid approximation of a PEPA model. In order to obtain those measures that cannot be obtained through this approach, we have provided a stochastic simulation algorithm which is based on the numerical representation scheme, together with a proof of the convergence of this algorithm.

These enhancements and investigations of PEPA have been applied to evaluate the large scale content adaptation systems, where both qualitative and quantitative analysis has been carried out. The main quantitative performance metric considered was the response time, derived from the fluid approximation of the model according to Little's law. Our results particularly showed that the average response time increased almost linearly as the number of users increased when the resources were fully utilised. On the other hand, for a fixed number of users, more resource could help to reduce the response time. But as more and more of a particular type of resource were added into the system, the lack of the other types of resource to match the increase in this particular type of resource would become the bottleneck, preventing further reduction of the response time. This suggested the need for capacity planning and optimisation of content adaptation systems. This thesis revealed that the average response time is approximately governed

by a set of corresponding nonlinear algebra equations. Based on these equations, sensitivity and scalability analysis, and capacity planning and optimisation for large scale systems could be made simpler and more efficient. In addition, some structural analysis of a subsystem of content adaptation has been presented.

PEPA with these associated techniques and methods is expected to have more applications in large scale systems in the future. Because there is no mature and complete protocol being proposed for content adaptation at the current moment, there is no corresponding complete protocol validation for the content adaptation system. However, we have provided many powerful techniques for the validation and evaluation for the future design of the content adaptation system. Therefore, not only large scale content adaptation systems but other computer and communications systems can be validated and evaluated by PEPA.

8.3 Limitations of the thesis and Future work

In this section, we address limitations of our work. Several topics for future research will be presented, which are motivated by those limitations as well as the achievements presented in this thesis.

The main limitation of our work is that many results shown in this thesis have certain restrictions. For example, the equivalent deadlock-checking theory (Theorem 4.5.5) presented in Chapter 4 was proved only for a class of EQ systems rather than general systems. The convergence result shown in Chapter 5 was obtained under a particular condition that cannot be easily verified currently. In Chapter 6, the convergence was demonstrated only for some typical classes of PEPA models. As a future work, we will investigate whether these restrictions are necessary and how to weaken or remove them if they are unnecessary. In particular, the conjecture that the particular condition is satisfied by any PEPA model, i.e. the open problem given in Chapter 5, should be proved or negated in the future. Since this condition relates the convergence problem to some famous constants of Markov chains, future investigation of this aspect can expand our understanding of the CTMCs underlying PEPA models.

In this thesis, the consistency between the ODEs and the CTMCs was demonstrated in an asymptotic sense, i.e. in the situation of the concentration level tending to infinity. In practice, as shown by the experiments in Chapter 7, the approximation between the ODEs and the CTMCs at finite concentration levels, both in terms of population and performance measure, are

of interest and thus need to be theoretically measured. Future work should involve theoretical error estimation for this approximation.

In Chapter 7, we have suggested the appropriate approaches to derive performance measures for small and large scale modelling respectively. However, we have not considered the middle scale case. A *middle scale* PEPA model is one where the numerical solution of the underlying CTMC is not available while the fluid-approximated result is not accurate. Although stochastic simulations can be used to derive performance measures for middle scale PEPA models, we still need to investigate how to derive these measures at a low computational cost for middle scale modelling.

In the context of probability theory and stochastic processes, the theory of large deviation was developed to study the asymptotics of probabilities of rare events with the help of variational problems, see [DZ98, FK06] for an introduction. This theory has been widely used for performance analysis [SW95] and been successfully applied to queueing networks [SW95, O'C95, SG95, AD98]. But to the best of our knowledge, there are no discussions of this theory in the context of stochastic process algebras. However, this theory could be expected to be helpful in the quantitative analysis using PEPA. For example, since the simulated performance measures are obtained or collected within finite time or finite iterations, their deviation from the expected ones, which can be represented by rare events, should be understood and estimated. Large deviation theory is a good method to investigate such deviation. We plan to apply this theory to the PEPA language in the future.

The future work mentioned above is mainly stimulated by the limitations of the current work. However, there are some topics for future research, which are motivated by the achievements. For example, we have established the relationship and connections between PEPA and Petri nets, which provides a new approach to investigate PEPA as well as Petri nets. Through the revealed connections, not only the work for Petri nets can be established parallel and analogously in the context of PEPA, but the theories and techniques developed for PEPA can also be applied to Petri nets. For example, our achievements on the fluid approximation such as the convergence and consistency results, can be expected to apply to Petri nets. To the best of our knowledge, although there is plenty of work on fluid approximation of Petri nets (e.g. [SR05, MRS06]), but there are no results on the problem of the convergence with respect to time, i.e. whether the solution converges as time tends to infinity. Investigation in this aspect, both in terms of PEPA and Petri nets, is one area of our future work.

As illustrated by the new deadlock-checking method presented in Chapter 4, the structure-based approach is powerful in the qualitative analysis of PEPA models. Future work is needed to study some other qualitative problems based on this approach, such as logical model-checking and qualitative reasoning [BGH09].

When the rates of change within the system model are generalised to allow activity rates to be governed by probability distributions rather than being deterministic, or more uncertainty is involved, the evolution of the system can be described by a set of stochastic differential equations (SDEs) [Hil05a]. Some preliminary study, particularly on the mapping semantics and some interpretation, has been presented in [Hay07b, Sle09]. But many problems have not been solved. For example, the positiveness of the solution of the derived SDEs, the relationship with stochastic simulations, as well as some characteristics (e.g. exponential stability [Mao94]). These problems lead to a direction for future research for PEPA.

Our research addressed in this thesis was carried out in the context of PEPA. Now the PEPA language has been extended into Bio-PEPA to model and analyse biochemical networks [CH09]. For the Bio-PEPA formalism, there are similar problems and issues under investigation, such as qualitative and quantitative analysis for Bio-PEPA models [Gue09]. Our achievements, such as the structure-based method for qualitative analysis and the theoretical developments for fluid approximations, can be expected to benefit and inspire the future research for Bio-PEPA.

References

[Abo] http://www.dcs.ed.ac.uk/pepa/about/.

[AD98] Rami Atar and Paul Dupuis, *Large deviations and queueing networks: methods for rate function identification*, http://arxiv.org/abs/math/9809204, 1998.

[ADLM08] Abdelhak Attou, Jie Ding, Dave Laurenson, and Klaus Moessner, *Performance modelling and evaluation of an adaptation management system*, Inernational Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2008) (Edinburgh, Scotland), June 2008.

[AM07] A. Attou and K. Moessner, *Context-aware service adaptation*, IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2007) (Athens, Greece), 2007.

[And91] William J. Anderson, *Continuous-time Markov chains: An applications-oriented appraoch*, Springer Series in Statistics, Springer-Verlag, 1991.

[ARI89] H. H. Ammar and S. M. Rezaul Islam, *Time scale decomposition of a class of generalized stochastic Petri net models*, IEEE Trans. Softw. Eng. **15** (1989), no. 6, 809–820.

[Bae05] J. C. M. Baeten, *A brief history of process algebra*, Theoretical Computer Science **335** (2005), no. 2-3, 131 – 146.

[BB] Sven Buchholz and Thmas Buchholz, *Replica placement in adaptative content distribution networks*, Proceedings of the 2004 ACM Symposium on Applied Computing, ACM, pp. 1705–1710.

[BB91] J. C. M. Baeten and J. A. Bergstra, *Real time process algebra*, Formal Aspects of Computing **3** (1991), no. 2, 142C188.

[BCMP75] Forest Baskett, K. Mani Chandy, Richard R. Muntz, and Fernando G. Palacios, *Open, closed, and mixed networks of queues with different classes of customers*, J. ACM **22** (1975), no. 2, 248–260.

[BDGK] J. Bradley, N. Dingle, S. Gilmore, and W. Knottenbelt, *Extracting passage times from PEPA models with the HYDRA tool: A case study*, Proceedings of the Nineteenth annual UK Performance Engineering Workshop, pp. 79–90.

[BG98] M. Bernardo and R. Gorrieri, *A tutorial on EMPA: A theory of concurrent processes with nondeterminism, priorities, probabilities and time*, Theoretical Computer Science **202** (1998), 1–54.

[BGdMT98] G. Bolch, S. Greiner, H. d. Meer, and K. S. Trivedi, *Queueing networks and Markov chains: Modelling and performance evaluation with computer science application*, John Wiley & Sons, INC., 1998.

[BGGW02] M. Butler, F. Giannetti, R. Gimson, and T. Wiley, *Device independence and the web*, IEEE Internet Computing **6** (2002), 81–86.

[BGH07] Jeremy T. Bradley, Stephen T. Gilmore, and Jane Hillston, *Analysing distributed internet worm attacks using continuous state-space approximation of process algebra models*, Journal of Computer and System Sciences (2007).

[BGH09] Paolo Ballarini, Maria Luisa Guerriero, and Jane Hillston, *Qualitative reasoning of stochasitc models and the role of flux*, Proceedings of 8th Workshop on Process Algebra and Stochastically Timed Activities (PASTA'09), 2009.

[BH97] H. Bohnenkamp and B. Haverkort, *Decomposition methods for the solution of stochastic process algebra models: a proposal*, Proc. of 5th Process Algebra and Performance Modelling Workshop, 1997.

[BH99] Henrik C. Bohnenkamp and Boudewijn R. Haverkort, *Semi-numerical solution of stochastic process algebra models*, ARTS '99: Proceedings of the 5th International AMAST Workshop on Formal Methods for Real-Time and Probabilistic Systems (London, UK), Springer-Verlag, 1999, pp. 228–243.

[BHKS08] Jeremy T. Bradley, Richard Hayden, William J. Knottenbelt, and Tamas Suto, *Extracting Response Times from Fluid Analysis of Performance Models*, SIPEW'08, SPEC International Performance Evaluation Workshop, Darmstadt, 27-28 June 2008, Lecture Notes in Computer Science, vol. 5119, May 2008, pp. 29–43.

[BHM+09] Soufiene Benkirane, Jane Hillston, Chris McCaig, Rachel Norman, and Carron Shankland, *Improved continuous approximation of PEPA models through epidemiological examples*, Electron. Notes Theor. Comput. Sci. **229** (2009), no. 1, 59–74.

[BHR+99] Staffan Björk, Lars Erik Holmquist, Johan Redström, Rolf Danielsson, Jussi Karlgren, and Kristofer Franzén, *WEST: A web browser for small terminals*, ACM conference on User Interface Standards and Technology (UIST) (Asheville, North Carolina, USA), November 1999.

[BID06] J. Bush, J. Irvine, and J. Dunlop, *Removing the barriers to ubiquitous services: A user perspective*, First International Workshop on Personalized Networks(PerNets'06), July 2006.

[BK04] Jeremy T. Bradley and William J. Knottenbelt, *The ipc/HYDRA tool chain for the analysis of PEPA models*, QEST'04, 1st IEEE International Conference on the Quantitative Evaluation of Systems, August 2004, pp. 334–335.

[BLT94] T. Bolognesi, F. Lucidi, and S. Trigila, *Converging towards a timed LOTOS standard*, Comput. Standards Interfaces **16** (1994), 87–118.

[BMSM+08] Roberto Barbuti, Andrea Maggiolo-Schettini, Paolo Milazzo, Paolo Tiberi, and Angelo Troina, *Stochastic calculus of looping sequences for the modelling and simulation of cellular pathways*, Transactions on Computational Systems Biology IX (Berlin, Heidelberg), Lecture Notes in Bioinformatics, Springer-Verlag, 2008, pp. 86–113.

[BMSMT06a] Roberto Barbuti, Andrea Maggiolo-Schettini, Paolo Milazzo, and Angelo Troina, *Bisimulation congruences in the calculus of looping sequences*, Theoretical Aspects of Computing - ICTAC 2006, Lecture Notes in Computer Science, vol. 4281, Springer Berlin / Heidelberg, 2006, pp. 93–107.

[BMSMT06b] _____, *A calculus of looping sequences for modelling microbiological systems*, Fundam. Inf. **72** (2006), no. 1-3, 21–35.

[Bou94] R. J. Boucherie, *A characterization of independence for competing Markov chains with applications to stochastic Petri nets*, IEEE Trans. Softw. Eng. **20** (1994), no. 7, 536–544.

[Bru98] Richard A. Brualdi, *Introductory combinatorics*, third ed., Prentice Hall, 1998.

[BS94] Richard J. Boucherie and Matteo Sereno, *On the traffic equations for batch routing queueing networks and stochastic Petri nets*, 1994, ftp://ftp.inria.fr/associations/ERCIM/research_rep.

[BT93] A. Blakemore and S.K. Tripathi, *Automated time scale decomposition and analysis of stochastic Petri nets*, Proc. of 5th International Workshop on Petri Nets and Performance Models, Oct 1993, pp. 248–257.

[Buc94] B. Buchholz, *Compositional analysis of a Markovian process algebra*, Proc. of 2nd Process Algebra and Performance Modelling Workshop (M. Rettelbach and U. Herzog, eds.), 1994.

[Bus06] J. Bush, *Architecture for ubiqiutous systems*, Deliverable, MobileVCE Core 4, August 2006.

[BZ99] Zdzisław Brzeźniak and Tomasz Zastawniak, *Basic stochastic processes*, Springer Undergraduate Mathematics Series, Springer, 1999.

[CCC05a] C. Canali, V. Cardellini, and M. Colajanni, *Performance comparison of distributed architectures for content adaptation and delivery of web resources*, Proc. of the International Workshop on Services and Infrastructure for the Ubiquitous and Mobile Internet (SIUMI'05), 2005.

[CCC05b] _____, *A two-level distributed architecture for efficient web content adaptation and delivery*, Proc. of the IEEE/IPSJ Symposium on Applications and the Internet (SAINT'05), January-February 2005.

[CCE+03] E. Cecchet, A. Chanda, S. Elnikety, J. Marguerite, and W. Zwaenepoel, *Performance comparison of middleware architectures for generating dynamic web content*, Lecture Notes in Computer Science, vol. 2672, Springer Berlin/Heidelberg, 2003.

[CCL05] C. Canali, S. Casolari, and R. Lancellotti, *Architectures for scalable and flexible web personalization services*, Proc. of the International Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications (AAA-IDEA'05), June 2005.

[CDGH08] Allan Clark, Adam Duguid, Stephen Gilmore, and Jane Hillston, *Espresso, a little coffee*, PASTA '08: Proceedings of the 7th Workshop on Process Algebra and Stochastically Timed Acitvities, 2008.

[CEV00] Surendar Chandra Carla, Carla Schlatter Ellis, and Amin Vahdat, *Differentiated multimedia web services using quality aware transcoding*, IEEE Journal on Selected Areas in Communications **18** (2000), no. 12.

[CGH05] Muffy Calder, Stephen Gilmore, and Jane Hillston, *Automatically deriving ODEs from process algebra models of signalling pathways*, Proc. of 3rd International Workshop on Computational Methods in Systems Biology (CMSB) (Edinburgh) (Gordon Plotkin, ed.), April 2005, pp. 204–215.

[CH96] Graham Clark and Jane Hillston, *Towards automatic derivation of performance measures from PEPA models*, Proceedings of UKPEW, 1996, pp. 17–26.

[CH02] G. Clark and J. Hillston, *Product form solution for an insensitive stochastic process algebra structure*, Performance Evaluation **50** (2002), no. 2–3, 129–151.

[CH09] Federica Ciocchetta and Jane Hillston, *Bio-PEPA: A framework for the modelling and analysis of biological systems*, Theoretical Computer Science **410** (2009), no. 33-34, 3065 – 3084.

[Che04] H. Chen, *An intelligent broker architecture for pervasive context-aware system*, Ph.D. thesis, University of Maryland, USA, 2004.

[Che05] Mu-FA Chen, *Eigenvalues, inequalities, and ergodic theory*, Probability and its Applications, Springer, 2005.

[Chi98] G. Chiola, *Timed Petri nets*, MATCH Summer School (Spain), Septemper 1998.

[Cla09] Allan Clark, *Response-time profiles for PEPA models compiled to ODEs*, PASTA '09: Proceedings of the 8th Workshop on Process Algebra and Stochastically Timed Acitvities, 2009.

[CM99] Gianfranco Ciardo and Andrew S. Miner, *A data structure for the efficient Kronecker solution of GSPNs*, Proc. 8th Int. Workshop on Petri Nets and Performance Models (PNPM99, IEEE Comp. Soc. Press, 1999, pp. 22–31.

[Cou77] Pierre Jacques Courtois, *Decomposability: queueing and computer system applications*, Academic Press, New York, 1977.

[CT91] G. Ciardo and K.S. Trivedi, *A decomposition approach for stochastic Petri net models*, Proc. of the Fourth International Workshop on Petri Nets and Performance Models (PNPM'91), Dec 1991, pp. 74–83.

[CTS98] J. M. Colom, E. Teruel, and M. Silva, *Logical properties of P/T system and their analysis*, MATCH Summer School (Spain), Septemper 1998.

[CWW07] W. H. Cheng, C. W. Wang, and J. L. Wu, *Video adaptation for small display based on content recomposition*, IEEE Transactions on Circuits and Systems for Video Technology **17** (2007), 43–58.

[Des] http://dl.kr.org/.

[Dey00] A. K. Dey, *Providing architectural support for building context-aware applications*, Ph.D. thesis, Georgia Institute of Technology, USA, 2000.

[DHL] Jie Ding, Jane Hillston, and Dave Laurenson, *Evaluating the response time of large scale content adaptation systems using peformance evaluation process algebra*, accepted by IEEE International Communications Conference 2010.

[DHL09] ———, *Performance modelling of content adaptation for a personal distributed environment*, Wireless Personal Communications **48** (2009), 93–112.

[DHR95] S. Donatelli, J. Hillston, and M. Ribaudo, *A comparison of Performance Evaluation Process Algebra and Generalized Stochastic Petri Nets*, Proc. 6th International Workshop on Petri Nets and Performance Models (Durham, North Carolina), 1995.

[DK00] Susanna Donatelli and Peter Kemper, *Integrating synchronization with priority into a Kronecker representation*, TOOLS '00: Proceedings of the 11th International Conference on Computer Performance Evaluation: Modelling Techniques and Tools (London, UK), Springer-Verlag, 2000, pp. 203–215.

[DLM08] Jie Ding, Ning Li, and Klaus Moessner, *Performance evaluation for a distributed adaptation management framework in content delivery networks (manuscript)*, The University of Edinburgh, 2008.

[DN08] R. W. R. Darling and J. R. Norris, *Differential equation approximations for Markov chains*, Probability Surveys **5** (2008), 37–79.

[Don94] Susanne Donatelli, *Superposed generalized stochastic Petri nets: definition and efficient solution.*, Lecture Notes in Computer Science; Application and Theory of Petri Nets 1994, Proceedings 15th International Conference, Zaragoza, Spain (R. Valette, ed.), vol. 815, Springer-Verlag, 1994, pp. 258–277.

[DZ98] Amir Dembo and Ofer Zeitouni, *Large deviations techniques and applications*, Springer, 1998.

[EK86] Stewartn N. Ethier and Thomas G. Kurtz, *Markov processes: Characterization and convergence*, John Wiley & Sons, Inc., 1986.

[EKBS04] Khalil El-Khatib, Gregor V. Bochmann, and Abdulmotaleb El Saddik, *A distributed content adaptation framework for content distribution networks*, http://beethoven.site.uottawa.ca/dsrg/PublicDocuments/Publications/ElKh04c.pdf, 2004.

[ER00] Amani Helmi El-Rayes, *Analysing performance of open queueing systems with stochastic process algebras*, Ph.D. thesis, School of Computer Science, University of Birmingham, 2000.

[FAD+97] Brian Fisher, Makrina Agelidis, John Dill, Paul Tan, Gerald Collaud, and Chris Jones, *Czweb: Fish-eye views for visualizing the world-wide web*, Proc. of the 7th Int. Conf. on Human-Computer Interaction (HCI International '97, 1997, pp. 719–722.

[FGC+97] Armando Fox, Steven D. Gribble, Yatin Chawathe, Eric A. Brewer, and Paul Gauthier, *Cluster-based scalable network services*, 16th ACM Symp. On Operating Systems Principles (Saint-Malo, France), 1997, pp. 78–91.

[FK06] Jin Feng and Thomas G. Kurtz, *Large deviations for stochastic processes*, Mathematical Surveys and Monographs, vol. 131, American Mathematical Society, 2006.

[FLQ04] J. M. Fourneau, M. Lecoz, and F. Quessette, *Algorithms for an irreducible and lumpable strong stochastic bound*, Linear Algebra and Its Applications **386** (2004), 167–185.

[FLW06] Harald Fecher, Martin Leucker, and Verena Wolf, *Don't know in probabilistic systems*, Model Checking Software. LNCS, Springer, 2006, pp. 71–88.

[FPS07] J. M. Fourneau, B. Plateau, and W. Stewart, *Product form for stochastic automata networks*, Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007, pp. 1–10.

[Gal08] Vashti Galpin, *Continuous approximation of PEPA models and Petri nets*, Proceedings of the European Simulation and Modelling Conference (ESM 2008) (Le Havre, France), 27-29 October 2008, pp. 492–499.

[GHR92] N. Götz, U. Herzog, and M. Rettelbach, *TIPP– a language for timed processes and performance evaluation*, Tech. report, Tech. Rep.4/92, IMMD7, University of Erlangen-Nörnberg, Germany, Nov. 1992.

[GHR97] Stephen Gilmore, Jane Hillston, and Laura Recalde, *Elementary structural analysis for PEPA*, Tech. report, The University of Edinburgh, UK, December 1997.

[GHR01] Stephen Gilmore, Jane Hillston, and Marina Ribaudo, *An efficient algorithm for aggregating PEPA models*, IEEE Trans. Softw. Eng. **27** (2001), no. 5, 449–464.

[GHS08] Nil Geisweiller, Jane Hillston, and Marco Stenico, *Relating continuous and discrete PEPA models of signalling pathways*, Theoretical Computer Science **404** (2008), no. 1-2, 97–111.

[Gil76] Daniel T. Gillespie, *A general method for numerically simulating the stochastic time evolution of coupled chemical reactions*, Journal of Computational Physics **22** (1976), no. 4, 403 – 434.

[Gil05] S. Gilmore, *Continuous-time and continuous-space process algebra*, Process Algebra and Stochastically Timed Activities (PASTA'05), 2005.

[Gue09] Maria Luisa Guerriero, *Qualitative and quantitative analysis of a Bio-PEPA model of the Gp130/JAK/STAT signalling pathway*, T. Comp. Sys. Biology **11** (2009), 90–115.

[Han94] H. Hansson, *Time and probability in formal design of distributed systems*, Real-Time Safety Critical Systems, vol. 1, Elsevier Science Inc., 1994.

[Har03] Peter G. Harrison, *Turning back time in Markovian process algebra*, Theor. Comput. Sci. **290** (2003), no. 3, 1947–1986.

[Har04] Peter G. Harrison, *Reversed processes, product forms and a non-product form*, Linear Algebra and its Applications **386** (2004), 359 – 381.

[Har06] P.G. Harrison, *Process algebraic non-product-forms*, Electronic Notes in Theoretical Computer Science **151** (2006), no. 3, 61 – 76.

[Har09] Peter G. Harrison, *Product-forms and functional rates*, Performance Evaluation **66** (2009), no. 11, 660 – 663.

[Hay07a] Richard Hayden, *Addressing the state space explosion problem for PEPA models through fluid-flow approximation*, http://pubs.doc.ic.ac.uk/fluid-spa-modelling/, July 2007.

[Hay07b] Richard A. Hayden, *Addressing the state space explosion problem for PEPA models through fluid-flow approximation (bachelor thesis)*, Imperial College, 2007.

[HB08] Richard A. Hayden and Jeremy T. Bradley, *ODE-based general moment approximations for PEPA*, Proceedings of 7th Workshop on Process Algebra and Stochastically Timed Activities (PASTA'08), 2008.

[HB09] Richard Hayden and Jeremy T. Bradley, *Fluid passage-time calculation in large Markov models*, Tech. report, Department of Computing, Imperial college, UK, May, 2009.

[HB10] _____, *Evaluating fluid semantics for passive stochastic process algebra cooperation*, Performance Evaluation (2010), In Press.

[Her98] H. Hermanns, *Interactive Markov chains*, Ph.D. thesis, Universität Erlangen-Nürnberg, Germany, 1998.

[HHK02] H. Hermanns, U. Herzog, and J. P. Katoen, *Process algebra for performance evaluation*, Theoretical Computer Science **264** (2002), 43–87.

[Hil96] J. Hillston, *A compositional approach to performance modelling (phd thesis)*, Cambridge University Press, 1996.

[Hil98] Jane Hillston, *Exploiting structure in solution: Decomposing compositional models*, Proceedings of 6th International Workshop on Process Algebra and Performance Modelling, Springer-Verlag, 1998, pp. 1–15.

[Hil05a] J. Hillston, *Fluid flow approximation of PEPA models*, International Conference on the Quantitative Evaluation of Systems (QEST'05), IEEE Computer Society, 2005.

[Hil05b] _____, *Tuning systems: From composition to performance*, The Computer Journal **48** (2005), no. 4, 385–400, The Needham Lecture paper.

[HK01] Jane Hillston and Leïla Kloul, *An efficient Kronecker representation for PEPA models*, PAPM-PROBMIV '01: Proceedings of the Joint International Workshop on Process Algebra and Probabilistic Methods, Performance Modeling and Verification (London, UK), Springer-Verlag, 2001, pp. 120–135.

[HK07] Jane Hillston and Leïla Kloul, *Formal techniques for performance analysis: blending SAN and PEPA*, Formal Aspects of Computing **19** (2007), 3–33.

[HLT89] W. Henderson, D. Lucic, and P. G. Taylor, *A net level performance analysis of stochastic Petri nets.*, J. Aust. Math. Soc. B. **31** (1989), no. 2, 176–187.

[HM95] Jane Hillston and Vassilis Mertsiotakis, *A simple time scale decomposition technique for stochastic process algebras*, The Computer Journal, 1995, pp. 566–577.

[HM09] Michael Harrison and Mieke Massink, *Modelling interactive experience, function and performance in ubiquitous systems*, Proc. of 4th International Workshop on Practical Applications of Stochastic Modelling (PASM'09) (Imperial College, Lodon), 2009.

[Hoa85] C. A. R. Hoare, *Communicating sequential processes*, Prentice Hall, 1985.

[How71] R. A. Howard, *Dynamic probability system: Volume 2, semi-Markov decision processes*, John Wiley & Sons, 1971.

[HRRS01] J. Hillston, L. Recalde, M. Ribaudo, and M. Silva, *A comparison of the expressiveness of SPA and bounded SPN models*, Proceedings of the 9th International Workshop on Petri Nets and Performance Models (Aachen, Germany) (B. Haverkort and R. German, eds.), IEEE Computer Science Press, 2001.

[HT91] William Henderson and Peter G. Taylor, *Embedded processes in stochastic Petri nets*, IEEE Trans. Softw. Eng. **17** (1991), no. 2, 108–116.

[HT99] Jane Hillston and Nigel Thomas, *Product form solution for a class of PEPA models*, Performance Evaluation **35** (1999), no. 3–4, 171–192.

[htt] http://www.mobilevce.com.

[HW90] J. H. Hubbard and B. H. West, *Differential equations: A dynamical systems approach (higher-dimensional systems)*, Texts in Applied Mathematics, no. 18, Springer, 1990.

[Joh08] M. John, *A spatial extension to the π-Calculus*, Electronic Notes in Theoretical Computer Science **194** (2008), no. 3, 133–148.

[JS90] C-C. Jou and S. A. Smolka, *Equivalences, congruences and complete axiomatizations of probabilistic processes*, Lecture Notes in Computer Science, vol. 458, Springer-Verlag, August 1990, pp. 367–383.

[JTW+07] H. Jiang, G. Tong, H. Wei, I. L. Yen, and F. Bastani, *A flexible content adaptation system using a rule-based approach*, IEEE Transactions on Knowledge and Data Engineering **19** (2007), 127–140.

[Kem96] Peter Kemper, *Numerical analysis of superposed GSPNs*, IEEE Trans. Softw. Eng. **22** (1996), no. 9, 615–628.

[Kie02] Andrzej M. Kierzek, *STOCKS: STOChastic kinetic simulations of biochemical systems with Gillespie algorithm*, Bioinformatics **18** (2002), no. 3, 470–481.

[KKLW07] Joost-Pieter Katoen, Daniel Klink, Martin Leucker, and Verena Wolf, *Three-valued abstraction for continuous-time Markov chains*, Formal Methods for Industrial Critical Systems, Lecture Notes in Computer Science, vol. 4590, Springer Berlin / Heidelberg, 2007, pp. 311–324.

[KM06] Ahmed Reda Kaced and Jean-Claude Moissinac, *SEMAFOR: A framework for authentication of adaptive multimedia content and delivery for heterogeneous networks*, International Conference on Internet Surveillance and Protection (ICISP '06), 2006, pp. 28–28.

[KN06] Céline Kuttler and Joachim Niehren, *Gene regulation in the π-calculus: Simulating cooperativity at the lambda switch*, Lecture Notes in Computer Science, vol. 4230, Springer Berlin / Heidelberg, 2006, pp. 24–55.

[Kur70] Thomas G. Kurtz, *Solutions of ordinary differential equations as limits of pure jump Markov processes*, Journal of Applied Probability **7** (1970), no. 1, 49–58.

[KV05] Leïla Kloul and Fabrice Valois, *Investigating unfairness scenarios in MANET using 802.11b*, PE-WASUN '05: Proceedings of the 2nd ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks (New York, NY, USA), ACM, 2005, pp. 1–8.

[LH05] T. Laakko and T. Hiltunen, *Adapting web content to mobile user agents*, IEEE Internet Computing **9** (2005), 46–53.

[Li06] Ning Li, *Requirements definition for the content/service support adaptation architecture*, Deliverable, MobileVCE Core 4, April 2006.

[Lit61] J. D. C. Little, *A proof of the queueing formula $l = \lambda w$*, Operations Research **9** (1961), 383 – 387.

[LL97] L. Léonard and G. Leduc, *An introduction to ET-LOTOS for the description of time-sensitive systems*, Networks and ISDN Systems **29** (1997), no. 3, 271–292.

[LL02a] W. Y. Lum and Francis C. M. Lau, *A context-aware decision engine for content adaptation*, IEEE Pervasive Computing **1** (2002), no. 3, 41–49.

[LL02b] ———, *On balancing between transcoding overhead and spatial consumption in content adaptation*, Proceedings of the 8th annual international conference on mobile computing and networking (Atlanta, USA), September 2002, pp. 239–250.

[LM06] N. Li and K. Moessner, *The MVCE management framework for context-aware content and service adaptation*, 1st International Workshop Semantic Media Adaptation and Personalization, December 2006.

[LM07] _____, *The MVCE knowledge-based content and service adaptation management framework*, Workshop on Applications and Services in Wireless Networks (Santander, Spain), 2007.

[LPQ+03] P. Lecca, C. Priami, P. Quaglia, B. Rossi, C. Laudanna, and G. Costantin, *Language modeling and simulation of autoreactive lymphocytes recruitment in inflamed brain vessels*, SIMULATION: Transactions of The Society for Modeling and Simulation International **80** (2003), 273–288.

[LR91] Aurel A. Lazar and Thomas G. Robertazzi, *Markovian Petri nets protocols with product form solution*, Perform. Eval. **12** (1991), no. 1, 67–77.

[LS91] K. Larsen and A. Skou, *Bisimulation through probabilistic testing*, Information and Computation **94** (1991), no. 1, 1–28.

[Mao94] Xuerong Mao, *Exponential stability of stochastic differential equations*, Monographs and Textbooks in Pure and Applied Mathematics, vol. 182, Marcel Dekker, Inc., 1994.

[MBC+00] Wei-Ying Ma, Ilja Bedner, Grace Chang, Allan Kuchinsky, and HongJiang Zhang, *A framework for adaptive content delivery in heterogeneous network environments*, San Jose, California, USA, 2000, SPIE Multimedia Computing and Networking.

[Mer97] Vassilis Mertsiotakis, *Time scale decomposition of stochastic process algebra models*, Proc. of 5th Process Algebra and Performance Modelling Workshop, 1997.

[Mer98] V. Mertsiotakis, *Approximate analysis methods for stochastic process algebras*, Ph.D. thesis, Universität Erlangen-Nürnberg, Erlangen, 1998.

[Mil83] R. Milner, *Calculi for synchrony and asynchrony*, Theoretical Computer Science **25** (1983), no. 3, 267–310.

[Mil89] _____, *Communication and concurrency*, Prentice Hall, 1989.

[Mil07] Paolo Milazzo, *Qualitative and quantitative formal modeling of biological systems*, Ph.D. thesis, University of Pisa, 2007.

[MR80] G. Memmi and G. Roucairol, *Linear algebra in net theory*, Net Theory and Applications (Berlin) (W. Brauer, ed.), Lecture Notes in Computer Science, vol. 84, Springer Verlag, 1980, pp. 213–223.

[MRS06] C. Mahulea, L. Recalde, and M. Silva, *On performance monotonicity and basic servers semantics of continuous Petri nets*, WODES06: 8th International Workshop on Discrete Event Systems (Ann Arbor, USA), July 2006, pp. 345—-351.

[MS96] Vassilis Mertsiotakis and Manuel Silva, *A throughput approximation algorithm for decision free processes*, Proc. of 6th Process Algebra and Performance Modelling Workshop, 1996, pp. 161–178.

[MS97] _____, *Throughput approximation of decision free processes using decomposition*, In Proc. of the 7th Int. Workshop on Petri Nets and Performance Models, 1997, pp. 174–182.

[MSCS99] R. Mohan, J. R. Smith, and L. Chung-Sheng, *Adapting multimedia internet content for universal access*, IEEE Transactions on Multimedia **1** (1999), 104–114.

[MT89] F. Moller and C. Tofts, *A temporal calculus for communicating systems*, Lecture Notes in Computer Science, vol. 458, Springer-Verlag, August 1989.

[Nor98] J.R. Norris, *Markov chains*, Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, July 1998.

[NS92] X. Nicollin and J. Sifakis, *An overview and synthesis on timed process algebras*, Lecture Notes in Computer Science, vol. 600, Springer-Verlag, 1992, pp. 526–548.

[O'C95] Neil O'Connell, *Large deviations in queueing networks*, ftp://www.stp.dias.ie/DAPG/dapg9413.ps, 1995.

[OWL] http://www.w3.org/.

[Per91] Lawrence Perko, *Differential equations and dynamical systems*, Texts in Applied Mathematics, no. 7, Springer-Verlag, 1991.

[PKP03] A. Pashtan, S. Kollipara, and M. Pearce, *Adapting content for wireless web services*, IEEE Internet Computing **7** (2003), 79–85.

[Pla84] B. Plateau, *De l'evaluation du parallélisme et de la synchronisation*, Ph.D. thesis, Université de Paris XII, Orsay, France, 1984.

[Pla85] Brigitte Plateau, *On the stochastic structure of parallelism and synchronization models for distributed algorithms*, SIGMETRICS Perform. Eval. Rev. **13** (1985), no. 2, 147–154.

[PQ05] Corrado Priami and Paola Quaglia, *Beta binders for biological interactions*, Computatinal Methods in Systems Biology, Lecture Notes in Computer Science, vol. 3082, Springer Berlin / Heidelberg, 2005, pp. 20–33.

[PRSS01] Corrado Priami, Aviv Regev, Ehud Shapiro, and William Silverman, *Application of a stochastic name-passing calculus to representation and simulation of molecular processes*, Inf. Process. Lett. **80** (2001), no. 1, 25–31.

[Rib95] Marina Ribaudo, *Stochastic Petri net semantics for stochastic process algebras*, Proceedings of the Sixth International Workshop on Petri Nets and Performance Models (Washington DC, USA), IEEE Computer Society, 1995.

[RS94] Michael Rettelbach and Markus Siegle, *Compositional minimal semantics for the stochastic process algebra TIPP*, Proc. of 2nd Process Algebra and Performance Modelling Workshop, 1994, p. pages.

[RV06] T. Razafindralambo and Fabrice Valois, *Performance evaluation of backoff algorithms in 802.11 ad-hoc networks*, PE-WASUN '06: Proceedings of the 3rd ACM international workshop on Performance evaluation of wireless ad hoc, sensor and ubiquitous networks (New York, NY, USA), ACM, 2006, pp. 82–89.

[SA61] H. A. Simon and A. Ando, *Aggregation of variables in dynamic systems*, Econometrica **29** (1961), 111–138.

[SC97] Laurent Saloff-Coste, *Lectures on finite Markov chains*, Lecture Notes in Mathematics (Berlin), vol. 1665, Springer, 1997.

[Sch95] S. Schneider, *An operational semantics for timed CSP*, Inform. and Comput. **116** (1995), 193–213.

[SD83] D. Stoyan and D. J. Daley, *Comparison methods for queues and other stochastic models*, John wiley & Sons, New Yourk, USA, 1983.

[Ser95] M. Sereno, *Towards a product form solution for stochastic process algebras*, The Computer Journal **38** (1995), no. 7, 622–632.

[SG95] A. Simonian and J. Guibert, *Large deviations approximation for fluid queues fed by a large number of on/off sources*, IEEE Journal on Selected Areas in Communications **13** (1995), no. 6, 1017–1027.

[Shi96] A. N. Shiryaev, *Probability*, second ed., Graduate Texts in Mathematics, no. 95, Springer-Verlag New York Inc., 1996.

[Sle09] Joris Slegers, *A Langevin interpretation of PEPA models*, Proc. of 4th International Workshop on Practical Applications of Stochastic Modelling (PASM'09) (Imperial College, Lodon), 2009.

[Smi09a] Michael J. A. Smith, *Abstraction and model checking in the Eclipse PEPA plug-in*, PASTA '09: Proceedings of the 8th Workshop on Process Algebra and Stochastically Timed Acitvities, 2009.

[Smi09b] ———, *Compositional abstraction of PEPA models*, 2009, http://lanther.co.uk/papers/PEPA_abstraction.pdf.

[Smi09c] ———, *A tool for abstraction and model checking of PEPA models*, 2009, http://lanther.co.uk/papers/PEPA_tool.pdf.

[SR05] Manuel Silva and Laura Recalde, *Continuization of timed Petri nets: From performance evaluation to observation and control.*, Lecture Notes in Computer Science: Applications and Theory of Petri Nets 2005: 26th International Conference, ICATPN 2005, Miami, USA, June 20-25, 2005. / Gianfranco Ciardo, Philippe Darondeau (Eds.), vol. 3536, Springer Verlag, june 2005, pp. 26–47.

[STC96] M. Silva, E. Teruel, and J. M. Colom, *Linear algebraic and linear programming techniques for the analyisis of place/transition net systems*, Lecture Notes in Computer Science, vol. 1491, Springer-Verlag, 1996.

[SW95] Adam Shwartz and Alan Weiss, *Large deviations for performance analysis: Queues, communication, and computing*, Chapman & Hall, 1995.

[TBID08] H. Tarus, J. Bush, J. Irvine, and J. Dunlop, *Multi-agent mediated electronic-marketplace for adaptation services*, 5th IEEE Consumer Communications and Networking Conference (CCNC) (Las Vegas, Nevada, USA), Jan. 2008.

[TDG09] Mirco Tribastone, Adam Duguid, and Stephen Gilmore, *The PEPA eclipse plug-in*, SIGMETRICS Perform. Eval. Rev. **36** (2009), no. 4, 28–33.

[TH00] J. Tomasik and J. Hillston, *Transforming PEPA models to obtain product form bounds*, Tech. Report EDI-INF-RR-0009, Laboratory for Foundations of Computer Science, The University of Edinburgh, February 2000.

[Tho06] Nigel Thomas, *Approximation in non-product form finite capacity queue systems*, Future Gener. Comput. Syst. **22** (2006), no. 7, 820–827.

[Tho09] Nigel Thomas, *Using ODEs from PEPA models to derive asymptotic solutions for a class of closed queueing networks*, PASTA '09: Proceedings of the 8th Workshop on Process Algebra and Stochastically Timed Acitvities, 2009.

[Tof92] C. Tofts, *Describing social insect behaviour using process algebra*, Transactions of the Society for Computer Simulation **9** (1992), no. 4, 227–283.

[Tri09] Mirco Tribastone, *Differential analysis of PEPA models*, Proceedings of 8th Workshop on Process Algebra and Stochastically Timed Activities (PASTA'09), 2009.

[TZ08] Nigel Thomas and Yishi Zhao, *Fluid flow analysis of a model of a secure key distribution centre*, Proceedings of 24th Annual UK Performance Engineering Workshop (Imperial College, London), 2008.

[UDD] http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm.

[UHCW06] M. Ullah, Schmidt H., K. H. Cho, and O. Wolkenhauer, *Deterministic modelling and stochastic simulation of biochemical pathways using matlab.*, Syst Biol (Stevenage) **153** (2006), no. 2, 53–60.

[VCH03] A. Vetro, C. Christopoulos, and S. Huifang, *Video transcoding architectures and techniques: an overview*, IEEE Signal Processing Magazine **20** (2003), 18–29.

[Wan05] Feng-Yu Wang, *Functional inequalities, Markov semigroups and spectral theory*, Science Press (China), 2005.

[WHFG92] R. Want, A. Hopper, V. Falcao, and J. Gibbons, *The active badge location system*, ACM Transactions on Information System **10** (1992), no. 1, 91–102.

[WLH09a] Hao Wang, Dave Laurenson, and Jane Hillston, *A general performance evaluation framework for network selection strategies in heterogeneous wireless networks*, submitted to IEEE Transaction on Mobile Computing (2009).

[WLH09b] _____, *A reservation optimised advance resource reservation scheme for deploying RSVP in mobile environments*, Wireless Personal Communications (2009), published online first.

[YL03] L. Wai Yip and F. C. M. Lau, *User-centric content negotiation for effective adaptation service in mobile computing*, IEEE Transactions on Software Engineering **29** (2003), 1100–1111.

[ZT08] Yishi Zhao and Nigel Thomas, *Approximate solution of a PEPA model of a key distribution centre*, Performance Evaluation: Metrics, Models and Benchmarks, Lecture Notes in Computer Science, vol. 5119, Springer Berlin / Heidelberg, 2008, pp. 44–57.

[ZT09] _____, *Efficient solutions of a PEPA model of a key distribution centre*, Performance Evaluation **In Press, Corrected Proof** (2009), –.

Appendix A
From Process Algebra to Stochastic Process Algebra

In this appendix, we will give an overview of the history from process algebras to stochastic process algebras. This part is mainly based on the literature [Bae05, NS92, Han94].

A.1 Process algebra

A process algebra is a formal description technique for the study of the behaviour of concurrent systems by algebraic means. It models parallel or distributed systems by their algebra and provides apparatus for reasoning about the structure and behaviour of the model. In the process algebra approach systems are modelled as collections of entities called agents or processes, which execute atomic actions. Process algebra can be viewed as an approach to both automata theory and concurrency theory, since a process, in process algebra, is described as an automaton which has several states and actions and it is also able to interact with other processes.

The history of process algebra can be traced back to the early seventies of the twentieth century. The early work centred around giving semantics to programming languages involving a parallel construct. After two breakthroughs as pointed out in [Bae05], replacing the idea that a program is a transformation from input to output by an approach where all intermediate states are important, and replacing the notion of global variables by the paradigm of message passing and local variables, the process algebras CCS [Mil89] and CSP [Hoa85] were developed. By the early eighties, process algebra was finally established as a separate area of research and gave underlying theories to many parallel and distributed systems, extending formal language and automata theory with the central ingredient of interaction. For the details of the history of process algebra, please refer to [Bae05].

The most widely used pure process algebras are CCS and CSP. In CCS, any action may be internal to an agent or may constitute the interaction between neighbouring agents. Agents

may proceed with their internal actions simultaneously, but the semantics given to the language imposes an interleaving on such concurrent behaviour, i.e., it does not allow two actions to occur simultaneously. The operational semantics of CCS uses a labelled transition system, from which a derivative tree or graph may be constructed. In the tree or graph, language terms form the nodes and transitions (actions) are the arcs. The graph records all the possible evolutions of a language expression or model [Hil05b]. This structure is the basis of the bisimulation style of equivalence as well as a useful tool for reasoning about agents and the system they represent.

CCS and CSP models have been used extensively for the specification and design of concurrent systems by deriving functional or qualitative properties such as freedom from deadlock and fairness.

A.2 Timed process algebra

In pure process algebra an action is instantaneous and only relative timing is represented via the traces of the process since time is abstracted away within a process. There are many variants of CCS with timing. For the purpose of synchronisation, time is incorporated in the process algebra such as SCCS [Mil83] in the simplest way, which synchronises all the actions according to an implicit global clock, so only one action can occur at each clock tick.

A more sophisticated representation of time, as in Temporal CCS (TCCS) [MT89], is to enable an agent to witness specified lengths of time in addition to witnessing actions. In TCCS actions are still assumed to be instantaneous while the time domain is taken to be the natural numbers.

Timed extensions of other process algebras like ACP [BB91], CSP [Sch95] and LOTOS [BLT94, LL97] have also been defined in the past decades. A main distinction between the several timed process algebras, as pointed out by [HHK02], concerns the interpretation of when actions can occur. In a *must-timing* interpretation, which is usually applied to internal actions, an action must occur as soon as it is enabled since it is not subject to synchronisation and thus it is unnecessary to delay it after it becomes enabled. In a *may-timing* semantics, however, an action may occur after a certain time delay, but may be subject to a further delay, for instance, since it has to synchronise with its context [HHK02]. Usually, the operational semantics of these languages are associated with both action transitions and time transitions, whether they are combined or are treated separately. An overview of the main issues in defining timed process algebra is presented in [NS92].

A.3 Probabilistic process algebra

In some systems there is uncertainty about the behaviour of a component and this uncertainty will be abstracted away so that all choices become nondeterministic. Probabilistic extensions of process algebra, as pointed out in [HHK02], are introduced to quantify the uncertainty of behaviour by replacing nondeterministic choice of actions with probabilistic choice.

The basic idea of probabilistic process algebras is to incorporate the probabilities into the labelled transition systems so that the transitions are associated with probabilities. There are two types of these systems. In the *generative* system a probability distribution is defined over the possible actions of the agent, while in the *reactive* system, the probability distribution is defined over the possible derivatives of an agent given that a particular action is performed [Hil96].

In the CCS tradition, several probabilistic extensions of CCS such as PCCS [JS90] and WSCCS [Tof92] have been proposed. It has been shown in [LS91] that probabilistic process algebras are more suitable tools to test equivalence between a system's specification and its implementation. For an overview of probabilistic process algebras we refer to [Han94].

A.4 Stochastic process algebra

Process algebra will often be used to model systems of which the behaviour with respect to dynamic properties such as throughput and response time are also of interest. Without quantified information about the timing characteristics of the system and the relative probabilities of alternative behaviours, it is not possible to derive those quantitative measures. Such problems encountered when carrying out performance analysis motivate the development of stochastic process algebras.

Stochastic process algebras extend classical process algebras by associating a random variable, representing duration, with every action. In addition to PEPA which has already been introduced in Chapter 2, there are several other stochastic process algebras, for example, TIPP (Timed Process and Performance Analysis) [GHR92], EMPA (Extended Markovian Process Algebra) [BG98] that have been proposed. In particular, the stochastic process algebra IMC (Interactive Markov Chains, [Her98]) takes a different approach in which time and action are separated. Therefore, uncertainty about the behaviour of a component could have more sense than only an embodiment of stochastic distribution. Moreover, a synchronised action in IMC

does not occur until all the engaged components are available, i.e., all the corresponding time delay has finished, which is in contrast to the semantics of PEPA.

Recently, in order to handle some features of biochemical networks, such as stoichiometry and different kinds of kinetic laws, the PEPA formalism has been extended into Bio-PEPA, a language for the modelling and the analysis of biochemical networks. For more details, please refer to [CH09]. There are other process algebras which have been considered in the context of biological systems. For example, the process algebra π-calculus and its biochemical stochastic extension [PRSS01], have been extensively used in systems biology [LPQ$^+$03, KN06] and have given rise to Beta Binders [PQ05]. The calculus Beta Binders is inspired by biological phenomena and enriches the standard π-calculus by allowing the modeller to represent biological features, such as the join between two bio-processes, and the split of one bio-process into two. In order to deal with spatial aspects of biological systems, π-calculus has been extended into SpacePI [Joh08], in which positions are associated with processes and processes can move autonomously according to a movement function.

Another language used for modelling biological systems and their evolution is the Calculus of Looping Sequences (CLS) [BMSMT06b, BMSMT06a, Mil07]. This calculus is based on term rewriting, which describes the biological system, and a set of rewrite rules, modelling the activities one would like to describe. A stochastic version of CLS is proposed in [BMSM$^+$08], in which rates are associated with rewrite rules in order to model the speed of the activities.

Appendix B
Two Proofs in Chapter 3

This appendix presents the proofs omitted in the chapter.

B.1 Proof of consistency between transition rate function and PEPA semantics

Now we show the proof of Proposition 3.4.1 (on page 64), i.e., the transition rate function in Definition 3.4.2 is consistent with the operational semantic rules of PEPA. The semantics have been presented in Section 2.3.3 in Chapter 2. For convenience, we just show a proof of the consistency between the definition of the transition rate function and the "cooperation" operational rule of PEPA, while omitting the proof for other operational rules.

By induction, it is sufficient to consider a simple case like the following

$$X = U_1[\mathbf{x}[U_1]] \underset{L}{\bowtie} U_2[\mathbf{x}[U_2]]$$

and $l \in L$ with $\text{pre}(l) = \{U_1, U_2\}$, where U_i is a local derivative, $\mathbf{x}[U_i] \geq 1$ is the number of instances of some components in U_i, $i = 1, 2$. Let $l \in L$, $w = (U_1 \to V_1, U_2 \to V_2)$ where $V_i \in \text{post}(U_i, l)$ and

$$U_i \overset{(l, r^{U_i \to V_i})}{\longrightarrow} V_i \quad i = 1, 2.$$

Then $X \overset{(l, f(\mathbf{x}, l^w))}{\longrightarrow} X'$ where

$$X' = (U_1[\mathbf{x}[U_1] - 1] \parallel V_1) \underset{L}{\bowtie} (U_2[\mathbf{x}[U_2] - 1] \parallel V_2).$$

According to the definition of transition rate function (Definition 3.4.2), the rate of the transition from X to X' is

$$f(\mathbf{x}, l^w) = \frac{r^{U_1 \to V_1}}{r_l(U_1)} \frac{r^{U_2 \to V_2}}{r_l(U_2)} \min\left\{r_l(U_1)\mathbf{x}[U_1], r_l(U_2)x[U_2]\right\},$$

where

$$r_l(U_i) = \sum_{v \in \text{post}(U_i,l)} r^{U_i \to V}, \quad i = 1, 2.$$

In the following, we show that the rate, namely R, of the transition from X to X' determined by the operational semantics of PEPA, is the same as the above transition rate function $f(\mathbf{x}, l^w)$. In fact, the rate of the transition from $U_1[\mathbf{x}[U_1]]$ to $(U_1[\mathbf{x}[U_1] - 1] \parallel V_1)$ is $r^{U_1 \to V_1}\mathbf{x}[U_1]$, and from $U_2[\mathbf{x}[U_2]]$ to $(U_2[\mathbf{x}[U_2] - 1] \parallel V_2)$ is $r^{U_2 \to V_2}\mathbf{x}[U_2]$. That is,

$$U_1[\mathbf{x}[U_1]] \xrightarrow{(l,r_1)} (U_1[\mathbf{x}[U_1] - 1] \parallel V_1), \qquad r_1 = r^{U_1 \to V_1}\mathbf{x}[U_1],$$

$$U_2[\mathbf{x}[U_2]] \xrightarrow{(l,r_2)} (U_2[\mathbf{x}[U_2] - 1] \parallel V_2), \qquad r_2 = r^{U_2 \to V_2}\mathbf{x}[U_2].$$

According to the operational semantics of PEPA, the rate R in the following transition,

$$\frac{U_1[\mathbf{x}[U_1]] \xrightarrow{(l,r_1)} (U_1[\mathbf{x}[U_1] - 1] \parallel V_1) \quad U_2[\mathbf{x}[U_2]] \xrightarrow{(l,r_2)} (U_2[\mathbf{x}[U_2] - 1] \parallel V_2)}{P \xrightarrow{(l,R)} P'} \ (l \in L_1)$$

$$= \frac{U_1[\mathbf{x}[U_1]] \xrightarrow{(l,r_1)} (U_1[\mathbf{x}[U_1] - 1] \parallel V_1) \quad U_2[\mathbf{x}[U_2]] \xrightarrow{(l,r_2)} (U_2[\mathbf{x}[U_2] - 1] \parallel V_2)}{\Big(U_1[x[U_1]] \underset{L_1}{\bowtie} U_2[\mathbf{x}[U_2]]\Big) \xrightarrow{(l,R)} \Big((U_1[\mathbf{x}[U_1] - 1] \parallel V_1) \underset{L_1}{\bowtie} (U_2[\mathbf{x}[U_2] - 1] \parallel V_2)\Big)} \ (l \in L_1),$$

is calculated as

$$\begin{aligned} R &= \frac{r_1}{r_l(U_1[\mathbf{x}[U_1]])} \frac{r_2}{r_l(U_2[\mathbf{x}[U_2]])} \min\{r_l(U_1[\mathbf{x}[U_1]]), r_l(U_2[\mathbf{x}[U_2]])\} \\ &= \frac{r^{U_1 \to V_1}\mathbf{x}[U_1]}{r_l(U_1)\mathbf{x}[U_1]} \frac{r^{U_2 \to V_2}\mathbf{x}[U_2]}{r_l(U_2)\mathbf{x}[U_2]} \min\{r_l(U_1)\mathbf{x}[U_1], r_l(U_2)\mathbf{x}[U_2]\} \\ &= \frac{r^{U_1 \to V_1}}{r_l(U_1)} \frac{r^{U_2 \to V_2}}{r_l(U_2)} \min\{r_l(U_1)\mathbf{x}[U_1], r_l(U_2)\mathbf{x}[U_2]\}, \end{aligned}$$

where $r_l(U_i[\mathbf{x}[U_i]]) = r_l(U_i)\mathbf{x}[U_i]$ is the apparent rate of l in the process $U_i[\mathbf{x}[U_i]]$ (see the apparent rate definition in Chapter 2).

Compare R with $f(\mathbf{x}, l^w)$, we have $R = f(\mathbf{x}, l^w)$. The proof is complete.

B.2 Proof of Proposition 3.4.3

Lemma B.2.1. *The function "$\min(\cdot)$" is Lipschitz continuous.*

Proof. Consider the two-dimensional case first, we need to prove

$$|\min(x_1, x_2) - \min(y_1, y_2)| \leq K\|(x_1, x_2) - (y_1, y_2)\| \stackrel{def}{=} K(|x_1 - y_1| + |x_2 - y_2|),$$

where K is some constant.

Noticing $\min(a, b) = \frac{a+b-|a-b|}{2}$, then

$$\begin{aligned}
\min(x_1, x_2) - \min(y_1, x_2) &= \frac{x_1 - y_1}{2} + \frac{|y_1 - x_2| - |x_1 - x_2|}{2} \\
&\leq \frac{1}{2}|x_1 - y_1| + \frac{1}{2}|(y_1 - x_2) - (x_1 - x_2)| \\
&= |x_1 - y_1|.
\end{aligned}$$

Similarly, $\min(y_1, x_2) - \min(y_1, y_2) \leq |x_2 - y_2|$. Hence

$$\begin{aligned}
|\min(x_1, x_2) - \min(y_1, y_2)| &= |\min(x_1, x_2) - \min(y_1, x_2) + \min(y_1, x_2) - \min(y_1, y_2)| \\
&\leq |\min(x_1, x_2) - \min(y_1, x_2)| + |\min(y_1, x_2) - \min(y_1, y_2)| \\
&\leq |x_1 - y_1| + |x_2 - y_2|.
\end{aligned}$$

For a general n-dimensional case, by induction we have

$$\begin{aligned}
\left|\min_{i\in\{1,\cdots,n\}}\{x_i\} - \min_{i\in\{1,\cdots,n\}}\{y_i\}\right| &= \left|\min\left(x_1, \min_{i\in\{2,\cdots,n\}}\{x_i\}\right) - \min\left(y_1, \min_{i\in\{2,\cdots,n\}}\{y_i\}\right)\right| \\
&\leq |x_1 - y_1| + \left|\min_{i\in\{2,\cdots,n\}}\{x_i\} - \min_{i\in\{2,\cdots,n\}}\{y_i\}\right| \\
&\leq \cdots\cdots \\
&\leq \sum_{i=1}^{n-2}|x_i - y_i| + \left|\min_{i=n-1,n}\{x_i\} - \min_{i=n-1,n}\{y_i\}\right| \\
&\leq \sum_{i=1}^{n}|x_i - y_i|.
\end{aligned}$$

The proof is complete. □

For convenience, we list Proposition 3.4.3 again.

Proposition 3.4.3. *Let l be an labelled activity, and $\mathbf{x}, \mathbf{y}$ be two states. The transition rate function $f(\mathbf{x}, l)$, defined in Definition 3.4.2 satisfies:*

1. *For any $H > 0$, $Hf(\mathbf{x}/H, l) = f(\mathbf{x}, l)$.*

2. *There exists* $M > 0$ *such that* $|f(\mathbf{x}, l) - f(\mathbf{y}, l)| \leq M||\mathbf{x} - \mathbf{y}||$ *for any* $\mathbf{x}, \mathbf{y}$ *and* l.

Proof of Proposition 3.4.3. If l is individual, the proof is trivial. Suppose l is shared.

1. Notice for any $H > 0$,

$$\min_{i=1,\cdots,n}\{a_i x_i\} = H \min_{i=1,\cdots,n}\{a_i x_i / H\},$$

so $f(\mathbf{x}, l) = Hf(\mathbf{x}/H, l)$.

2. Consider (B.2) in Definition 3.4.2, that is

$$f(\mathbf{x}, l^w) = \left(\prod_{i=1}^{k} \frac{r_l^{U_i \to V_i}}{r_l(U_i)}\right) \min_{i\in\{1,\cdots,k\}}\{\mathbf{x}[U_i] r_l(U_i)\}.$$

Since for each i, $0 < r_l^{U_i \to V_i} \leq r_l(U_i)$, so $\prod_{i=1}^{k} \frac{r_l^{U_i \to V_i}}{r_l(U_i)} \leq 1$. Let $M = \max_{U\in\mathcal{D}}\{r_l(U)\}$. Then for any l, by Lemma B.2.1,

$$\begin{aligned}
|f(\mathbf{x}, l) - f(\mathbf{y}, l)| &= \left(\prod_{i=1}^{k} \frac{r_l^{U_i \to V_i}}{r_l(U_i)}\right) \left| \min_{i\in\{1,\cdots,k\}}\{\mathbf{x}[U_i] r_l(U_i)\} - \min_{i\in\{1,\cdots,k\}}\{\mathbf{y}[U_i] r_l(U_i)\} \right| \\
&\leq \left| \min_{i\in\{1,\cdots,k\}}\{\mathbf{x}[U_i] r_l(U_i)\} - \min_{i\in\{1,\cdots,k\}}\{\mathbf{y}[U_i] r_l(U_i)\} \right| \\
&\leq \sum_{i=1}^{k} |\mathbf{x}[U_i] r_l(U_i) - \mathbf{y}[U_i] r_l(U_i)| \\
&= \sum_{i=1}^{k} r_l(U_i) \, |\mathbf{x}[U_i] - \mathbf{y}[U_i]| \\
&\leq M \sum_{U\in\mathcal{D}} |\mathbf{x}[U] - \mathbf{y}[U]| \\
&= M\|\mathbf{x} - \mathbf{y}\|.
\end{aligned}$$

This completes the proof.

Appendix C

Some Theorems and Functional Analysis of Markov chains

C.1 Some theorems

Theorem C.1.1. *([HW90], page 14). If* $\mathbf{x}' = \mathbf{f}(t, \mathbf{x})$ *is defined on a set* U *in* $\mathbb{R} \times \mathbb{R}^n$ *with Lipschitz condition*

$$\|\mathbf{f}(t, \mathbf{x}_1) - \mathbf{f}(t, \mathbf{x}_2)\| < K\|\mathbf{x}_1 - \mathbf{x}_2\|$$

for all $(t, \mathbf{x}_1)$ *and* $(t, \mathbf{x}_2)$ *on* U*, then there exists a unique solution* $\mathbf{x} = \mathbf{u}(t)$ *for a given set of initial condition* $\mathbf{x}(t_0)$*.*

Theorem C.1.2. *(**Lebesgue's Theorem on Dominated Convergence**, [Shi96], page 187). Let* $\eta, \xi, \xi_1, \xi_2, \ldots$ *be random variables such that* $|\xi_n| \leq \eta$*,* $E\eta \leq \infty$ *and* $\xi_n \to \xi$ *(a.s.). Then* $E|\eta| \leq \infty$*,* $E\xi_n \to E\xi$ *and* $E|\xi_n - \xi| \to 0$ *as* $n \to \infty$*.*

Theorem C.1.3. *(**Kurtz theorem** [EK86], page 456). Let* X_n *be a family of density dependent CTMCs with the infinitesimal generators*

$$q^{(n)}_{k,k+l} = nf(k/n, l),$$

where $f(x, l)$ $(x \in E \subset R^h,\ l \in \mathbb{Z}^h)$ *is a continuous function,* k *is a numerical state vector and* l *is a transition vector.*

Suppose $X(t) \in E$ *satisfies*

$$\frac{\mathrm{d}x}{\mathrm{d}t} = F(x)$$

where $F(x) = \sum_l lf(x, l)$*. Suppose that for each compact* $K \subset E$*,*

$$\sum_l \|l\| \sup_{x \in K} f(x, l) < \infty \tag{C.1}$$

and there exists $M_K > 0$ *such that*

$$\|F(x) - F(y)\| \leq M_K \|x - y\|, \quad x, y \in K. \tag{C.2}$$

If $\lim_{n\to\infty} \frac{X_n(0)}{n} = x_0$*, then for every* $t \geq 0$*,*

$$\lim_{n\to\infty} \sup_{s\leq t} \left\| \frac{X_n(s)}{n} - X(s) \right\| = 0 \quad a.s. \tag{C.3}$$

C.2 Spectral gaps and Log-Sobolev constants of Markov chains

This section introduces the spectral gap and Log-Sobolev constant of a Markov chain. The material presented here is extracted from [SC97].

Let (K, π) be a Markov chain on a finite set S, where K is a Markov kernel and π is the stationary probability distribution associated with K. For any real function f, g on S, define an inner product "$\langle \cdot, \cdot \rangle$" as

$$\langle f, g \rangle = \sum_{x\in S} f(x)g(x)\pi(x).$$

Denote $\|f\|_2 = \sqrt{\langle f, f \rangle}$, and

$$l^2(\pi) = \{f : \|f\|_2 < \infty\}.$$

Then $l^2(\pi)$ is a Hilbert space with the norm $\|\cdot\|_2$. We say that K^* is *adjoint* to K if

$$\langle Kf, g \rangle = \langle f, K^* g \rangle, \quad \forall f, g \in L^2(\pi).$$

It follows that

$$K^*(x, y) = \frac{\pi(y)}{\pi(x)} K(y, x).$$

If $K = K^*$, then K is called *self-adjoint*. If K is self-adjoint on $l^2(\pi)$, then (K, π) is *reversible* (note: this is different from the reversible definition given in Definition 4.4.4).

For a function in $f \in l^2(\pi)$, denote its mean and variance by $\pi(f)$ and $Var(f)$ respectively, that is

$$\pi(f) = \sum_{x\in S} f(x)\pi(x), \quad Var(f) = \pi((f - \pi(f))^2).$$

Definition C.2.1. *(**Dirichlet form**). The form*

$$\mathcal{E}(f,g) = \langle (I-K)f, g\rangle$$

is called the Dirichlet form associated with $H_t = e^{-t(I-K)}$.

Remark C.2.1. *The Dirichlet form $\mathcal{E}$ satisfies*

$$\mathcal{E}(f,f) = \langle (I-K)f, f\rangle = \left\langle \left(I - \frac{K+K^*}{2}\right) f, f \right\rangle,$$

$$\mathcal{E}(f,f) = \frac{1}{2}\sum_{x,y}(f(x)-f(y))^2 K(x,y)\pi(x).$$

Definition C.2.2. *(**Spectral gap**). Let K be a Markov kernel with Dirichlet form $\mathcal{E}$. The spectral gap $\lambda = \lambda(K)$ is defined by*

$$\lambda = \min\left\{\frac{\mathcal{E}(f,f)}{Var(f)} : Var(f) \neq 0\right\}.$$

Remark C.2.2. *In general λ is the smallest non zero eigenvalue of $I - \frac{K+K^*}{2}$. If K is self-adjoint, then λ is the smallest non zero eigenvalue of $I-K$. Clearly, we also have*

$$\lambda = \min\left\{\mathcal{E}(f,f) : \|f\|_2 = 1, \pi(f) = 0\right\}.$$

The definition of the logarithmic Sobolev (Log-Sobolev) constant α is similar to that of the spectral gap λ where the variance has been replaced by

$$\mathcal{L}(f) = \sum_{x\in S} f(x)^2 \log\left(\frac{f(x)^2}{\|f\|_2^2}\right)\pi(x).$$

Definition C.2.3. *(**Log-Sobolev constant**). Let K be an irreducible Markov chain with stationary measure π. The logarithmic Sobolev constant $\alpha = \alpha(K)$ is defined by*

$$\alpha = \min\left\{\frac{\mathcal{E}(f,f)}{\mathcal{L}(f)} : \mathcal{L}(f) \neq 0\right\}.$$

Lemma C.2.1. *. For any finite Markov chain K with stationary measure π, the Log-Sobolev constant α and the spectral gap λ satisfy*

$$\frac{1-2\pi(*)}{\log[1/\pi(*)-1]}\lambda \leq \alpha \leq \frac{\lambda}{2},$$

where $\pi(*) = \min_x \pi(x)$.

A Markov chain on S determines a directed graph (S, E). There is an edge between vertices x and y if and only if $K(x, y) > 0$. E is the set of all edges on S.

Definition C.2.4. ***(Adapted set)*** *Let K be an irreducible Markov chain on a finite set S. An edge set $\mathcal{A} \subset \mathcal{S} \times \mathcal{S}$ is said to be* adapted *to K if $\mathcal{A}$ is symmetric (that is $(x, y) \in \mathcal{A} \Rightarrow (y, x) \in \mathcal{A}$), $(S, \mathcal{A})$ is connected, and*

$$(x, y) \in \mathcal{A} \Rightarrow K(x, y) + K(y, x) > 0.$$

In this case we also say that the graph $(S, \mathcal{A})$ is adapted.

Let $\mathcal{A}$ be an adapted edge set. A path γ in $(S, \mathcal{A})$ is a sequence of vertices $\gamma = (x_0, \cdots, x_k)$ such that $(x_{i-1}, x_i) \in \mathcal{A}, i = 1, 2, \cdots, k$. Equivalently, γ can be viewed as a sequence of edges $\gamma = (e_1, \cdots, e_k)$ with $e_k = (x_{i-1}, x_i) \in \mathcal{A}, i = 1, 2, \cdots, k$. The length of such path is $|\gamma| = k$. Let Γ be the set of all paths γ in $(S, \mathcal{A})$ which have no repeated edges. For each $e = (x, y) \in S \times S$, set

$$\Gamma(x, y) = \{\gamma = (x_0, \cdots, x_k) \in \Gamma : x = x_0, y = x_k\},$$

$$J(e) = \frac{K(x, y)\pi(x) + K(y, x)\pi(y)}{2}.$$

Theorem C.2.2. *Let K be an irreducible chain with stationary measure π on a finite set S. Let $\mathcal{A}$ be an adapted edge set. For each $(x, y) \in S \times S$ choose exactly one path $\gamma(x, y)$ in $\Gamma(x, y)$. Then $\lambda \geq \frac{1}{B}$ where*

$$B = \max_{e \in \mathcal{A}} \left\{ \frac{1}{J(e)} \sum_{x,y \in S : \gamma(x,y) \ni e} |\gamma(x, y)| \pi(x)\pi(y) \right\}.$$

The *boundary* ∂A of a set $A \subset S$ is the set

$$\partial A = \{e = (x, y) \in S \times S : x \in A, y \in A^c \text{ or } x \in A^c, y \in A\}.$$

Thus, the boundary is the set of all pairs connecting A and A^c. Define the measure of the

boundary ∂A of $A \subset S$ by

$$J(\partial A) = \frac{1}{2} \sum_{x \in A, y \in A^c} [K(x,y)\pi(x) + K(y,x)\pi(y)].$$

Definition C.2.5. *(**Isoperimetric constant**). The isoperimetric constant of the chain (K, π) is defined by*

$$I = I(K, \pi) = \sum_{A \subset S : \pi(A) \leq 1/2} \left\{ \frac{J(\partial A)}{\pi(A)} \right\}.$$

Theorem C.2.3. *(**Cheeger's inequality**). The spectral gap λ and the isoperimetric constant I is related by*

$$\frac{I^2}{8} \leq \lambda \leq I.$$

Appendix D
Proofs and Some Background Theories in Chapter 6

D.1 Some basic results in mathematical analysis

The following lemma can be found in any good book on differential calculus.

Lemma D.1.1. *Let $y(t)$ be a differentiable function defined for $t \geq 0$. Suppose $a, b \in \mathbb{R}$, $a \neq 0$. If $y(t)$ satisfies $\dfrac{\mathrm{d}y}{\mathrm{d}t} \geq ay(t) + b,\ t > 0$, then*

$$y(t) \geq e^{at}\left(y(0) + \frac{b}{a}\right) - \frac{b}{a}.$$

Similarly, if $y(t)$ satisfies $\dfrac{\mathrm{d}y}{\mathrm{d}t} \leq ay(t) + b,\ t > 0$, then

$$y(t) \leq e^{at}\left(y(0) + \frac{b}{a}\right) - \frac{b}{a}.$$

Proof. Let $W(t) = y(t)e^{-at}$, then $\dfrac{\mathrm{d}W}{\mathrm{d}t} = e^{-at}\left(\dfrac{\mathrm{d}y}{\mathrm{d}t} - ay\right) \geq be^{-at}$. Integrating on both sides, so $W(t) - W(0) \geq b\int_0^t e^{-as}ds$. Thus $y(t)e^{-at} - y(0) \geq \frac{b}{a}(1 - e^{-at})$. So $y(t) \geq e^{at}\left(y(0) + \frac{b}{a}\right) - \frac{b}{a}$. The second conclusion can be similarly proved. □

Theorem D.1.2. *(**Fundamental Inequality**, [HW90], page 14). If $\dfrac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}, t)$ is defined on a set U in $\mathbb{R}^n \times \mathbb{R}$ with the Lipschitz condition*

$$\|\mathbf{f}(\mathbf{x}_1, t) - \mathbf{f}(\mathbf{x}_2, t)\| < K\|\mathbf{x}_1 - \mathbf{x}_2\|$$

for all $(\mathbf{x}_1, t)$ and $(\mathbf{x}_2, t)$ on U, and if for $\epsilon_i, \delta \in \mathbb{R}$, and $\mathbf{u}_1(t)$ and $\mathbf{u}_2(t)$ are two continuous, piecewise differentiable functions on U into $\mathbb{R}^n$ with

$$\left\|\frac{\mathrm{d}\mathbf{u}_i(t)}{\mathrm{d}t} - \mathbf{f}(\mathbf{u}_i(t), t)\right\| \leq \epsilon_i, \quad \textit{and} \quad \|\mathbf{u}_1(t_0) - \mathbf{u}_2(t_0)\| \leq \delta,$$

then

$$\|\mathbf{u}_1(t) - \mathbf{u}_2(t)\| \leq \delta e^{K(t-t_0)} + \left(\frac{\epsilon_1 + \epsilon_2}{K}\right)\left(e^{K(t-t_0)} - 1\right).$$

The following theorem is well-known, and can be found in standard calculus books.

Theorem D.1.3. ***(Properties of continuous functions)****. For a continuous function $f(x)$ defined on a compact set I (e.g. closed intervals), it has the following properties:*

1. *$f(x)$ is bounded on I, that is, there exists $m, M \in \mathbb{R}$ such that $m \leq f(x) \leq M$ for any $x \in I$.*

2. *$f(x)$ achieves its minimum and maximum on I, i.e. there exists $x_1, x_2 \in I$ such that $f(x_1) = \inf_{x\in I} f(x)$ and $f(x_2) = \sup_{x\in I} f(x)$.*

D.2 Some theories of differential equations

D.2.1 The Jordan Canonical Form

In this subsection, we use $\Re(z)$ and $\Im(z)$ to respectively represent the real and imaginary parts of a complex number z. The following is mainly extracted from [Per91] (page 39∼42).

Theorem D.2.1. ***(The Jordan Canonical Form)****. Let A be a real matrix with real eigenvalues λ_j, $j = 1, \cdots, k$ and complex eigenvalues $\lambda_j = a_j + ib_j$ and $\bar{\lambda}_j = a_j - ib_j$, $j = k+1, \cdots, n$. Then there exists a basis $\{\mathbf{v}_1, \cdots, \mathbf{v}_k, \mathbf{v}_{k+1}, \mathbf{u}_{k+1}, \cdots, \mathbf{v}_n, \mathbf{u}_n\}$ for $\mathbb{R}^{2n-k}$ where $\mathbf{v}_j$, $j = 1, \cdots, k$ and $\mathbf{w}_j$, $j = k+1, \cdots, n$ are generalized eigenvectors of A, $\mathbf{u}_j = \Re(\mathbf{w}_j)$ and $\mathbf{v}_j = \Im(\mathbf{w}_j)$ for $j = k+1, \cdots, n$, such that the matrix $P = \{\mathbf{v}_1, \cdots, \mathbf{v}_k, \mathbf{v}_{k+1}, \mathbf{u}_{k+1}, \cdots, \mathbf{v}_n, \mathbf{u}_n\}$ is invertible and*

$$P^{-1}AP = \begin{bmatrix} B_1 & & \\ & \ddots & \\ & & B_r \end{bmatrix}$$

where the elementary Jordan blocks $B = B_j$, $j = 1, \cdots, r$ are either of the form

$$B = \begin{bmatrix} \lambda & 1 & 0 & \cdots & 0 \\ 0 & \lambda & 1 & \cdots & 0 \\ \cdots & & & & \\ 0 & \cdots & & \lambda & 1 \\ 0 & \cdots & & 0 & \lambda \end{bmatrix} \tag{D.1}$$

for λ one of the real eigenvalues of A or of the form

$$B = \begin{bmatrix} D & I_2 & 0 & \cdots & 0 \\ 0 & D & I_2 & \cdots & 0 \\ \cdots & & & & \\ 0 & \cdots & & D & I_2 \\ 0 & \cdots & & 0 & D \end{bmatrix} \tag{D.2}$$

with

$$D = \begin{bmatrix} a & -b \\ b & a \end{bmatrix}, \quad I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \textit{and} \quad 0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

for $\lambda = a + ib$ one of the complex eigenvalues of A.

The Jordan canonical form of A yields some explicit information about the form of $\mathbf{x} = e^{At}\mathbf{x}_0$, i.e. the solution of the initial value problem

$$\begin{cases} \dfrac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = A\mathbf{x} \\ \mathbf{x}(0) = \mathbf{x}_0 \end{cases} \tag{D.3}$$

That is,

$$\mathbf{x}(t) = P \,\mathrm{diag}\left[e^{B_j t}\right] P^{-1}\mathbf{x}_0, \tag{D.4}$$

where B_j are the elementary Jordan blocks of A, $j = 1, \cdots, r$. Here $\mathrm{diag}\left[e^{B_j t}\right]$ represents

$$\mathrm{diag}\left[e^{B_j t}\right] = \begin{pmatrix} e^{B_1 t} & 0 & \cdots & 0 \\ 0 & e^{B_2 t} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{B_r t} \end{pmatrix}.$$

In the following, the notation $\mathrm{diag}[\cdot]$ indicates the similar meaning. If $B_j = B$ is an $m \times m$ matrix of the form (D.1) and λ is a real eigenvalue of A then

$$e^{Bt} = e^{\lambda t} \begin{bmatrix} 1 & t & t^2/2! & \cdots & t^{m-1}/(m-1)! \\ 0 & 1 & t & \cdots & t^{m-2}/(m-2)! \\ 0 & 0 & 1 & \cdots & t^{m-3}/(m-3)! \\ \cdots & & & & \\ 0 & \cdots & & 1 & t \\ 0 & \cdots & & 0 & 1 \end{bmatrix}. \tag{D.5}$$

If $B_j = B$ is an $2m \times 2m$ matrix of the form (D.2) and $\lambda = a + ib$ is a complex eigenvalue of A, then

$$e^{Bt} = e^{at} \begin{bmatrix} R & Rt & Rt^2/2! & \cdots & Rt^{m-1}/(m-1)! \\ 0 & R & Rt & \cdots & Rt^{m-2}/(m-2)! \\ 0 & 0 & R & \cdots & Rt^{m-3}/(m-3)! \\ \cdots & & & & \\ 0 & \cdots & & R & Rt \\ 0 & \cdots & & 0 & R \end{bmatrix} \tag{D.6}$$

where R is the rotation matrix

$$R = \begin{bmatrix} \cos bt & -\sin bt \\ \sin bt & \cos bt \end{bmatrix}.$$

Theorem D.2.2. *If $\mathbf{x}(t)$ satisfies (D.3), then each coordinate in $\mathbf{x}(t)$ is a linear combination of functions of the form*

$$t^k e^{at} \cos bt \quad \textit{or} \quad t^k e^{at} \sin bt$$

where $\lambda = a + ib$ is an eigenvalue of the matrix $A_{n\times n}$ and $0 \le k \le n-1$.

Corollary D.2.3. *If the eigenvalues of A are either zeros or have negative real parts, and $\mathbf{x}(t)$ is bounded in $[0, \infty)$, then $\mathbf{x}(t)$ converges to a finite limit as time goes to infinity.*

Proof. The solution is composed of the terms like $t^k e^{at} \cos bt$ and $t^k e^{at} \sin bt$. If $a < 0$, then $t^k e^{at} \cos bt$ and $t^k e^{at} \sin bt$ converge as time goes to infinity. If $a = b = 0$, we will see $k = 0$. In fact, in this case $t^k e^{at} \cos bt = t^k$. If $k > 0$, then this term t^k in the solution will make the solution unbounded as t tends to infinity. So k must be zero in the terms corresponding to

$a = b = 0$. Thus, $t^k e^{at} \cos bt = 1$ and $t^k e^{at} \sin bt = 0$. So the solution converges. □

D.2.2 Some obtained results

As the above subsections illustrate, the following problem

$$\begin{cases} \dfrac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = A\mathbf{x} \\ \mathbf{x}(0) = \mathbf{x}_0 \end{cases}$$

has solution $\mathbf{x}(t) = e^{At}\mathbf{x}_0$, which equals

$$\mathbf{x}(t) = P \,\mathrm{diag}\left[e^{B_j t}\right] P^{-1}\mathbf{x}_0, \tag{D.7}$$

where B_j are the elementary Jordan blocks of A, $j = 1, \cdots, r$. Suppose the rank of $A_{n\times n}$ is $n-1$. This means that zero is a one fold eigenvalue of A.

According to (D.7), we construct a corresponding $\hat{\mathbf{x}}$ in the form

$$\hat{\mathbf{x}} = P \,\mathrm{diag}\left[\hat{B}_j\right] P^{-1}\mathbf{x}_0, \tag{D.8}$$

where $\hat{B}_j(t)$ is defined as follows. If $e^{B_j t}$ in (D.7) has the form of (D.6), then $\hat{B}_j$ is defined by

$$\hat{B}_j = \mathbf{0}_{2m\times 2m}. \tag{D.9}$$

If $e^{B_j t}$ has the form of (D.5), and the corresponding real eigenvalue $\lambda < 0$, then $\hat{B}_j$ is defined by

$$\hat{B}_j = \mathbf{0}_{m\times m} \tag{D.10}$$

If $\lambda = 0$, we know that zero is a one fold eigenvalue of A due to its rank $n-1$. So $m = 1$. Then,

$$\hat{B}_j = 1. \tag{D.11}$$

In short, only for the zero eigenvalue is $\hat{B}_j$ set to one, otherwise it is set to zeros. The readers are suggested to see the discussions in Section 6.4.2 for instance. Obviously, we have

Lemma D.2.4. *If zero is a one fold eigenvalue of A and all other eigenvalues of A have negative*

real parts, then

$$\lim_{t\to\infty} |\hat{\mathbf{x}}(t) - \mathbf{x}(t)| = 0.$$

Proof. If B is the Jordan block corresponding to the one fold zero eigenvalue, then according to (D.11), $e^{Bt} = e^0 = 1$, and $e^{Bt} - \hat{B} = 1 - 1 = 0$. For any non-zero eigenvalue, since $\hat{B} = 0$ then $e^{Bt} - \hat{B} = e^{Bt}$. Notice that by (D.5) and (D.6),

$$\|e^{Bt}\| \leq C_1(t)e^{-\Lambda t},$$

where $C_1(t)$ is a polynomial of t with the maximum order k, and

$$\Lambda = \inf\{-\Re(\lambda) \mid \lambda \text{ is non-zero eigenvalue of } A\} > 0.$$

Therefore,

$$\|\mathbf{x}(t) - \hat{\mathbf{x}}\| \leq \sum_B \|e^{Bt} - \hat{B}\| \leq C_2(t)e^{-\Lambda t} \longrightarrow 0 \tag{D.12}$$

as t goes to infinity, where $C_2(t)$ is a polynomial function of t. □

The above construction can be extended to the problem of

$$\frac{\mathrm{d}X}{\mathrm{d}t} = A(t)X \tag{D.13}$$

with the initial value $X(0)$, which is discussed in Section 6.5.3. The solution is

$$X(t) = e^{tC(t)}X(0),$$

where $C(t) = \dfrac{1}{t}\displaystyle\int_0^t A(s)ds$. We can similarly define a function f such that $C(t) = f(\beta(t))$, where $\beta(t)$ is similarly defined according to $A(t)$. Therefore,

$$X(t) = e^{tC(t)}X(0) = e^{tf(\beta(t))}X(0).$$

For a fixed β,

$$e^{tf(\beta)}X(0) = P(\beta)\,\mathrm{diag}\left[e^{B_j(\beta)t}\right]P(\beta)^{-1}X(0), \tag{D.14}$$

where $B_j(\beta)$ are the elementary Jordan blocks of $f(\beta)$, $j = 1, \cdots, r(\beta)$. Repeating the previous construction process with $B_j(\beta)$ for each j, we obtain the constructed matrix $\hat{B}(\beta)_j$. We

define

$$h(\beta) = P(\beta)\,\mathrm{diag}\left[\hat{B}(\beta)_j\right] P(\beta)^{-1} X(0). \tag{D.15}$$

For convenience, suppose the dimension of $A(t)$ in (D.13) is $n \times n$. We should point out that for any t, the rank of $A(t)$ is $n-1$, and thus for any t, $A(t)$'s zero eigenvalue is one fold. In fact, the rank of any infinitesimal generator with dimension $n \times n$ is $n-1$. This implies that any $n-1$ columns or rows of this generator are linearly independent. According to the definition of $A(t)$ in Section 6.5.3, $A(t)$ is an infinitesimal generator if $\alpha(t) \neq 0$. If $\alpha(t) = 0$, $A(t)$ is also a generator after one column is modified (see Proposition 6.5.1 and 6.5.2), which means that the other $n-1$ columns are linearly independent. So whatever t is, the rank of $A(t)$ is $n-1$. Thus, the zero eigenvalue is one fold. Therefore, $f(\beta)$'s zero eigenvalue is also one fold for any β. So each entry of all blocks $\hat{B}(\beta)_j$ is zero, except for the one corresponding to the zero eigenvalue, in which case this block is a scalar one. This implies that for any β all entries of the matrix $\mathrm{diag}[\hat{B}(\beta)_j]$ are zeros, except for a diagonal entry with one.

By permutation, $\mathrm{diag}\left[\hat{B}(\beta)_j\right]$ can always be transformed into the form

$$\begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Correspondingly, $P(\beta)$ is permuted into $U(\beta)$. Therefore, the formulae (D.15) can be written as

$$h(\beta) = U(\beta) \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} U(\beta)^{-1} X(0), \tag{D.16}$$

Now we prove a proposition which is used in Section 6.5.3.

Proposition D.2.1. *For the $e^{tf(\beta)}X(0)$ in (D.14) and $h(\beta)$ in (D.15), we have*

$$\lim_{t\to\infty} \|e^{tf(\beta)}X(0) - h(\beta)\| = 0.$$

Proof. By a similar estimation as (D.12), we have

$$\|e^{tf(\beta)}X(0) - h(\beta)\| \leq C(t)e^{-\Lambda_1 t}.$$

where $C(t)$ is a polynomial of t. By a similar proof to Lemma 6.4.2, we have

$$\Lambda_1 = \inf_{\beta\in[0,1]}\{-\Re(\lambda)|\lambda \text{ is } f(\beta)\text{'s non-zero eigenvalue}\} > 0.$$

Then

$$\|e^{tf(\beta)}X(0) - h(\beta)\| \leq C(t)e^{-\Lambda_1 t} \longrightarrow 0$$

as t goes to infinity. □

Let $\hat{X}(t) = h(\beta(t))$, where $\beta(t) \in [0,1]$, and notice $X(t) = f(\beta(t))$. As a consequence of this proposition, we have

Corollary D.2.5. *Let $X(t)$ be the solution of $\frac{\mathrm{d}X}{\mathrm{d}t} = A(t)X$ which is discussed in Section 6.5.3, and let $\hat{X}(t) = h(\beta(t))$, then*

$$\lim_{t\to\infty}\|X(t) - \hat{X}(t)\| = 0.$$

D.3 Eigenvalue properties of coefficient matrices of Model 3

In this subsection, we claim that all eigenvalues of Q_i $(i = 1,2,3,4)$ appearing in (6.11) in Section 6.3.1 other than zeros have negative real parts.

We do not worry about Q_1 and Q_4 since they are lower or upper block triangular matrices and the eigenvalues of this kind of matrices can be well estimated: all eigenvalues of Q_1 and Q_4 are either zeros or have negative real parts. All that we want to do here is to show that both Q_2 and Q_3 also have this property.

By symbolic calculation using Matlab, Q_3's eigenvalues are

$$\lambda_{1,2,3} = 0(\text{three folds}), \lambda_4 = -c_4 - c_3,$$

$$\lambda_5 = -\frac{1}{2}(a_1 + a_2 + c_1 + c_2) + \frac{1}{2}\sqrt{(a_1 - a_2 + c_1 + c_2)^2 - 4a_1c_2},$$

$$\lambda_6 = -\frac{1}{2}(a_1 + a_2 + c_1 + c_2) - \frac{1}{2}\sqrt{(a_1 - a_2 + c_1 + c_2)^2 - 4a_1c_2}.$$

If $(a_1 - a_2 + c_1 + c_2)^2 - 4a_1c_2 < 0$, then the real parts of λ_5 and λ_6 are $-\frac{1}{2}(a_1 + a_2 + c_1 + c_2)$,

which is negative. Otherwise,

$$(a_1 - a_2 + c_1 + c_2)^2 - 4a_1c_2 \geq 0.$$

In this case,

$$(a_1 - a_2 + c_1 + c_2)^2 - 4a_1c_2 \leq (a_1 - a_2 + c_1 + c_2)^2 < (a_1 + a_2 + c_1 + c_2)^2,$$

so

$$-\frac{1}{2}(a_1 + a_2 + c_1 + c_2) + \frac{1}{2}\sqrt{(a_1 - a_2 + c_1 + c_2)^2 - 4a_1c_2} < 0.$$

This means that λ_5 and λ_6 are both negative. Thus λ_i $(i = 1, 2, \cdots, 6)$ are either 0 or have negative real parts.

Similarly, Q_2's eigenvalues are $\delta_{1,2,3} = 0$, $\delta_4 = -c_1 - c_2$,

$$\delta_5 = -\frac{1}{2}(a_1 + a_2 + c_3 + c_4) + \frac{1}{2}\sqrt{(a_2 - a_1 + c_3 + c_4)^2 - 4a_2c_3},$$

$$\delta_6 = -\frac{1}{2}(a_1 + a_2 + c_3 + c_4) - \frac{1}{2}\sqrt{(a_2 - a_1 + c_3 + c_4)^2 - 4a_2c_3}.$$

By similar argument, we still have that δ_i $(i = 1, 2, \cdots, 6)$ are either zeros or have negative real parts.

D.4 Eigenvalue property for more general cases

In this subsection, we show that the eigenvalue properties hold for a more general case. Suppose in a system, there are three kinds of component X, Y, Z, with one synchronisation between X and Y, and one synchronisation between Y and Z. By a similar discussion in Section 6.4.1, the derived ODEs are as follows,

$$\begin{aligned}\begin{pmatrix} \frac{\mathrm{d}X}{\mathrm{d}t} \\ \frac{\mathrm{d}Y}{\mathrm{d}t} \\ \frac{\mathrm{d}Z}{\mathrm{d}t} \end{pmatrix} =& I_{\{r_1x_i \leq s_1y_j, r_2y_m \leq s_2z_n\}} Q_1 \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + I_{\{r_1x_i \leq s_1y_j, r_2y_m > s_2z_n\}} Q_2 \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \\ &+ I_{\{r_2x_i > s_1y_j, r_2y_m \leq s_2z_n\}} Q_3 \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + I_{\{r_1x_i > s_1y_j, r_2y_m > s_2z_n\}} Q_4 \begin{pmatrix} X \\ Y \\ Z \end{pmatrix},\end{aligned}$$

where

$$Q_1 = \left(\begin{array}{c|c||c} Q_{11}^{(1)} & & \\ \hline Q_{21}^{(1)} & Q_{22}^{(1)} & \\ \hline\hline & Q_{32}^{(1)} & Q_{33}^{(1)} \end{array}\right) = \left(\begin{array}{c||c} D_1 & \\ \hline\hline Q_{32}^{(1)} & Q_{33}^{(1)} \end{array}\right), \quad D_1 = \left(\begin{array}{c|c} Q_{11}^{(1)} & \\ \hline Q_{21}^{(1)} & Q_{22}^{(1)} \end{array}\right),$$

$$Q_2 = \left(\begin{array}{c|c||c} Q_{11}^{(2)} & & \\ \hline Q_{21}^{(2)} & Q_{22}^{(2)} & Q_{23}^{(2)} \\ \hline\hline & & Q_{33}^{(2)} \end{array}\right) = \left(\begin{array}{c||c} D_2 & Q_{23}^{(2)} \\ \hline\hline & Q_{33}^{(2)} \end{array}\right), \quad D_2 = \left(\begin{array}{c|c} Q_{11}^{(2)} & \\ \hline Q_{21}^{(2)} & Q_{22}^{(2)} \end{array}\right),$$

$$Q_3 = \left(\begin{array}{c|c||c} Q_{11}^{(3)} & Q_{12}^{(3)} & \\ \hline & Q_{22}^{(3)} & \\ \hline\hline & Q_{32}^{(3)} & Q_{33}^{(3)} \end{array}\right) = \left(\begin{array}{c||c} D_3 & \\ \hline\hline Q_{32}^{(3)} & Q_{33}^{(3)} \end{array}\right), \quad D_3 = \left(\begin{array}{c|c} Q_{11}^{(3)} & Q_{21}^{(3)} \\ \hline & Q_{22}^{(3)} \end{array}\right),$$

$$Q_4 = \left(\begin{array}{c|c||c} Q_{11}^{(4)} & Q_{12}^{(4)} & \\ \hline & Q_{22}^{(4)} & Q_{23}^{(4)} \\ \hline\hline & & Q_{33}^{(4)} \end{array}\right) = \left(\begin{array}{c||c} D_4 & Q_{23}^{(4)} \\ \hline\hline & Q_{33}^{(4)} \end{array}\right), \quad D_4 = \left(\begin{array}{c|c} Q_{11}^{(4)} & Q_{21}^{(4)} \\ \hline & Q_{22}^{(4)} \end{array}\right).$$

We notice that the diagonal blocks of these matrices are just lower-triangular or upper-triangular. Thus, by Lemma 6.5.2, for any k, Q_k's eigenvalues are composed by D_k's and $Q_{33}^{(k)}$'s eigenvalues, where D_k's eigenvalues determined by $Q_{11}^{(k)}$'s and $Q_{22}^{(k)}$'s. So, the collection of the eigenvalues of $Q_{11}^{(k)}$, $Q_{22}^{(k)}$ and $Q_{33}^{(k)}$, determines $Q^{(k)}$'s eigenvalues. Since all eigenvalues of $Q_{\alpha\alpha}^{(k)}$ $(\alpha = 1, 2, 3)$ are either zeros or have negative real parts, therefore $Q^{(k)}$'s eigenvalues also share this property.

D.5 A proof of (6.42) in Section 6.4.2.2

This subsection gives a proof of (6.42) presented in Section 6.4.2.2.

Let

$$B(t) = \begin{pmatrix} -a\beta(t) & b \\ a\beta(t) & -b \end{pmatrix} = U(t) \begin{pmatrix} 0 & 0 \\ 0 & -(a\beta(t)+b) \end{pmatrix} U^{-1}(t),$$

where

$$U(t) = \begin{pmatrix} \frac{b}{a\beta(t)+b} & 1 \\ \frac{a\beta(t)}{a\beta(t)+b} & -1 \end{pmatrix}, \quad U^{-1}(t) = \begin{pmatrix} 1 & 1 \\ \frac{a\beta(t)}{a\beta(t)+b} & -\frac{b}{a\beta(t)+b} \end{pmatrix}.$$

Let $X(t) = e^{tB(t)}X(0)$ and

$$\hat{X}(t) = U(t) \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} U^{-1}(t)X(0),$$

where $X(0) = (x_1(0), x_2(0))^T$ and $x_1(0) + x_2(0) = M$. Obviously,

$$\hat{X}(t) = \begin{pmatrix} \frac{bM}{a\beta(t)+b} \\ \frac{a\beta(t)M}{a\beta(t)+b} \end{pmatrix}.$$

We will prove

$$\lim_{t\to\infty} \|X(t) - \hat{X}(t)\| = 0.$$

In fact,

$$(tB(t))^k = \left[U(t) \begin{pmatrix} 0 & 0 \\ 0 & -(a\beta(t)+b) \end{pmatrix} U(t)^{-1} \right]^k = U(t) \begin{pmatrix} 0^k & 0 \\ 0 & [-t(a\beta(t)+b)]^k \end{pmatrix} U(t)^{-1},$$

and $e^{-t(a\beta(t)+b)} = \sum_{k=0}^{\infty} \frac{[-t(a\beta(t)+b)]^k}{k!}$. Therefore,

$$\begin{aligned} e^{tB(t)} &= \sum_{k=0}^{\infty} \frac{(tB(t))^k}{k!} \\ &= \sum_{k=0}^{\infty} \frac{1}{k!} \left[U(t) \begin{pmatrix} 0^k & 0 \\ 0 & -[t(a\beta(t)+b)]^k \end{pmatrix} U(t)^{-1} \right] \\ &= U(t) \begin{pmatrix} 1 & 0 \\ 0 & \sum_{k=0}^{\infty} \frac{[-t(a\beta(t)+b)]^k}{k!} \end{pmatrix} U(t)^{-1} \\ &= U(t) \begin{pmatrix} 1 & 0 \\ 0 & e^{-t(a\beta(t)+b)} \end{pmatrix} U(t)^{-1}. \end{aligned}$$

By simple calculations,

$$\begin{aligned} & U(t) \begin{pmatrix} 1 & 0 \\ 0 & e^{-t(a\beta(t)+b)} \end{pmatrix} U(t)^{-1} \\ &= \begin{pmatrix} \frac{b}{a\beta(t)+b} & 1 \\ \frac{a\beta(t)}{a\beta(t)+b} & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & e^{-t(a\beta(t)+b)} \end{pmatrix} \begin{pmatrix} 1 & 1 \\ \frac{a\beta(t)}{a\beta(t)+b} & -\frac{b}{a\beta(t)+b} \end{pmatrix} \\ &= \begin{pmatrix} \frac{b}{a\beta(t)+b} + e^{-t(a\beta(t)+b)} \frac{a\beta(t)}{a\beta(t)+b} & \frac{b}{a\beta(t)+b} - e^{-t(a\beta(t)+b)} \frac{b}{a\beta(t)+b} \\ \frac{a\beta(t)}{a\beta(t)+b} - e^{-t(a\beta(t)+b)} \frac{a\beta(t)}{a\beta(t)+b} & \frac{a\beta(t)}{a\beta(t)+b} + e^{-t(a\beta(t)+b)} \frac{b}{a\beta(t)+b} \end{pmatrix}. \end{aligned}$$

Thus

$$e^{tB(t)} = \begin{pmatrix} \frac{b}{a\beta(t)+b} + e^{-t(a\beta(t)+b)} \frac{a\beta(t)}{a\beta(t)+b} & \frac{b}{a\beta(t)+b} - e^{-t(a\beta(t)+b)} \frac{b}{a\beta(t)+b} \\ \frac{a\beta(t)}{a\beta(t)+b} - e^{-t(a\beta(t)+b)} \frac{a\beta(t)}{a\beta(t)+b} & \frac{a\beta(t)}{a\beta(t)+b} + e^{-t(a\beta(t)+b)} \frac{b}{a\beta(t)+b} \end{pmatrix},$$

and therefore,

$$e^{tB(t)} - \begin{pmatrix} \frac{b}{a\beta(t)+b} & \frac{b}{a\beta(t)+b} \\ \frac{a\beta(t)}{a\beta(t)+b} & \frac{a\beta(t)}{a\beta(t)+b} \end{pmatrix} = \begin{pmatrix} e^{-t(a\beta(t)+b)} \frac{a\beta(t)}{a\beta(t)+b} & -e^{-t(a\beta(t)+b)} \frac{b}{a\beta(t)+b} \\ -e^{-t(a\beta(t)+b)} \frac{a\beta(t)}{a\beta(t)+b} & e^{-t(a\beta(t)+b)} \frac{b}{a\beta(t)+b} \end{pmatrix}.$$

Notice that $0 \leq \frac{a\beta(t)}{a\beta(t)+b} \leq \frac{a}{a+b}$ and $\frac{b}{a\beta(t)+b} \leq \frac{b}{a+b}$, because $0 \leq \beta(t) \leq 1$, so as $t \longrightarrow \infty$,

$$0 \leq e^{-t(a\beta(t)+b)} \frac{a\beta(t)}{a\beta(t)+b} \leq e^{-bt} \frac{a}{a+b} \longrightarrow 0,$$

$$0 \leq e^{-t(a\beta(t)+b)} \frac{b}{a\beta(t)+b} \leq e^{-bt} \frac{b}{a+b} \longrightarrow 0.$$

Therefore,

$$\begin{aligned} &\lim_{t\to\infty} \left\| e^{tB(t)} - \begin{pmatrix} \frac{b}{a\beta(t)+b} & \frac{b}{a\beta(t)+b} \\ \frac{a\beta(t)}{a\beta(t)+b} & \frac{a\beta(t)}{a\beta(t)+b} \end{pmatrix} \right\| \\ &= \lim_{t\to\infty} \left\| \begin{pmatrix} e^{-t(a\beta(t)+b)} \frac{a\beta(t)}{a\beta(t)+b} & -e^{-t(a\beta(t)+b)} \frac{b}{a\beta(t)+b} \\ -e^{-t(a\beta(t)+b)} \frac{a\beta(t)}{a\beta(t)+b} & e^{-t(a\beta(t)+b)} \frac{b}{a\beta(t)+b} \end{pmatrix} \right\| \\ &= 0. \end{aligned} \tag{D.17}$$

Since

$$\begin{pmatrix} \frac{b}{a\beta(t)+b} & \frac{b}{a\beta(t)+b} \\ \frac{a\beta(t)}{a\beta(t)+b} & \frac{a\beta(t)}{a\beta(t)+b} \end{pmatrix} \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} \frac{b(x_1(0)+x_2(0))}{a\beta(t)+b} \\ \frac{a\beta(t)(x_1(0)+x_2(0))}{a\beta(t)+b} \end{pmatrix} = \begin{pmatrix} \frac{bM}{a\beta(t)+b} \\ \frac{a\beta(t)M}{a\beta(t)+b} \end{pmatrix}$$

and $X(0) = (x_1(0), x_2(0))^T$, then

$$\begin{aligned} &\left\| X(t) - \begin{pmatrix} \frac{bM}{a\beta(t)+b} \\ \frac{a\beta(t)M}{a\beta(t)+b} \end{pmatrix} \right\| \\ &= \left\| e^{tB(t)} X(0) - \begin{pmatrix} \frac{b}{a\beta(t)+b} & \frac{b}{a\beta(t)+b} \\ \frac{a\beta(t)}{a\beta(t)+b} & \frac{a\beta(t)}{a\beta(t)+b} \end{pmatrix} X(0) \right\| \\ &\leq \left\| e^{tB(t)} - \begin{pmatrix} \frac{b}{a\beta(t)+b} & \frac{b}{a\beta(t)+b} \\ \frac{a\beta(t)}{a\beta(t)+b} & \frac{a\beta(t)}{a\beta(t)+b} \end{pmatrix} \right\| \|X(0)\|. \end{aligned}$$

By (D.17), as time goes to infinity, we have

$$\left\| X(t) - \begin{pmatrix} \frac{bM}{a\beta(t)+b} \\ \frac{a\beta(t)M}{a\beta(t)+b} \end{pmatrix} \right\| \leq \left\| e^{tB(t)} - \begin{pmatrix} \frac{b}{a\beta(t)+b} & \frac{b}{a\beta(t)+b} \\ \frac{a\beta(t)}{a\beta(t)+b} & \frac{a\beta(t)}{a\beta(t)+b} \end{pmatrix} \right\| \|X(0)\| \longrightarrow 0.$$

That is

$$\lim_{t\to\infty} \|X(t) - \hat{X}(t)\| = \lim_{t\to\infty} \left\| X(t) - \begin{pmatrix} \frac{bM}{a\beta(t)+b} \\ \frac{a\beta(t)M}{a\beta(t)+b} \end{pmatrix} \right\| = 0.$$

D.6 A proof of Lemma 6.4.1

Let $\lambda(\beta)$ be a nonzero eigenvalue of the following $f(\beta)$:

$$f(\beta) = \begin{pmatrix} -a\beta & b \\ a\beta & -b \end{pmatrix}.$$

We will prove the following lemma which states that the real part of $\lambda(\beta)$ is negative. The proof given here does not rely on the explicit expression of the eigenvalue.

Lemma 6.4.1: For any $\beta \in [0,1]$, $\Re(\lambda(\beta)) < 0$, where $\lambda(\beta)$ is a nonzero eigenvalue of $f(\beta)$.

Proof. After a shift $\max\{a\beta, b\}I$, $f(\beta)$ becomes $\tilde{f}(\beta) = f(\beta) + \max\{a\beta, b\}I$, which is a nonnegative matrix. Then similarly to the proof of Theorem 6.5.4, which is based on the Perron-Frobenious theorem (Theorem 6.5.1), we can conclude that the eigenvalue other than zero has negative real part. □