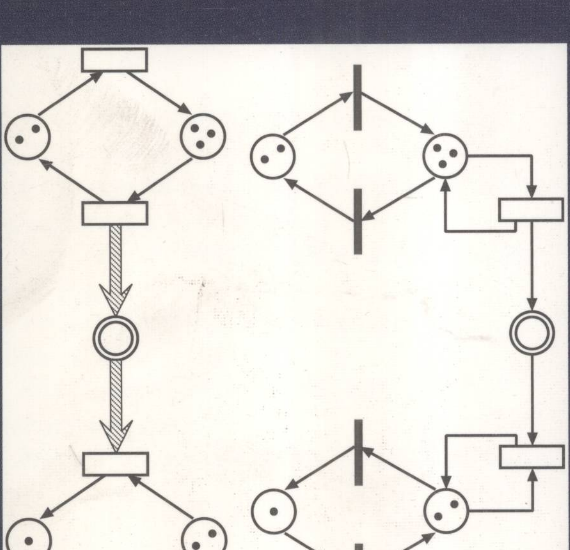
Marco Bernardo
Jane Hillston (Eds.)

Tutorial

LNCS 4486

Formal Methods for Performance Evaluation

7th International School on Formal Methods for the Design of Computer, Communication and Software Systems, SFM 2007
Bertinoro, Italy, May/June 2007, Advanced Lectures



Springer

Marco Bernardo Jane Hillston (Eds.)

Formal Methods for Performance Evaluation

7th International School on Formal Methods for the Design of Computer, Communication and Software Systems, SFM 2007
Bertinoro, Italy, May 28-June 2, 2007
Advanced Lectures

Springer

Volume Editors

Marco Bernardo
Università di Urbino "Carlo Bo"
Istituto di Scienze e Tecnologie dell'Informazione
Piazza della Repubblica 13, 61029 Urbino, Italy
E-mail: bernardo@sti.uniurb.it

Jane Hillston
The University of Edinburgh
Laboratory for Foundations of Computer Science
Mayfield Road, Edinburgh EH9 3JZ, UK
E-mail: jeh@inf.ed.ac.uk

Library of Congress Control Number: 2007926022

CR Subject Classification (1998): D.2.4, D.2, D.3, F.3, C.3, C.2.4

LNCS Sublibrary: SL 2 – Programming and Software Engineering

ISSN 0302-9743
ISBN-10 3-540-72482-6 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-72482-7 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12063000 06/3180 5 4 3 2 1 0

Lecture Notes in Computer Science 4486

Preface

This volume presents the set of papers accompanying the lectures of the seventh International School on Formal Methods for the Design of Computer, Communication and Software Systems (SFM).

This series of schools addresses the use of formal methods in computer science as a prominent approach to the rigorous design of computer, communication and software systems. The main aim of the SFM series is to offer a good spectrum of current research in foundations as well as applications of formal methods, which can be of help for graduate students and young researchers who intend to approach the field.

SFM 2007 was devoted to formal techniques for performance evaluation and covered several aspects of the field, including formalisms for performance modeling (Markov chains, queueing networks, stochastic Petri nets, and stochastic process algebras), equivalence checking and model checking, efficient solution techniques, and software performance engineering.

The opening paper by Stewart presents Markov chains, the fundamental performance modeling formalism in use since the early 1900s. The author outlines the events that have led to the present state of the art in the numerical approach to Markov chain performance modeling and describes current solution methods and ongoing research efforts.

The paper by Balsamo and Marin is about queueing networks, a class of stochastic models extensively applied to represent and analyze resource-sharing systems such as communication and computer systems. The authors mostly focus on product-form queueing networks, which allow one to define efficient algorithms to evaluate average performance measures.

The paper by Balbo illustrates generalized stochastic Petri nets, a modeling formalism that can be conveniently used both for the functional verification of complex models of discrete-event dynamic systems and for their performance and reliability evaluation.

The paper by Clark, Gilmore, Hillston, and Tribastone provides an introduction to stochastic process algebras and their use in performance modeling, with a focus on the PEPA formalism. The authors describe the compositional modeling capabilities of the formalism and the tools available to support Markov-chain-based analysis.

The paper by Bernardo defines and compares several Markovian behavioral equivalences with respect to a number of criteria such as their discriminating power, the exactness of the Markov-chain-level aggregations they induce, the achievement of the congruence property, the existence of sound and complete axiomatizations, the existence of logical characterizations, and the existence of efficient verification algorithms.

The paper by Kwiatkowska, Norman, and Parker presents an overview of model checking for both discrete-time and continuous-time Markov chains, which deals with algorithms for verifying them against specifications written in probabilistic extensions of temporal logic, including quantitative properties with rewards. The authors also outline the main features supported by the probabilistic model checker PRISM.

The paper by Gribaudo and Telek summarizes the basic concepts and the potential use of Markov fluid models, together with the factors that determine the limits of their solvability and practical guidelines that can be extracted from these factors to establish the applicability of fluid models in practice.

The paper by Knottenbelt and Bradley explores an array of techniques for analyzing stochastic performance models with large state spaces. The authors concentrate on explicit techniques suitable for unstructured state spaces and show how memory and run-time requirements can be reduced using a combination of probabilistic algorithms, disk-based solution techniques, and communication-efficient parallelism based on hypergraph partitioning.

The paper by Ciardo discusses some important classes of decision diagrams and shows how they can be effectively employed to derive symbolic algorithms for the analysis of large discrete-state models. In particular, the author presents both explicit and symbolic algorithms for state-space generation, CTL model checking, and continuous-time Markov chain solution.

The paper by Smith reviews the origins of software performance engineering (SPE) and covers its fundamental elements: the data required, the software performance models, and the SPE process. The author also illustrates how to apply the modeling and analysis techniques and reports on the current status as well as the outstanding problems.

The closing paper by Woodside is about using the SPT/MARTE annotations to capture important performance features of a software design, such as platform operations, component submodel composition, state machine uses, and communication costs and delays. The author also addresses the relationship of the annotated design model to the different kinds of performance model that can be extracted.

We believe that this book offers a comprehensive view of what has been done and what is going on worldwide in the field of formal methods for performance evaluation. We wish to thank all the lecturers and all the participants for a lively and fruitful school. We also wish to thank the entire staff of the University Residential Center of Bertinoro for the organizational and administrative support. Finally, we are very grateful to BiCi – Bertinoro international Center for informatics, which kindly provided a sponsorship for this event under the Leonardo Melandri Program.

June 2007

Marco Bernardo
Jane Hillston

Lecture Notes in Computer Science

For information about Vols. 1–4369

please contact your bookseller or Springer

Vol. 4510: P. Van Hentenryck, L. Wolsey (Eds.), Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. X, 391 pages. 2007.

Vol. 4486: M. Bernardo, J. Hillston (Eds.), Formal Methods for Performance Evaluation. VII, 469 pages. 2007.

Vol. 4483: C. Baral, G. Brewka, J. Schlipf (Eds.), Logic Programming and Nonmonotonic Reasoning. IX, 327 pages. 2007. (Sublibrary LNAI).

Vol. 4481: J.T. Yao, P. Lingras, W.-Z. Wu, M. Szczuka, N.J. Cercone, D. Ślęzak (Eds.), Rough Sets and Knowledge Technology. XIV, 576 pages. 2007. (Sublibrary LNAI).

Vol. 4480: A. LaMarca, M. Langheinrich, K.N. Truong (Eds.), Pervasive Computing. XIII, 369 pages. 2007.

Vol. 4470: Q. Wang, D. Pfahl, D.M. Raffo (Eds.), Software Process Change – Meeting the Challenge. XI, 346 pages. 2007.

Vol. 4464: E. Dawson, D.S. Wong (Eds.), Information Security Practice and Experience. XIII, 361 pages. 2007.

Vol. 4463: I. Măndoiu, A. Zelikovsky (Eds.), Bioinformatics Research and Applications. XV, 653 pages. 2007. (Sublibrary LNBI).

Vol. 4462: D. Sauveron, K. Markantonakis, A. Bilas, J.-J. Quisquater (Eds.), Information Security Theory and Practices. XII, 255 pages. 2007.

Vol. 4459: C. Cérin, K.-C. Li (Eds.), Advances in Grid and Pervasive Computing. XVI, 759 pages. 2007.

Vol. 4453: T. Speed, H. Huang (Eds.), Research in Computational Molecular Biology. XVI, 550 pages. 2007. (Sublibrary LNBI).

Vol. 4450: T. Okamoto, X. Wang (Eds.), Public Key Cryptography – PKC 2007. XIII, 491 pages. 2007.

Vol. 4448: M. Giacobini et al. (Ed.), Applications of Evolutionary Computing. XXIII, 755 pages. 2007.

Vol. 4447: E. Marchiori, J.H. Moore, J.C. Rajapakse (Eds.), Evolutionary Computation,Machine Learning and Data Mining in Bioinformatics. XI, 302 pages. 2007.

Vol. 4446: C. Cotta, J. van Hemert (Eds.), Evolutionary Computation in Combinatorial Optimization. XII, 241 pages. 2007.

Vol. 4445: M. Ebner, M. O'Neill, A. Ekárt, L. Vanneschi, A.I. Esparcia-Alcázar (Eds.), Genetic Programming. XI, 382 pages. 2007.

Vol. 4444: T. Reps, M. Sagiv, J. Bauer (Eds.), Program Analysis and Compilation, Theory and Practice. X, 361 pages. 2007.

Vol. 4443: R. Kotagiri, P.R. Krishna, M. Mohania, E. Nantajeewarawat (Eds.), Advances in Databases: Concepts, Systems and Applications. XXI, 1126 pages. 2007.

Vol. 4440: B. Liblit, Cooperative Bug Isolation. XV, 101 pages. 2007.

Vol. 4439: W. Abramowicz (Ed.), Business Information Systems. XV, 654 pages. 2007.

Vol. 4438: L. Maicher, A. Sigel, L.M. Garshol (Eds.), Leveraging the Semantics of Topic Maps. X, 257 pages. 2007. (Sublibrary LNAI).

Vol. 4433: E. Şahin, W.M. Spears, A.F.T. Winfield (Eds.), Swarm Robotics. XII, 221 pages. 2007.

Vol. 4432: B. Beliczynski, A. Dzielinski, M. Iwanowski, B. Ribeiro (Eds.), Adaptive and Natural Computing Algorithms, Part II. XXVI, 761 pages. 2007.

Vol. 4431: B. Beliczynski, A. Dzielinski, M. Iwanowski, B. Ribeiro (Eds.), Adaptive and Natural Computing Algorithms, Part I. XXV, 851 pages. 2007.

Vol. 4430: C.C. Yang, D. Zeng, M. Chau, K. Chang, Q. Yang, X. Cheng, J. Wang, F.-Y. Wang, H. Chen (Eds.), Intelligence and Security Informatics. XII, 330 pages. 2007.

Vol. 4429: R. Lu, J.H. Siekmann, C. Ullrich (Eds.), Cognitive Systems. X, 161 pages. 2007. (Sublibrary LNAI).

Vol. 4427: S. Uhlig, K. Papagiannaki, O. Bonaventure (Eds.), Passive and Active Network Measurement. XI, 274 pages. 2007.

Vol. 4426: Z.-H. Zhou, H. Li, Q. Yang (Eds.), Advances in Knowledge Discovery and Data Mining. XXV, 1161 pages. 2007. (Sublibrary LNAI).

Vol. 4425: G. Amati, C. Carpineto, G. Romano (Eds.), Advances in Information Retrieval. XIX, 759 pages. 2007.

Vol. 4424: O. Grumberg, M. Huth (Eds.), Tools and Algorithms for the Construction and Analysis of Systems. XX, 738 pages. 2007.

Vol. 4423: H. Seidl (Ed.), Foundations of Software Science and Computational Structures. XVI, 379 pages. 2007.

Vol. 4422: M.B. Dwyer, A. Lopes (Eds.), Fundamental Approaches to Software Engineering. XV, 440 pages. 2007.

Vol. 4421: R. De Nicola (Ed.), Programming Languages and Systems. XVII, 538 pages. 2007.

Vol. 4420: S. Krishnamurthi, M. Odersky (Eds.), Compiler Construction. XIV, 233 pages. 2007.

Vol. 4419: P.C. Diniz, E. Marques, K. Bertels, M.M. Fernandes, J.M.P. Cardoso (Eds.), Reconfigurable Computing: Architectures, Tools and Applications. XIV, 391 pages. 2007.

Vol. 4418: A. Gagalowicz, W. Philips (Eds.), Computer Vision/Computer Graphics Collaboration Techniques. XV, 620 pages. 2007.

Vol. 4416: A. Bemporad, A. Bicchi, G. Buttazzo (Eds.), Hybrid Systems: Computation and Control. XVII, 797 pages. 2007.

Vol. 4415: P. Lukowicz, L. Thiele, G. Tröster (Eds.), Architecture of Computing Systems - ARCS 2007. X, 297 pages. 2007.

Vol. 4414: S. Hochreiter, R. Wagner (Eds.), Bioinformatics Research and Development. XVI, 482 pages. 2007. (Sublibrary LNBI).

Vol. 4412: F. Stajano, H.J. Kim, J.-S. Chae, S.-D. Kim (Eds.), Ubiquitous Convergence Technology. XI, 302 pages. 2007.

Vol. 4411: R.H. Bordini, M. Dastani, J. Dix, A.E.F. Seghrouchni (Eds.), Programming Multi-Agent Systems. XIV, 249 pages. 2007. (Sublibrary LNAI).

Vol. 4410: A. Branco (Ed.), Anaphora: Analysis, Algorithms and Applications. X, 191 pages. 2007. (Sublibrary LNAI).

Vol. 4409: J.L. Fiadeiro, P.-Y. Schobbens (Eds.), Recent Trends in Algebraic Development Techniques. VII, 171 pages. 2007.

Vol. 4407: G. Puebla (Ed.), Logic-Based Program Synthesis and Transformation. VIII, 237 pages. 2007.

Vol. 4406: W. De Meuter (Ed.), Advances in Smalltalk. VII, 157 pages. 2007.

Vol. 4405: L. Padgham, F. Zambonelli (Eds.), Agent-Oriented Software Engineering VII. XII, 225 pages. 2007.

Vol. 4403: S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, T. Murata (Eds.), Evolutionary Multi-Criterion Optimization. XIX, 954 pages. 2007.

Vol. 4401: N. Guelfi, D. Buchs (Eds.), Rapid Integration of Software Engineering Techniques. IX, 177 pages. 2007.

Vol. 4400: J.F. Peters, A. Skowron, V.W. Marek, E. Orłowska, R. Słowiński, W. Ziarko (Eds.), Transactions on Rough Sets VII, Part II. X, 381 pages. 2007.

Vol. 4399: T. Kovacs, X. Llorà, K. Takadama, P.L. Lanzi, W. Stolzmann, S.W. Wilson (Eds.), Learning Classifier Systems. XII, 345 pages. 2007. (Sublibrary LNAI).

Vol. 4398: S. Marchand-Maillet, E. Bruno, A. Nürnberger, M. Detyniecki (Eds.), Adaptive Multimedia Retrieval: User, Context, and Feedback. XI, 269 pages. 2007.

Vol. 4397: C. Stephanidis, M. Pieper (Eds.), Universal Access in Ambient Intelligence Environments. XV, 467 pages. 2007.

Vol. 4396: J. García-Vidal, L. Cerdà-Alabern (Eds.), Wireless Systems and Mobility in Next Generation Internet. IX, 271 pages. 2007.

Vol. 4395: M. Daydé, J.M.L.M. Palma, Á.L.G.A. Coutinho, E. Pacitti, J.C. Lopes (Eds.), High Performance Computing for Computational Science - VECPAR 2006. XXIV, 721 pages. 2007.

Vol. 4394: A. Gelbukh (Ed.), Computational Linguistics and Intelligent Text Processing. XVI, 648 pages. 2007.

Vol. 4393: W. Thomas, P. Weil (Eds.), STACS 2007. XVIII, 708 pages. 2007.

Vol. 4392: S.P. Vadhan (Ed.), Theory of Cryptography. XI, 595 pages. 2007.

Vol. 4391: Y. Stylianou, M. Faundez-Zanuy, A. Esposito (Eds.), Progress in Nonlinear Speech Processing. XII, 269 pages. 2007.

Vol. 4390: S.O. Kuznetsov, S. Schmidt (Eds.), Formal Concept Analysis. X, 329 pages. 2007. (Sublibrary LNAI).

Vol. 4389: D. Weyns, H.V.D. Parunak, F. Michel (Eds.), Environments for Multi-Agent Systems III. X, 273 pages. 2007. (Sublibrary LNAI).

Vol. 4385: K. Coninx, K. Luyten, K.A. Schneider (Eds.), Task Models and Diagrams for Users Interface Design. XI, 355 pages. 2007.

Vol. 4384: T. Washio, K. Satoh, H. Takeda, A. Inokuchi (Eds.), New Frontiers in Artificial Intelligence. IX, 401 pages. 2007. (Sublibrary LNAI).

Vol. 4383: E. Bin, A. Ziv, S. Ur (Eds.), Hardware and Software, Verification and Testing. XII, 235 pages. 2007.

Vol. 4381: J. Akiyama, W.Y.C. Chen, M. Kano, X. Li, Q. Yu (Eds.), Discrete Geometry, Combinatorics and Graph Theory. XI, 289 pages. 2007.

Vol. 4380: S. Spaccapietra, P. Atzeni, F. Fages, M.-S. Hacid, M. Kifer, J. Mylopoulos, B. Pernici, P. Shvaiko, J. Trujillo, I. Zaihrayeu (Eds.), Journal on Data Semantics VIII. XV, 219 pages. 2007.

Vol. 4379: M. Südholt, C. Consel (Eds.), Object-Oriented Technology. VIII, 157 pages. 2007.

Vol. 4378: I. Virbitskaite, A. Voronkov (Eds.), Perspectives of Systems Informatics. XIV, 496 pages. 2007.

Vol. 4377: M. Abe (Ed.), Topics in Cryptology – CT-RSA 2007. XI, 403 pages. 2006.

Vol. 4376: E. Frachtenberg, U. Schwiegelshohn (Eds.), Job Scheduling Strategies for Parallel Processing. VII, 257 pages. 2007.

Vol. 4374: J.F. Peters, A. Skowron, I. Düntsch, J. Grzymała-Busse, E. Orłowska, L. Polkowski (Eds.), Transactions on Rough Sets VI, Part I. XII, 499 pages. 2007.

Vol. 4373: K. Langendoen, T. Voigt (Eds.), Wireless Sensor Networks. XIII, 358 pages. 2007.

Vol. 4372: M. Kaufmann, D. Wagner (Eds.), Graph Drawing. XIV, 454 pages. 2007.

Vol. 4371: K. Inoue, K. Satoh, F. Toni (Eds.), Computational Logic in Multi-Agent Systems. X, 315 pages. 2007. (Sublibrary LNAI).

Vol. 4370: P.P Lévy, B. Le Grand, F. Poulet, M. Soto, L. Darago, L. Toubiana, J.-F. Vibert (Eds.), Pixelization Paradigm. XV, 279 pages. 2007.

¥581.00元

Table of Contents

Performance Modelling and Markov Chains

William J. Stewart*

Department of Computer Science, North Carolina State University,
Raleigh, NC 27695, USA
billy@csc.ncsu.edu

Abstract. Markov chains have become an accepted technique for modeling a great variety of situations. They have been in use since the early 1900's, but it is only in recent years with the advent of high speed computers and cheap memory that they have begun to be applied to large-scale modeling projects. This paper outlines the events that have lead to the present state-of-the-art in the numerical approach to Markov chain performance modeling and describes current solution methods and ongoing research efforts.

1 Introduction

1.1 A.A. Markov

It is appropriate in a talk of this nature to include a few sentences about the life and work of A.A. Markov. Three sources were used to gather the information in this section, namely a paper by Basherin and Naoumov, presented at the Fourth International Conference on the Numerical Solution of Markov Chains, [2], a paper by Seneta [67] given at the 2006 Markov Anniversary Meeting, and a web page maintained by the School of Mathematics and Statistics at the University of St Andrews, Scotland [74]. Andrei Andreevich Markov was born on June 14, 1856, in Ryazan, Russia. He was the fourth of his father's 6 children by his first marriage. His father also had 3 children by his second wife. An inborn deformity of the knee meant that the very young Markov had to use crutches. At age ten he was operated on and thereafter walked with a slight limp. However, it was leg problems that lead to his death. In later life, he developed an anuerysm in the leg which required multiple surgeries, one of which proved fatal. He died on July 20, 1922 at the age of 66 and is buried in the Mytrophany Cemetery in St. Petersburg.

Four years after Markov's birth, the family moved to St. Petersburg where Markov attended school. Apparently he was unsuccessful at many subjects, except mathematics, at which he excelled. He attended the University of St. Petersburg where he studied under P.L. Chebyshev. He was awarded the gold medal and was offered an academic position within the university. His doctorate (1884) was entitled "On Certain Applications of the Algebraic Continuous Fractions". When Chebychev left the university in 1883, Markov took over his probability theory course. Twenty years later he was made an honorary professor, and shortly thereafter, he retired, although he continued to lecture on probability theory and the theory of continuous fractions.

* Corresponding author.

M. Bernardo and J. Hillston (Eds.): SFM 2007, LNCS 4486, pp. 1–33, 2007.

A.A. Markov married Maria Valvatieva, the daughter of the owner of the estate on which Markov's father worked, in 1883. They has a son who was given the same name, Andrei Andreevich, and who became a renown mathematician in his own right. A.A. Markov Jr. worked in the fields of algebra, topology and mathematical logic and headed the Department of Mathematical Logic at Moscow State University. Interestingly, this position was filled by A.N. Kolmogorov after Markov Jr.'s death. It was Kolmogorov who laid the foundations for the general theory of Markov processes.

Much of Markov's work was concerned with investigations into the weak law of large numbers and the central limit theorem. His introduction of the Markov chain was to show that Chebyshev's approach to extending the weak law of large numbers to sums of dependent random variables could be extended even further. He introduced a *simple chain* as "an infinite sequence $x_1,\ x_2,\ \ldots,\ x_k,\ x_{k+1},\ \ldots$ of variables connected in such a way that x_{k+1} for any k is independent of $x_1,\ x_2,\ \ldots,\ x_{k-1}$, in case x_k is known". Furthermore he extended this to *complex chains* in which "every number is directly connected not with a single but with several preceding numbers". For the most part, his studies on chains involved simple homogeneous chains. His works address the concept of irreducibility and he shows that the dominant eigenvalue of an irreducible Markov chain must be one and that no other eigenvalue can exceed this in modulus. Markov's work has been the basis for much research and provides a powerful method of analysis that is in vogue today.

1.2 Vic Wallace and the Recursive Queue Analyzer

To the best of this author's knowledge, the first applications of Markov chains to the performance evaluation of computer systems occurred in the early 1960's at the University of Michigan. The *Michigan Computer Modelling Project* was established in the Electrical Engineering department under the direction of Dr. Harrry Goode, the father of "Systems Engineering". Unfortunately Goode was killed in a car accident shortly after its establishment. The project was supported by the *Rome Air Development Center* and later by ARPA. The participants in the original project included Vic Wallace, as principal investigator, Richard Evans, Eugene Lawler, Dennis Fife, Robert Carlson, Richard Rosenberg, Robert Rosen and John Smith. The original project was followed by others under the direction of Keki Irani. The specification for the first software package for Markov modelling, the *Recursive Queue Analyzer*, RQA-0, appeared in the first report of the research group in 1964. This was a prototype program for the numerical solution of the equilibrium state probabilities of Markov chain models and included the use of sparse matrix algorithms and compact storage for the transition matrix. A software package based on these specifications, called RQA-1, was written by Wallace and Rosenberg and appeared in 1966.

In a paper presented at the first international conference on the numerical solution of Markov chains [78], (and from which much of the content of this section is extracted), Wallace cites a number of reasons why numerical solution techniques took so long to develop and suggests that these same reasons were still present some 25 years after the introduction of the Recursive Queue Analyzer. Among these was the idea that computer Markov modelling was too new and unfamiliar, being more "mathematical" and abstract, and not to be trusted. He also suggested that other techniques like product form networks and simulation, were distracting attention because they could be more readily understood, and there were still enough problems that they could solve. Perhaps more significantly, he postulated that the numerical techniques tended to be quite fragile, that traps abounded to snare the unwary!

To illustrate one aspect of this, Wallace refers to an unsuccessful experience in the application of RQA-1. He recalls in the above referenced paper, that although they had been doing numerical Markov analysis for many years, "I can recall the embarrassment in 1970 of encountering a seriously ill-conditioned model for the first time while confidently engaged in consulting. No adjustments seemed to eliminate the problem, and tens of thousands of iterations were getting nowhere." It turns out that the corporation for which Wallace was consulting was ICL, International Computers Limited, England. The problem was to model a SCAN strategy on a disk store with a large number of cylinders. The difficulty, it later turned out, was the occurrence of multiple eigenvalues close to unity. The problems that Wallace faced are still with us today, even though much progress has been made in the intervening years. These are the problems of finding stable and efficient algorithms for computing numerical solutions and data structures in which to store and manipulate the transition matrices.

1.3 Alan Scherr

Alan Scherr enters the picture as a PhD student at M.I.T. in the process of completing his thesis in electrical engineering in 1965. The university had recently acquired a new "Compatible Time Sharing System" which allowed 300 users to simultaneously access its software and run their programs. Scherr's performance problem was to characterize the system's usage which he did by extensive simulations. On presenting his thesis he was told that it just was not "academic" enough, that it needed more mathematical formulas. To jump this final hurdle, Scherr used some techniques he had learned in a recently completed operations research course to construct a very primitive Markov chain model. He designated the integer n to represent the state of the entire system when n users had submitted their requests — which leaves $300 - n$ users busy typing and preparing to submit their request. What was truly amazing and unexpected was the accuracy with which this simple model appeared to capture the actual behavior of the computer. It was the success of this simple model that encouraged others to adopt the Markov chain approach for performance evaluation. Scherr's thesis was later awarded the Grace Murray Hopper award by the ACM. More information on the work of Scherr is can be found in a paper by Von Hilgers and Langville [76].

2 Context for Current State-of-the-Art

In the context of Performance Evaluation, numerical analysis methods refer to those methods which work with a Markov chain representation of the system under evaluation and use techniques from the domain of numerical analysis to compute stationary and/or transient state probabilities or other measures of interest. It is often possible to represent the behavior of a physical system by describing all the different states that it can occupy and by indicating how the system moves from one state to another in time.

If the time spent in any state is exponentially distributed, the system may be represented by a *Markov process*. Even when the system does not possess this exponential property explicitly, it is usually possible to construct a corresponding implicit representation. When the state space is discrete, the term *Markov chain* is employed. The system being modelled by the chain is assumed to occupy one and only one of these states at any moment in time and the evolution of the system is represented by transitions of the Markov chain from one state to another. The information that is most often sought from such a model is the probability of being in a given state or subset of states at a certain time after the system becomes operational. Often this time is taken to be sufficiently long that all influence of the initial starting state has been erased. The probabilities thus obtained are referred to as the *long-run* or *stationary probabilities*. Probabilities at a particular time t are called *transient probabilities*.

It follows that the three steps involved in carrying out this type of evaluation are firstly, to describe the system to be analyzed as a Markov chain; secondly, to determine from this Markov chain description, a matrix of transition rates or probabilities; and thirdly, from this matrix representation, to numerically compute all performance measures of interest. The first involves characterizing the states of the system and formulating the manner in which it moves from one state to another; the second requires finding a manner in which to store the transition matrix efficiently; the third requires the application of matrix equation solving techniques to compute stationary or transient probabilities.

The Recursive Queue Analyzer, RQA-1, [77], essentially avoided the first step by requiring a user to describe the transition matrix directly. Since the envisaged applications derived from queueing networks, the nonzero elements in the transition matrix often repeat at well defined patterns in the matrix. RQA-1 defined a data structure which attempted to capture such regularities and to store them as the trio (nonzero element, pattern, initialization point). The amount of storage used was therefore minimized. The numerical solution technique employed by RQA-1 was the *Power method*. In the literature, the authors reported some success with this approach, [78].

This was followed in the early 1970's by this author's *Markov Chain Analyzer*, MARCA, [73]. MARCA provided a means of expressing a Markov chain as a system of "Balls and Buckets", essentially allowing a single state of the chain to be represented as a vector, the *state descriptor vector*. The user has to characterize the way in which the system changes states by describing the interactions among the components of the state descriptor vector. With this information MARCA automatically generates the transition rate matrix and stores it in a compact form. The original solution method used in MARCA was simultaneous iteration, a forerunner of the currently popular projection methods. In 1974, a restriction of MARCA to queueing networks was developed and incorporated into QNAP (Queueing Network Analysis Package), [56].

The 80's witnessed the popularization of the matrix-geometric approach of Neuts, [49], and the establishment of Generalized Stochastic Petri Nets (GSPN) as a valuable modelling paradigm, [1, 15]. These advances were accompanied by a flurry of activity in numerical aggregation and disaggregation methods, [13, 44, 65] — extending the seminal work of Courtois, [18], on nearly completely decomposable systems. This period also saw advances in the computation of bounds, [26, 66], in specification techniques, [6], in state-space exploration, [12, 7], and so on. The most popular applications were, and still are, in the fields of computer communications and reliability modelling, [36, 40, 81, 82]. And, of course, this period was also rich in advances led by the numerical analysis community, especially in the development of projection methods, preconditioning techniques and in sparse matrix technology in general, [62].

To this day, research continues along the same paths. Advances continue to be made in all the aforementioned areas. In addition we see an increased emphasis placed on stochastic automata networks (SANs), and other structured analysis approaches, [10, 11, 16, 41, 53, 72]. These have advanced hand in hand with compositional approaches, such as the stochastic process algebra package, PEPA, [32]. Given the ease of with which GUIs (Graphical User Interaces) can now be programmed and the availability of cheap memory and fast CPUs, many more numerical analysis software packages specificially designed for performance evaluation have made their apparation. This period also witnessed international conferences devoted to the topic, the first in 1990 and the second in 1995. It is significant that the third and fourth in this series were held jointly — with the PNPM (Petri Nets and Performance Models) conference and the PAPM (Process Algebra and Performance Modelling) conference in 1999 in Zaragoza, Spain and with the PNPM (Petri Nets and Performance Models) conference and the Performance Tools and Techniques, Tools'03 conference in Urbana-Champaign in 2003. In a short paper like this, it is not possible to cover all aspects of the current state-of-the-art in anything other than a perfunctory fashion. To offset this to some degree, an extensive bibliography is provided.

3 The Numerical Solution of Markov Chains

In Markov chain performance evaluation, all the desired performance measures are largely computed from the stationary and transient distributions of the Markov chain. In particular, the computation of stationary distributions is generally referred to as solving the global balance equations. These performance measures are computed from the stochastic transition probability matrix P of the Markov chain The elements p_{ij} of this matrix P are the *conditional probabilities* that on leaving state i the Markov chain next moves to state j. The relevant equations may be written as

$$\pi P = \pi, \tag{1}$$

or alternatively, as

$$\pi Q = 0, \tag{2}$$

where $P = Q\Delta t + I$ and $\Delta t \leq (\max_i |q_{ii}|)^{-1}$. When we perform this operation we essentially convert the continuous-time system represented by the *transition rate* matrix, Q, to a discrete-time system represented by the stochastic *transition probability* matrix, P. In the discrete-time system, transitions take place at intervals of time Δt, this parameter being chosen so that the probability of two transitions taking place in time Δt is negligible. The stationary distribution, π, may be computed from either of these equations. The transient distribution is computed form the Chapmann-Kolmogoroff differential difference equations

$$\begin{cases} \dfrac{d\pi(t)}{dt} = \pi(t)Q, \; t \in [0,\; T] \\ \pi(0) \;= \pi_0 \qquad \text{an initial probability distribution.} \end{cases}$$

We shall discuss computational aspects of the stationary and transient distributions momentarily, but for the moment we shall address a topic that is receiving much attention in the current literature, the set of **right-hand** eigenvectors of P. An understanding of the significance of these vectors is becoming increasing important in applications such as data mining and search engine development.

3.1 Significance of Subdominant, Right-Hand Eigenvectors

It is known that a *left-hand* eigenvector corresponding to a unit eigenvalue of the stochastic transition probability matrix of a Markov chain is its stationary probability vector. As yet, no physical significance has been ascribed to the left-hand eigenvectors corresponding to eigenvalues different from unity. The situation is otherwise for the set of right-hand eigenvectors. When the Markov chain under consideration is irreducible and noncyclic, its stochastic matrix has a single eigenvalue of modulus 1; i.e., the unit eigenvalue. In this case some information concerning the tendencies of the states to form groups may be obtained from an examination of the right-hand eigenvectors corresponding to the *subdominant* eigenvalues, i.e., the eigenvalues with modulus closest to but strictly less than 1.0. The reason is as follows:

The equilibrium position of the system is defined by the stationary probability vector, i.e., the left-hand eigenvector corresponding to the unit eigenvalue. With each state of the system can be associated a real number, which determines its "distance" from this equilibrium position. This distance may be regarded as the number of iterations (or the length of time) required to reach the equilibrium position if the system starts in the state for which the distance is being measured. Such measurements are, of course, only relative, but they serve as a means of comparison among the states.

Let the row vector $w_i^{(1)} = (0, 0, \ldots, 1, \ldots, 0)$ with i^{th} component equal to 1, denote that initially the system is in state i. We shall assume that P possesses a full set of n linearly independent eigenvectors. Similar results may be obtained when eigenvectors and principal vectors are used instead. Let $x_1, x_2, \ldots, x_n$ be the left-hand eigenvectors of P (i.e., $x_j^T P = \lambda_j x_j^T$ for all $j = 1, 2, \ldots, n$), arranged into descending order according to the magnitude of their corresponding eigenvalues. Writing $w_i^{(1)}$ as a linear combination of these eigenvectors, we have

$$w_i^{(1)} = c_{i1}x_1^T + c_{i2}x_2^T + \ldots + c_{in}x_n^T$$

where $c_{i1}, c_{i2}, \ldots, c_{in}$ are the constants that define the linear combination. Repeated postmultiplication of $w_i^{(1)}$ by P yields the steady-state probability vector. We have

$$w_i^{(1)}P = c_{i1}x_1^T P + c_{i2}x_2^T P + \ldots + c_{in}x_n^T P \tag{3}$$

$$= c_{i1}x_1^T + c_{i2}\lambda_2 x_2^T + \ldots + c_{in}\lambda_n x_n^T = w_i^{(2)}, \tag{4}$$

and in general

$$w_i^{(k+1)} = c_{i1}x_1^T + c_{i2}\lambda_2^k x_2^T + \ldots + c_{in}\lambda_n^k x_n^T.$$

If the system initially starts in some other state $j \neq i$, we have

$$w_j^{(k+1)} = c_{j1}x_1^T + c_{j2}\lambda_2^k x_2^T + \ldots + c_{jn}\lambda_n^k x_n^T.$$

Since only the constant coefficients differ, the difference in the length of time taken to reach the steady state from any two states i and j depends only on these constant terms. Further, if λ_2 is of strictly larger modulus than $\lambda_3, \lambda_4, \ldots$, then for large k, $\lambda_2^k \gg \lambda_l^k$ for $l \geq 3$, and it is the terms c_{i2} and c_{j2} in particular that contribute to the difference. Considering all possible starting states, we obtain

$$\begin{pmatrix} w_1^{(k+1)} \\ w_2^{(k+1)} \\ \vdots \\ w_n^{(k+1)} \end{pmatrix} = \begin{pmatrix} c_{11}x_1^T + c_{12}\lambda_2^k x_2^T + \ldots + c_{1n}\lambda_n^k x_n^T \\ c_{21}x_1^T + c_{22}\lambda_2^k x_2^T + \ldots + c_{2n}\lambda_n^k x_n^T \\ \vdots \\ c_{n1}x_1^T + c_{n2}\lambda_2^k x_2^T + \ldots + c_{nn}\lambda_n^k x_n^T \end{pmatrix},$$

i.e.,

$$W^{(k+1)} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{pmatrix} \begin{pmatrix} 1 & & & \\ & \lambda_2^k & & \\ & & \ddots & \\ & & & \lambda_n^k \end{pmatrix} \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{pmatrix} \equiv C\Lambda^k X^T.$$

To obtain the matrix C, consider the following: the matrix $W^{(1)} = (w_1^{(1)}, w_2^{(1)}, \ldots, w_n^{(1)})^T$ was originally written in terms of the set of left-hand eigenvectors as

$$W^{(1)} = CX^T,$$

but since $W^{(1)} = I$, we obtain

$$I = CX^T,$$

i.e., $C = (X^T)^{-1} = Y$, the set of right-hand eigenvectors of P. Therefore, it is from the second column of the matrix C, i.e., the subdominant right-hand eigenvector of the matrix P, that an appropriate measure of the relative distance of each state from the stationary probability vector may be obtained. The third and subsequent columns may be employed to obtain subsidiary effects.

States whose corresponding component value in this vector is large in magnitude are, in a relative sense, far from the equilibrium position. Also, the states corresponding to component values that are relatively close together form a cluster, or a subset, of states. If, for example, the components of this vector are close either to $+1$ or to -1, then it may be said that those states corresponding to values close to $+1$ form a subset of states that is far from the remaining states, and vice versa. In this manner, it may be possible to determine which states constitute near essential and near cyclic subsets of states. This technique has been proven useful in a variety of applications in which groups of like states need to be identified and isolated.

3.2 Steady State Distributions

We now turn to the computation of steady-state distributions. A number of powerful numerical procedures are available to us, but unfortunately there is not a single one that works well in all circumstances. It is necessary to choose a particular method that responds well to the size, structure and required performance measures. We examine the various possibilities in the next few sections.

3.2.1 Direct Methods

Direct methods, in contrast to iterative and projection methods, perform a fixed number of numerical operations to compute a solution to a system of equations. All direct methods for systems of linear equations are based on Gaussian elimination. In our case, we apply these methods to equation (1) which is a homogeneous system of linear equations. If the Markov chain is ergodic, the fact that the system of equations is homogeneous does not create any problems, since we may replace any of the n equations by the normalizing equation, $\sum_{j=1}^{n} \pi_j = 1$, and thereby convert it into a nonhomogeneous system with nonsingular coefficient matrix and nonzero right hand side. The solution in this case is well defined. It turns out that replacing an equation with the normalizing equation is not really necessary. The usual approach taken is to construct an LU decomposition of Q and replace the final zero diagonal element of U with an arbitrary value. The solution computed by backsubstitution on U must then be normalized. Furthermore, since the diagonal elements are equal to the negated sum of the off-diagonal elements (Q is, in a restricted sense, diagonally dominant), it is not necessary to perform pivoting while computing the LU decomposition. This simplifies the algorithm considerably. The problem of the size and nonzero structure (the placement of the nonzero elements within the matrix) still remain. Obviously this method will work, and work well, when the number of states is small. It will also work well when the nonzero structure of Q fits into a narrow band along the diagonal. In these cases a very stable variant, referred to as the GTH (Grassmann, Taskar and Heyman, [30]) algorithm may be used. In this variant, all subtraction is avoided by computing diagonal elements as the sum of off-diagonal elements. This is possible since the zero-row-sum property of an infinitesimal generator is invariant under the basic operation of Gaussian elimination, namely adding a multiple of one row into another. For an efficient implementation, the GTH variant requires convenient access to both the rows and the columns of the matrix.

3.2.2 Basic Iterative Methods

When the number of states becomes large and the structure in not banded, the direct approach loses its appeal and one is obliged to turn to other methods. For iterative methods we first take the approach of solving equation (2) in which P is a matrix of transitions probabilities. Let the initial probability distribution vector be given by $\pi^{(0)}$. After the first transition, the probability vector is given by $\pi^{(1)} = \pi^{(0)}P$; after k transitions it is given by $\pi^{(k)} = \pi^{(k-1)}P = \pi^{(0)}P^k$. If the Markov chain is ergodic, then $\lim_{k\rightarrow\infty} \pi^{(k)} = \pi$. This method of determining the stationary probability vector, by successively multiplying some initial probability distribution vector by the matrix of transition probabilities, is called the *Power* method. Observe that all that is required is a vector-matrix multiplication operation. This may be conveniently performed on