András Horváth
Miklós Telek (Eds.)

LNCS 4054

# Formal Methods and Stochastic Models for Performance Evaluation

Third European Performance Engineering Workshop, EPEW 2006
Budapest, Hungary, June 2006
Proceedings

EPEW

Springer

András Horváth Miklós Telek (Eds.)

# Formal Methods and Stochastic Models for Performance Evaluation

Third European Performance Engineering Workshop, EPEW 2006
Budapest, Hungary, June 21-22, 2006
Proceedings

Springer

Volume Editors

András Horváth
Università di Torino, Dipartimento di Informatica
Corso Svizzera 185, 10149 Torino, Italy
E-mail: horvath@di.unito.it

Miklós Telek
Budapest University of Technology and Economics, Department of Telecommunications
P.O. Box 91, 1521 Budapest, Hungary
E-mail: telek@hit.bme.hu

Library of Congress Control Number: 2006927343

CR Subject Classification (1998): D.2.4, C.2.4, F.3, D.4, C.4

LNCS Sublibrary: SL 2 – Programming and Software Engineering

ISSN 0302-9743
ISBN-10 3-540-35362-3 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-35362-1 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: 11777830 06/3142 5 4 3 2 1 0

# Lecture Notes in Computer Science

For information about Vols. 1–3956

please contact your bookseller or Springer

Vol. 4060: K. Futatsugi, J.-P. Jouannaud, J. Meseguer (Eds.), Algebra, Meaning and Computation. XXXVIII, 643 pages. 2006.

Vol. 4058: L.M. Batten, R. Safavi-Naini (Eds.), Information Security and Privacy. XII, 446 pages. 2006.

Vol. 4055: J. Lee, J. Shim, S.-g. Lee, C. Bussler, S. Shim (Eds.), Data Engineering Issues in E-Commerce and Services. IX, 290 pages. 2006.

Vol. 4054: A. Horváth, M. Telek (Eds.), Formal Methods and Stochastic Models for Performance Evaluation. VIII, 239 pages. 2006.

Vol. 4053: M. Ikeda, K.D. Ashley, T.-W. Chan (Eds.), Intelligent Tutoring Systems. XXVI, 821 pages. 2006.

Vol. 4044: P. Abrahamsson, M. Marchesi, G. Succi (Eds.), Extreme Programming and Agile Processes in Software Engineering. XII, 230 pages. 2006.

Vol. 4043: A.S. Atzeni, A. Lioy (Eds.), Public Key Infrastructure. XI, 261 pages. 2006.

Vol. 4041: S.-W. Cheng, C.K. Poon (Eds.), Algorithmic Aspects in Information and Management. XI, 395 pages. 2006.

Vol. 4040: R. Reulke, U. Eckardt, B. Flach, U. Knauer, K. Polthier (Eds.), Combinatorial Image Analysis. XII, 482 pages. 2006.

Vol. 4039: M. Morisio (Ed.), Reuse of Off-the-Shelf Components. XIII, 444 pages. 2006.

Vol. 4038: P. Ciancarini, H. Wiklicky (Eds.), Coordination Models and Languages. VIII, 299 pages. 2006.

Vol. 4037: R. Gorrieri, H. Wehrheim (Eds.), Formal Methods for Open Object-Based Distributed Systems. XVII, 474 pages. 2006.

Vol. 4036: O. H. Ibarra, Z. Dang (Eds.), Developments in Language Theory. XII, 456 pages. 2006.

Vol. 4034: J. Münch, M. Vierimaa (Eds.), Product-Focused Software Process Improvement. XVII, 474 pages. 2006.

Vol. 4033: B. Stiller, P. Reichl, B. Tuffin (Eds.), Performability Has its Price. X, 103 pages. 2006.

Vol. 4031: M. Ali, R. Dapoigny (Eds.), Innovations in Applied Artificial Intelligence. XXIII, 1353 pages. 2006. (Sublibrary LNAI).

Vol. 4027: H.L. Larsen, G. Pasi, D. Ortiz-Arroyo, T. Andreasen, H. Christiansen (Eds.), Flexible Query Answering Systems. XVIII, 714 pages. 2006. (Sublibrary LNAI).

Vol. 4026: P. Gibbons, T. Abdelzaher, J. Aspnes, R. Rao (Eds.), Distributed Computing in Sensor Systems. XIV, 566 pages. 2006.

Vol. 4025: F. Eliassen, A. Montresor (Eds.), Distributed Applications and Interoperable Systems. XI, 355 pages. 2006.

Vol. 4024: S. Donatelli, P. S. Thiagarajan (Eds.), Petri Nets and Other Models of Concurrency - ICATPN 2006. XI, 441 pages. 2006.

Vol. 4021: E. André, L. Dybkjær, W. Minker, H. Neumann, M. Weber (Eds.), Perception and Interactive Technologies. XI, 217 pages. 2006. (Sublibrary LNAI).

Vol. 4020: A. Bredenfeld, A. Jacoff, I. Noda, Y. Takahashi, RoboCup 2005: Robot Soccer World Cup IX. XVII, 727 pages. 2006. (Sublibrary LNAI).

Vol. 4018: V. Wade, H. Ashman, B. Smyth (Eds.), Adaptive Hypermedia and Adaptive Web-Based Systems. XVI, 474 pages. 2006.

Vol. 4016: J.X. Yu, M. Kitsuregawa, H.V. Leong (Eds.), Advances in Web-Age Information Management. XVII, 606 pages. 2006.

Vol. 4013: L. Lamontagne, M. Marchand (Eds.), Advances in Artificial Intelligence. XIII, 564 pages. 2006. (Sublibrary LNAI).

Vol. 4012: T. Washio, A. Sakurai, K. Nakashima, H. Takeda, S. Tojo, M. Yokoo (Eds.), New Frontiers in Artificial Intelligence. XIII, 484 pages. 2006. (Sublibrary LNAI).

Vol. 4011: Y. Sure, J. Domingue (Eds.), The Semantic Web: Research and Applications. XIX, 726 pages. 2006.

Vol. 4010: S. Dunne, B. Stoddart (Eds.), Unifying Theories of Programming. VIII, 257 pages. 2006.

Vol. 4009: M. Lewenstein, G. Valiente (Eds.), Combinatorial Pattern Matching. XII, 414 pages. 2006.

Vol. 4007: C. Àlvarez, M. Serna (Eds.), Experimental Algorithms. XI, 329 pages. 2006.

Vol. 4006: L.M. Pinho, M. González Harbour (Eds.), Reliable Software Technologies – Ada-Europe 2006. XII, 241 pages. 2006.

Vol. 4005: G. Lugosi, H.U. Simon (Eds.), Learning Theory. XI, 656 pages. 2006. (Sublibrary LNAI).

Vol. 4004: S. Vaudenay (Ed.), Advances in Cryptology - EUROCRYPT 2006. XIV, 613 pages. 2006.

Vol. 4003: Y. Koucheryavy, J. Harju, V.B. Iversen (Eds.), Next Generation Teletraffic and Wired/Wireless Advanced Networking. XVI, 582 pages. 2006.

Vol. 4001: E. Dubois, K. Pohl (Eds.), Advanced Information Systems Engineering. XVI, 560 pages. 2006.

Vol. 3999: C. Kop, G. Fliedl, H.C. Mayr, E. Métais (Eds.), Natural Language Processing and Information Systems. XIII, 227 pages. 2006.

Vol. 3998: T. Calamoneri, I. Finocchi, G.F. Italiano (Eds.), Algorithms and Complexity. XII, 394 pages. 2006.

Vol. 3997: W. Grieskamp, C. Weise (Eds.), Formal Approaches to Software Testing. XII, 219 pages. 2006.

Vol. 3996: A. Keller, J.-P. Martin-Flatin (Eds.), Self-Managed Networks, Systems, and Services. X, 185 pages. 2006.

Vol. 3995: G. Müller (Ed.), Emerging Trends in Information and Communication Security. XX, 524 pages. 2006.

Vol. 3994: V.N. Alexandrov, G.D. van Albada, P.M.A. Sloot, J. Dongarra (Eds.), Computational Science – ICCS 2006, Part IV. XXXV, 1096 pages. 2006.

Vol. 3993: V.N. Alexandrov, G.D. van Albada, P.M.A. Sloot, J. Dongarra (Eds.), Computational Science – ICCS 2006, Part III. XXXVI, 1136 pages. 2006.

Vol. 3992: V.N. Alexandrov, G.D. van Albada, P.M.A. Sloot, J. Dongarra (Eds.), Computational Science – ICCS 2006, Part II. XXXV, 1122 pages. 2006.

Vol. 3991: V.N. Alexandrov, G.D. van Albada, P.M.A. Sloot, J. Dongarra (Eds.), Computational Science – ICCS 2006, Part I. LXXXI, 1096 pages. 2006.

Vol. 3990: J. C. Beck, B.M. Smith (Eds.), Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. X, 301 pages. 2006.

Vol. 3989: J. Zhou, M. Yung, F. Bao, Applied Cryptography and Network Security. XIV, 488 pages. 2006.

Vol. 3987: M. Hazas, J. Krumm, T. Strang (Eds.), Location- and Context-Awareness. X, 289 pages. 2006.

Vol. 3986: K. Stølen, W.H. Winsborough, F. Martinelli, F. Massacci (Eds.), Trust Management. XIV, 474 pages. 2006.

Vol. 3984: M. Gavrilova, O. Gervasi, V. Kumar, C.J. K. Tan, D. Taniar, A. Laganà, Y. Mun, H. Choo (Eds.), Computational Science and Its Applications - ICCSA 2006, Part V. XXV, 1045 pages. 2006.

Vol. 3983: M. Gavrilova, O. Gervasi, V. Kumar, C.J. K. Tan, D. Taniar, A. Laganà, Y. Mun, H. Choo (Eds.), Computational Science and Its Applications - ICCSA 2006, Part IV. XXVI, 1191 pages. 2006.

Vol. 3982: M. Gavrilova, O. Gervasi, V. Kumar, C.J. K. Tan, D. Taniar, A. Laganà, Y. Mun, H. Choo (Eds.), Computational Science and Its Applications - ICCSA 2006, Part III. XXV, 1243 pages. 2006.

Vol. 3981: M. Gavrilova, O. Gervasi, V. Kumar, C.J. K. Tan, D. Taniar, A. Laganà, Y. Mun, H. Choo (Eds.), Computational Science and Its Applications - ICCSA 2006, Part II. XXVI, 1255 pages. 2006.

Vol. 3980: M. Gavrilova, O. Gervasi, V. Kumar, C.J. K. Tan, D. Taniar, A. Laganà, Y. Mun, H. Choo (Eds.), Computational Science and Its Applications - ICCSA 2006, Part I. LXXV, 1199 pages. 2006.

Vol. 3979: T.S. Huang, N. Sebe, M.S. Lew, V. Pavlović, M. Kölsch, A. Galata, B. Kisačanin (Eds.), Computer Vision in Human-Computer Interaction. XII, 121 pages. 2006.

Vol. 3978: B. Hnich, M. Carlsson, F. Fages, F. Rossi (Eds.), Recent Advances in Constraints. VIII, 179 pages. 2006. (Sublibrary LNAI).

Vol. 3977: N. Fuhr, M. Lalmas, S. Malik, G. Kazai (Eds.), Advances in XML Information Retrieval and Evaluation. XII, 556 pages. 2006.

Vol. 3976: F. Boavida, T. Plagemann, B. Stiller, C. Westphal, E. Monteiro (Eds.), Networking 2006. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems. XXVI, 1276 pages. 2006.

Vol. 3975: S. Mehrotra, D.D. Zeng, H. Chen, B.M. Thuraisingham, F.-Y. Wang (Eds.), Intelligence and Security Informatics. XXII, 772 pages. 2006.

Vol. 3973: J. Wang, Z. Yi, J.M. Zurada, B.-L. Lu, H. Yin (Eds.), Advances in Neural Networks - ISNN 2006, Part III. XXIX, 1402 pages. 2006.

Vol. 3972: J. Wang, Z. Yi, J.M. Zurada, B.-L. Lu, H. Yin (Eds.), Advances in Neural Networks - ISNN 2006, Part II. XXVII, 1444 pages. 2006.

Vol. 3971: J. Wang, Z. Yi, J.M. Zurada, B.-L. Lu, H. Yin (Eds.), Advances in Neural Networks - ISNN 2006, Part I. LXVII, 1442 pages. 2006.

Vol. 3970: T. Braun, G. Carle, S. Fahmy, Y. Koucheryavy (Eds.), Wired/Wireless Internet Communications. XIV, 350 pages. 2006.

Vol. 3969: Ø. Ytrehus (Ed.), Coding and Cryptography. XI, 443 pages. 2006.

Vol. 3968: K.P. Fishkin, B. Schiele, P. Nixon, A. Quigley (Eds.), Pervasive Computing. XV, 402 pages. 2006.

Vol. 3967: D. Grigoriev, J. Harrison, E.A. Hirsch (Eds.), Computer Science – Theory and Applications. XVI, 684 pages. 2006.

Vol. 3966: Q. Wang, D. Pfahl, D.M. Raffo, P. Wernick (Eds.), Software Process Change. XIV, 356 pages. 2006.

Vol. 3965: M. Bernardo, A. Cimatti (Eds.), Formal Methods for Hardware Verification. VII, 243 pages. 2006.

Vol. 3964: M. Ü. Uyar, A.Y. Duale, M.A. Fecko (Eds.), Testing of Communicating Systems. XI, 373 pages. 2006.

Vol. 3963: O. Dikenelli, M.-P. Gleizes, A. Ricci (Eds.), Engineering Societies in the Agents World VI. XII, 303 pages. 2006. (Sublibrary LNAI).

Vol. 3962: W. IJsselsteijn, Y. de Kort, C. Midden, B. Eggen, E. van den Hoven (Eds.), Persuasive Technology. XII, 216 pages. 2006.

Vol. 3960: R. Vieira, P. Quaresma, M.d.G.V. Nunes, N.J. Mamede, C. Oliveira, M.C. Dias (Eds.), Computational Processing of the Portuguese Language. XII, 274 pages. 2006. (Sublibrary LNAI).

Vol. 3959: J.-Y. Cai, S. B. Cooper, A. Li (Eds.), Theory and Applications of Models of Computation. XV, 794 pages. 2006.

Vol. 3958: M. Yung, Y. Dodis, A. Kiayias, T. Malkin (Eds.), Public Key Cryptography - PKC 2006. XIV, 543 pages. 2006.

¥359.00元

# Preface

The idea to establish a European forum for academic and industrial researchers working on various aspects of performance modeling and analysis of manufactoring and information systems gave rise to an annual series of workshops, referred to as European Performance Engineering Workshop (EPEW). The first two EPEW workshops were held in Toledo, Spain, October 1-2, 2004, and Versailles, France, September 1-3, 2005. This volume contains the proceedings of the third EPEW workshop held at the Technical University of Budapest, Budapest, Hungary, June 21-22, 2006.

These proceedings comprise the 16 accepted contributed papers of EPEW 2006. To ensure the high-quality evaluation of the submitted papers we extended the Program Committee of EPEW 2006 with international experts from all over the world. Each submitted papers went through a rigorous review by at least three international reviewers. Based on the reviews, the subsequent discussions of reviewers with different judgement and an Internet-based Program Committee meeting held on March 30, 2006, we selected 40% of the submitted papers. We therefore owe special thanks to all members of the Program Committee and to all external referees for the excellent work they did for the proper evaluation of the papers.

The final workshop program, as well as this volume, are made up of five thematic sessions:

- Stochastic process algebra
- Workloads and benchmarks
- Theory of stochastic processes
- Formal dependability and performance evaluation
- Queues, theory and practice

These sessions cover a wide range of performance evaluation methods and compose an overview of the current research directions in performance evaluation. Some papers focus on a particular research field (e.g., convergence rate of specific Markov chains) while others provide a combination of research methodologies from essentially different fields (e.g., model checking and stochastic fluid models). We hope that these proceedings offer interesting research results for everyone dealing with performance evaluation.

Last, but not least, we would like to thank the publication chair and the local organizers of the workshops for their work. A special thanks to Levente Bodrog for creating and maintaining the website of the conference.

April 2007

András Horváth<br>
Miklós Telek

# Organization

## Workshop Chair

Miklós Telek, Technical University of Budapest (Hungary)

## Program Chair

András Horváth, University of Torino (Italy)

## Publication Chair

Katinka Wolter, Humboldt University (Germany)

## Local Organizers

Levente Bodrog, Technical University of Budapest (Hungary)
Gábor Horváth, Technical University of Budapest (Hungary)

## Program Committee

Jeremy Bradley
Mario Bravetti
Peter Buchholz
Gianfranco Ciardo
Tadeusz Czachórski
Jean-Michel Fourneau
Reinard German
Stephen Gilmore
Boudewijn Haverkort
Holger Hermanns
András Horváth
Alain Jean-Marie
Carlos Juiz
Helen Karatza
Leïla Kloul
Kim G. Larsen
Fernando Lopez Pelayo
Raymond Marie
Andrew Miner
Manuel Núñez
András Pataricza
Brigitte Plateau
Antonio Puliafito
Marina Ribaudo
Evgenia Smirni
Mark Squillante
Miklós Telek
Nigel Thomas
Katinka Wolter
Wlodek M. Zuberek

## External Referees

Salvador Alcaraz
Ashok Argent-Katwala
Matteo Baldoni
Pieter-Tjerk de Boer
Leonardo Brenner
Katja Gilly
Gábor Horváth
H. Kalatunga
Matthias Kuntz
Jenny Liu
Daniele Manini
Mercedes G. Merayo
Miklós Molnár
Zsolt Pándi
Ismael Rodríguez
Pere Pau Sancho
Ihab Sbeity
Marco Scarpa
Matteo Sereno
Jean-Marc Vincent

# Table of Contents

## Stochastic Process Algebra

## Workloads and Benchmarks

## Theory of Stochastic Processes

## Formal Dependability and Performance Evaluation

## Queues, Theory and Practice

# A Precedence PEPA Model for Performance and Reliability Analysis

Jean-Michel Fourneau and Leïla Kloul

PRiSM, Université de Versailles Saint-Quentin,
45 Av. des Etats Unis, 78000 Versailles, France
{jmf, kle}@prism.uvsq.fr

**Abstract.** We propose new techniques to simplify the computation of the cycle times and the absorption times for a large class of PEPA models. These techniques allow us to simplify the model description to reduce the number of states of the underlying Markov chain. The simplification processes are associated with stochastic comparisons of random variables. Thus the simplified models are stochastic bounds for the original ones.

## 1 Introduction

In the recent years, several researchers have investigated ways to solve steady-state distributions for Stochastic Process Algebra models with exponential duration of activities such as PEPA models [10]. The tensor based representation [11] allows us to build large state spaces in a very efficient manner. However solving the steady-state distribution remains a difficult problem even if the bisimulation technique allows us to reduce the state space. Recently the process algebra formalism has also been used to solve transient problems [7], still under the Markovian assumption.

Here, we advocate a completely different approach which is not totally related to this Markovian assumption. First, we want to compute the distribution of the cycle time (if the model is well defined) or the distribution of the absorption time (if the model has an absorbing state) instead of the steady-state distribution. The cycle time is the delay between two successive visits to a specific state while the absorption time is the time until absorption. Cycle time is closely related to the throughput of the system while the distribution of the absorption time allows us to define the reliability of a system. By taking the average of these distributions, one can obtain the mean throughput and the average population with Little's formula or the mean time to failure. These quantities are in general significant for models based on customer's point of view rather than server's states.

We propose a two-level hierarchical approach. At the higher level, we consider a precedence PEPA model. Each component of the precedence model is a sub-model isolated from the other components. Because of the exponential duration of the activities in a PEPA component, these sub-models can be associated with continuous time Phase type distributions.

A. Horváth and M. Telek (Eds.): EPEW 2006, LNCS 4054, pp. 1–15, 2006.

Computing absorption time distribution is usually done by uniformization and analysis of transient discrete-time Markov chains. This technique requires a large number of vector-matrix multiplications. The matrix size is the number of states in the Markov chain. So it is important to find techniques which can be used to reduce this number of states. Cycle times computation are not necessarily based on Markovian assumption, even if exponential delays of individual activities may lead to the usual Markovian numerical analysis. For a class of decision-free Petri nets, cycle times are defined by recurrence relations [3]. Furthermore these relations are linear but on the max-plus semigroup. Such structures have been studied extensively in the context of random variables (see for instance Baccelli et al [1]). For more general systems, the computation of the cycle times is a complex problem. The stochastic comparison appears to be a promising technique to cope with this complexity.

If we need to compute the cycle time of a PEPA model which is too complex to analyse numerically, we design automatically a new model such that its cycle time is a bound for the exact one. This bound is stochastic: we do not compare reals but distribution functions. Thus stochastic bounds are far more accurate than worst-case analysis. If the new model has a reduced state space, we may then use numerical methods (or even analytical results) to efficiently solve the problem. Note that bounding some performance measures is often sufficient as quite often we only need to verify the requirements in terms of threshold. Stochastic bounds may also be applied to Markov chains (see [8] for a survey of the various techniques involved and [12] for an example of delays due to a Fair Queueing discipline).

Here we propose high level techniques which transform a PEPA model into simpler PEPA model. These techniques are based on stochastic bounds. They allow us to divide the problem into sub-problems or to replace a complete PEPA sub-model by a single activity. Here we just give some theoretical results, we will present in a sequel paper the algorithms we need and some numerical results.

The rest of the paper is organised as follows. In Section 2, we present some concepts of stochastic comparison while Section 3 gives a simple introduction to PEPA, the SPA we consider. Section 4 is devoted to the precedence PEPA model. Section 5 contains the main results of the paper. Finally in Section 6, we conclude our work with some remarks and future work.

## 2 A Simple Introduction to Stochastic Comparison

We restrict ourselves to finite Continuous Time Markov Chains (CTMC). Stoyan [14] defined the strong stochastic ordering ("st" ordering for short) by the set of non-decreasing functions. Bounds on the distribution imply bounds on these functions as well. Important performance measures such as average population, loss rates or tail probabilities are non decreasing functions. The second part of the definition for discrete random variables is much more convenient for an algebraic formulation and an algorithmic setting.

**Definition 1.** *Let $X$ and $Y$ be random variables taking values on a totally ordered space. Then $X$ is said to be less than $Y$ in the strong stochastic sense, that is, $X <_{st} Y$ iff $E[f(X)] \leq E[f(Y)]$ for all non decreasing functions $f$ whenever the expectations exist.*

*If $X$ and $Y$ take values on the finite state space $\{1, 2, \ldots, n\}$ with $p$ and $q$ as probability distribution vectors, then $X$ is said to be less than $Y$ in the strong stochastic sense, that is, $X <_{st} Y$ iff $\sum_{j=k}^{n} p_j \leq \sum_{j=k}^{n} q_j$ for $k = 1, 2, \ldots, n$.*

**Example 1.** *Let $a = (0.1, 0.3, 0.4, 0.2)$ and $b = (0.1, 0.1, 0.5, 0.3)$. We have $a <_{st} b$ as:*

$$\left[\begin{array}{ll} 0.2 & \leq 0.3 \\ 0.2 + 0.4 & \leq 0.3 + 0.5 \\ 0.2 + 0.4 + 0.3 & \leq 0.3 + 0.5 + 0.1 \end{array}\right.$$

Sufficient conditions for comparison for CTMC are known for a long time [14]. The stochastic comparison of CTMC implies that their steady-state and transient distributions are also ordered.

**Theorem 1 (Stoyan [14], page 193).** *Let us consider two CTMC $Z1$ and $Z2$ on the same state space whose transition rate matrix are respectively $Q1$ and $Q2$. If*

1. *$Z1_0 <_{st} Z2_0$*
2. *$\sum_{k \geq l} Q1(i,k) \leq \sum_{k \geq l} Q2(j,k)$ for all $i \leq j$ and for all $l$ which satisfy $l \leq i$ or $l \geq j$.*

*then $Z1 <_{st} Z2$.*

It may be important to compare Phase type random variables with exponential ones because it allows building a smaller Markov chain. Let us first define a family of random variables well known in reliability modelling [4].

**Definition 2 (New Better than Used in Expectation).** *Let $X_t$ be the residual time of $X$, given that $X > t$. $X$ is said to be NBUE if $E(X_t) \leq E(X)$ for all $t$.*

For instance, Erlang, uniform and constant random variables are NBUE. This family leads to another stochastic ordering: the increasing convex ordering which is used to compare random variables with exponentials.

**Definition 3.** *Let $X$ and $Y$ be two random variables on the same space $\epsilon$, $X$ is smaller in increasing convex order than $Y$, if and only if $E(f(X)) \leq E(f(Y))$ for all convex and non decreasing functions $f$ on $\epsilon$, provided that the expectations exist. The relation is denoted by $X <_{icx} Y$.*

**Property 1 ([14]).** *If $X$ is NBUE of mean $m$, then $X$ is smaller in increasing convex ordering than an exponentially distributed random variable of mean $m$.*

The icx ordering also provides a very intuitive lower bound.

**Property 2 ([14]).** *For any arbitrary positive random variable $X$, $E(X) <_{icx} X$.*

We also have two very simple properties which will be used to derive bounds at the higher level of a model from bounds obtained at the lower level.

**Property 3.** *The Max and Plus operators are convex and non decreasing functions.*

**Property 4.** *Let $X$ and $Y$ be two r.v. such that $X <_{icx} Y$, then for all convex and non decreasing function $f$, we have $f(X) <_{icx} f(Y)$.*

Finally, we can compare the absorption time of Markov chains [5] as stated in the following property.

**Property 5.** *Let $Z1$ and $Z2$ be two homogeneous Markov chains with an absorbing state $n$ and let $T_a(Z1)$ and $T_a(Z2)$ denote absorption times for the two chains. If $Z1 <_{st} Z2$ or $Z1 <_{icx} Z2$ then $T_a(Z2) <_{st} T_a(Z1)$.*

Note that the "st" comparison of absorption times is now on random variables $T_a$ defined on the time instants, not on the states.

## 3 PEPA

In PEPA, a system is viewed as a set of *components* which carry out *activities*. Each activity is characterised by an *action type* and a duration which is exponentially distributed. Thus each activity is defined by a couple $(\alpha, r)$ where $\alpha$ is the action type and $r$ is the *activity rate*. Because of the exponential distribution of the activity duration, the underlying Markov process of a PEPA model is a continuous time Markov process.

PEPA formalism provides a set of combinators which allows expressions to be built, defining the behaviour of components, via the activities they engage in. Below, we present informally the combinators we are interested in and which are necessary to our model. For more details about the formalism, see [10].

**Constant:** noted $S \overset{def}{=} P$, it allows us to assign names to components. To component $S$, we assign the behaviour of component $P$.

**Prefix:** noted $(\alpha, r).P$, this combinator is the basic mechanism by which the behaviours of components are constructed. The component carries out activity $(\alpha, r)$ and subsequently behaves as component $P$.

**Choice:** noted $P_1 + P_2$, this combinator represents competition between components. The system may behave either as component $P_1$ or as $P_2$. All current activities of the components are enabled. The first activity to complete, determined by the race condition, distinguishes one of these components, the other is discarded.

**Cooperation:** noted $P_1 \underset{L}{\bowtie} P_2$, it allows the synchronisation of components $P_1$ and $P_2$ over the activities in the cooperation set $L$. Components may proceed independently with activities whose types do not belong to this set. A particular case of the cooperation is when $L = \emptyset$. In this case, components proceed with activities independently and are noted $P_1 || P_2$.

In a cooperation, the rate of a shared activity is defined as the rate of the slowest component. For a component $P_1$ and an action type $\alpha$, the working capacity is termed the *apparent rate* of $\alpha$ in $P_1$. It is the sum of the rates of the $\alpha$ type activities enabled in $P_1$. The apparent rate of $\alpha$ in a cooperation between $P_1$ and $P_2$ over $\alpha$ will be the minimum of the apparent rate of $\alpha$ in $P_1$ and the apparent rate of $\alpha$ in $P_2$.

The rate of an activity may be unspecified for a component and is noted $\top$. Such a component is said to be *passive* with respect to this action type and the rate of this shared activity is defined by the other component in cooperation.

In PEPA, when a component $C$ carries out an activity $(\alpha, r)$ and subsequently behaves as component $C'$, this one is said to be a *derivative* of $C$. From any PEPA component $C$, the *derivative set*, denoted $ds(C)$, is the set of derivatives (behaviours) which can evolve from the component. This set is defined recursively.

The evolution of a PEPA model is governed by the Structured Operational Semantics (SOS) rules of the language [10]. These rules define the admissible transitions or state changes associated with each combinator.

Necessary (but not sufficient) conditions for the ergodicity of the Markov process in terms of the structure of the PEPA model have been identified and can be readily checked [10]. These conditions imply that the model must be a *cyclic* PEPA component. The model should be constructed as a cooperation of *sequential* components, i.e. components constructed using only prefix, choice and constants. This leads to formally define the syntax of PEPA expressions in terms of *model components* $P$ and *sequential components* $S$:

$$P ::= A \mid P \underset{L}{\bowtie} P \mid P/L \qquad\qquad S ::= (\alpha, r).S \mid S + S \mid A_s$$

where $A$ denotes a constant which is either a model or a sequential component and $A_s$ denotes a constant which is a sequential component. Thus the compositional structure of PEPA models is at the level of the cooperating components; such models are considered as *well-defined.*

## 4 The Precedence PEPA Model

We consider that a system is represented by a set of components which have the same general behaviour as they wake up, proceed with their activities and then make other components wake up. The components are assumed to be initially asleep (off) and cannot proceed with the execution of their activities unless they are woken up. We assume a precedence relation between the enabling of the components in the set as the results of some components can be used as an input by other components. The components are labelled to allow a representation of this precedence relation. We assume the following properties for the set of labels:

1. the set is totally ordered,
2. the set has a unique minimal element which is denoted by $Comp_0$ for convenience,
3. and the set has a unique maximal element which is denoted by $Comp_n$.

We assume that $Comp_0$ constitutes the starting component of the system and $Comp_n$ the last one to be enabled. When $Comp_n$ completes, the system is assumed to have the same behaviour, restarting from the beginning, i.e. $Comp_0$ (Figure 1). Furthermore, we assume that the precedence relation between the components is a Directed Acyclic Graph (DAG) modified by this return arc from $Comp_n$ to $Comp_0$.

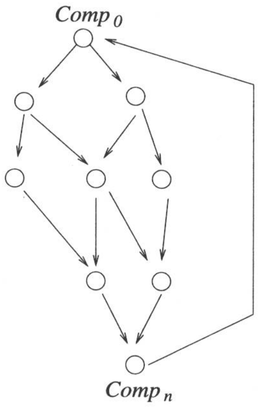

**Fig. 1.** The precedence relation between the components

Our system specifications allow us to consider two kinds of analysis, performance analysis and reliability analysis. The former exploits the presence of the return arc from $Comp_n$ to $Comp_0$ to compute performance measures such as the cycle times. The latter is only possible if we have in our system an absorbing state, that is the precedence relation between the components is a real DAG. Moreover, $Comp_n$ must contain an absorbing state.

### 4.1 Formal Description of the System

To represent the precedence relation characterising our system, we define two families of sets $P_i$ and $S_i$. $P_i$ is the set of components which must complete their activities before $Comp_i$ is woken up and $S_i$ represents the set of components which are enabled when $Comp_i$ has completed its activities. Note that the two families of sets have to be consistent.

We describe the system using $n + 1$ components. Each component $Comp_k$, $k = 0 \ldots n$, is woken up thanks to activity $wake\_up_{jk}$ where $j$ is a predecessor of $Comp_k$, that is $j \in P_k$. Once awake, the component can then proceed with its own activities $\alpha_{k,l}$, $l = 1..m_k$, where $m_k$ is the number of activities of $Comp_k$. Note that these activities $(\alpha_{k,l})$ are all individual activities and once $Comp_k$ has finished executing them, it will wake up the components which are in its set of successors $S_k$.

The behaviour of the components of the system are modelled using the following equations:

$$Comp_0 \stackrel{def}{=} (start, w_1).(\alpha_{0,1}, r_{0,1})...(\alpha_{0,m_0}, r_{0,m_0}). \prod_{i \in S_0} (wake\_up_{0i}, s).Comp_0$$

$$Comp_i \stackrel{def}{=} (wake\_up_{0i}, \top).(\alpha_{i,1}, r_{i,1})...(\alpha_{i,m_i}, r_{i,m_i}). \prod_{k \in S_i} (wake\_up_{ik}, s).Comp_i \qquad \forall i \in S_0$$

$$Comp_k \stackrel{def}{=} \prod_{j \in P_k} (wake\_up_{jk}, \top).(\alpha_{0,k}, r_{0,k})...(\alpha_{k,m_k}, r_{k,m_k}). \prod_{j \in S_k} (wake\_up_{kj}, s).Comp_k \qquad \forall k \in S_i$$

$$Comp_n \stackrel{def}{=} \prod_{j \in P_n} (wake\_up_{jn}, \top).(\alpha_{n,1}, r_{n,1})...(\alpha_{n,m_n}, r_{n,m_n}).(end, w_2).Comp_n$$

where the notation of the form $\prod_{k \in A_i} (\beta_{ik}, r)$ refers to $(\beta_{ii_1}, r).(\beta_{ii_2}, r).\cdots.(\beta_{i_{|A_i|}}, r)$.

The use of $\prod_{k \in S_i} (wake\_up_{ik}, s)$ allows us to model the case where $Comp_i$ wakes up all the components in its successors set one by one. Whereas the use of $\prod_{j \in P_n} (wake\_up_{jn}, \top)$ like in $Comp_n$ models the case where a component has to wait for several predecessors to complete their activities before proceeding with its own activities.

Additionally, we consider another component *Clock*, which allows starting, and restarting the system only once $Comp_n$ has completed its activities. This additional component has to synchronise with $Comp_0$ on activity *start* then on activity *end* with $Comp_n$.

$$\begin{aligned} Clock &\stackrel{def}{=} (start, \top).Clock_0 \\ Clock_0 &\stackrel{def}{=} (end, \top).Clock \end{aligned}$$

The behaviour of the complete system is modelled as the interaction of its components as follows:

$$\begin{aligned} System \stackrel{def}{=} Clock \underset{\{start,end\}}{\bowtie} (\ldots(Comp_0 \underset{\{wake\_up_{0i}/i \in S_0\}}{\bowtie} (\ldots||Comp_i||\ldots)_{i \in S_0}) \\ \underset{\{wake\_up_{ik}/k \in S_i\}}{\bowtie} (\ldots||Comp_k||\ldots)\ldots)_{k \in S_i} \ldots (\ldots||Comp_j||\ldots)_{j \in P_n} \\ \underset{\{wake\_up_{jn}/j \in P_n\}}{\bowtie} Comp_n)\ldots) \end{aligned}$$

### 4.2 Reliability Analysis Using the PEPA Model

Component *Clock* is only necessary in the case where a performance analysis is targeted as it allows modelling the return arc of the precedence relation between the components. Reciprocally, whenever reliability analysis is the objective, component *Clock* is not only unnecessary, but has to be removed from the model. As all its activities have an unspecified rate (⊤), its removal from the model has no impact on the remaining components.