FINAL PROGRAM

Second International Workshop on Software and Performance: WOSP2000

Chateau Laurier Hotel, Ottawa, Canada
September 17 - 20, 2000

Sponsored by ACM SIGMetrics and SIGSoft
In-cooperation with IFIP WG 6.3 and 7.3 and CMG
Assisted by Hewlett-Packard and Nortel Networks

Saturday Sept 16   Welcome

6:00-8:00 PM Registration and Reception, Tudor Room (first floor...this is above the ground floor) (drinks tickets)

Sunday, Sept 17   Tutorials

8:00 on, all day... Registration just outside the Adam Room (ground floor)

Participants are free to attend tutorials in Track A (in the Adam Room) or Track B (in the Canadian Room, on the floor below), as they prefer. Tutorial abstracts are attached at the end of the program.
8:00 Morning coffee, just outside the Adam Room (ground floor).

10:30 to 10:45 Coffee break, just outside the Adam Room.

Tutorial Track A, on UML and Performance, in the Adam Room (ground floor)

Tutorial A1. 9:00 to 12:00
Designing Concurrent, Distributed, and Real-Time Applications with UML
Hassan Gomaa, Department of Information and Software Engineering, George Mason University.

Lunch: 12.30 - 1:30 *Lunch will be provided in the Quebec Suite, first floor*

Tutorial A2. 1:30 to 3:00
Deriving Performance Models from UML Models by Graph Transformations
Dorina C. Petriu, Carleton University, Ottawa, Canada

3:00 to 3:30 Coffee break, just outside the Adam Room.

Tutorial A3. 3:30 to 5:00
UML Performance Data Model
J.-L. Anciano, A. Llamosí, R. Puigjaner, University of the Balearic Islands, Mallorca, Spain.

Tutorial Track B on Performance Modeling and Capacity, in the Canadian Room (lower level, below the ground floor; access via the stairs near the elevators)

Tutorial B1. 9:00 to 12:00
A Practical Approach to Capacity Modeling (**Canadian Room)**
Tim Norton, Simalytic Solutions LLC, Colorado Springs.

Lunch: 12.30 - 1:30 *Lunch will be provided in the Quebec Suite, first floor*

Tutorial B2. 1:30 to 3:00
Reliable software performance modelling using stochastic process algebra (Canadian Room)
Nigel Thomas and Jeremy Bradley, Research Institute in Software Evolution, University of Durham, UK

3:00 to 3:30 Coffee break, just outside the Adam Room.

Tutorial B3. Detection of Reversed Processes and Product Forms in Markovian Process Algebra (Canadian Room)
Peter Harrison, Dept. of Computing, Imperial College of Science, Technology and Medicine, London

Sunday Evening Reception

6:30 - 8:00 Reception and Registration, Quebec Suite (first floor) (drinks tickets)

MONDAY, September 18, 2000

Conference Sessions are in the Laurier Room, on the ground floor at the back (keep going)

Presentations (including brief questions) are timed at: Position papers (*) 10 min, Experience papers (**) 15 min, Full papers (***) 20 min, followed by a general discussion and question period covering the entire session.

7:30 onwards Continental breakfast in the Laurier Room Alcove (ground floor)
8:00 onwards - Registration desk (Laurier Room alcove)

9:00 to 9:15 - Workshop Welcome by Conference Organizers

9:15 to 10:30 - Keynote Address: SPE for Web Applications: New Challenges?
Dr. Connie U. Smith
Performance Engineering Services, Santa Fe, USA

10:30 - 11:00 Coffee break

11:00 to 12:00 Algebraic and Petri-Net Based Approaches to Software Performance
Chair: P. Harrison, Imperial College, London, UK

AEMPA: A Process Algebraic Description Language for the Performance Analysis of Software Architectures (***)
M. Bernardo, Universita di Torino; P. Ciancarini, Universita di Bologna; L. Donatiello, Italy

Quantitative System Evaluation with DSPNexpress 2000 (**)
C. Lindemann, A. Thummler, A. Klemm, M. Lohmann, and O. P. Waldhorst, all of the University of Dortmund, Germany

Visual representation of stochastic process algebra models (*)
N. Thomas, and M. Munro, University of Durham, UK; P. King, and R. Pooley, Heriot Watt University, UK

General Discussion

12:00 to 1:30 Lunch (on your own; the market area is full of restaurants)

1:30 to 3:00 Networks and Mobility
Chair: D. Petriu, Carleton University, Canada

Expressing Meaningful Processing Requirements among Heterogeneous Nodes in an Active Network (***)
V. Galtier, K. L. Mills, Y. Carlinet, S. Leight, and A. Rukhin, National Institute of Standards and Technology, USA

Optimizing Software Performance for IP Frame Reassembly in an Integrated Architecture (***)
P. M. Ewert and N. Manjikian, Queen’s University, Canada

A Performance Monitoring System for Voice-over-IP Gateways (**)
A. E. Conway, Infolibria, Inc., USA

Performance Evaluation of Mobility-based Software Architectures (*)
V. Grassi, Universita di Roma "Tor Vergata", Italy; V. Cortellessa, West Virginia University, USA

General Discussion

3:00 to 3:30 Break

3:30 to 5:15 From Specifications to Performance Models
Chair: H. Gomaa, George Mason University, USA

Deriving Performance Models of Software Architectures from Message Sequence Charts of SA (***)
F. Andolfi, and F. Aquilani, Universita dell'Aquila; S. Balsamo, Universita di Venezia, P. Inverardi, Universita dell'Aquila, Italy

Deriving a Queueing Network based Performance Model from UML Diagrams (***)
V. Cortellesa, West Virginia University, USA
R. Mirandola, Universita di Roma "Torvergata", Italy

A UML Tool for an Automatic Generation of Simulation Programs (**)
L.B. Arief and N. A. Speirs, University of Newcastle upon Tyne, UK

Using UML Models for Performance Calculation (**)
F. Hoeben, Hoeben Software Architecture Consulting, The Netherlands

UML Extensions for the Specification of Latency Constraints in Architectural Models (**)
M. de Miguel, and T. Lambolais, Thomson-LCR; M. Hannouz, SOFTEAM; S. Betge-Brezetz and S. Piekarec, Alcatel-CRC, France

General Discussion

Monday evening: CONFERENCE BANQUET (ticket required)
6:00 to 7:00 Cash bar, Drawing Room Foyer, Chateau Laurier Hotel, ground floor
7:00 to 10:00 Dinner in the Drawing Room

TUESDAY, September 19, 2000

7:30 onwards Continental breakfast in the Laurier Room Alcove (ground floor)
8:00 onwards - Registration desk (Laurier Room alcove)
Sessions in the Laurier Room (ground floor)

8:30 to 10:00   The SPE Process and Infrastructure
Chair: P. Garg, Hewlett Packard, USA

Evaluating the Performance Engineering Process (***)
A. Schmietendorf, A. Scholz, and C. Rautenstrauch, University of Magdeburg, Germany

Design of a Performance Technology Infrastructure to Support the Construction of Responsive Software (***)
E. Papaefstathiou, Microsoft Research Limited, UK

Using Regression Splines for Software Performance Analysis and Software Characterization (***)
M. Courtois and M. Woodside, Carleton University, Canada

Towards a Common Process Model for Systems Development and Performance Engineering (*)
Peter H. Hughes, NTNU, UK

General Discussion

10:00 to 10:30 Break

10:30 to 12:00   Patterns and Anti-Patterns
Chair: K. L. Mills, National Institute of Standards and Technology, USA

Design and Performance Modeling of Component Interconnection Patterns for Distributed Software Architectures (***)
H. Gomaa and D. Menasce, George Mason University, USA

Software Performance Antipatterns (***)
Connie Smith, Performance Engineering Services, USA
Lloyd Williams, Software Engineering Research, USA

A Pattern-Based Approach to Model Software Performance (**)
J. Merseguer, J. Campos, and E. Mena, University of Zaragoza, Spain

Design Patterns in Performance Prediction (*)
A. I. Verkamo, J. Gustafsson, L. Nenonen, and J. Paakki, University of Helsinki, Finland

How System Architectures Impede Interoperability (*)
L. A. Davis, J. Payton, and R. Gamble, University of Tulsa, USA

General Discussion

12:00 to 1:30 Lunch (on your own; the market area is full of restaurants)

1:30 to 3:00 Panel and discussion: Architecture and Performance
Panelists: H.Gomaa (Chair), P. Harrison, P. Inverardi, K. Mills, D. Petriu

The quantitative analysis of a software system architecture allows the early detection of potential performance problems. The analysis is for the software architecture conceptually executing on a given hardware configuration with a given external workload applied to it. Early detection of potential performance problems allows alternative softwaredesigns and hardware configurations to be investigated.

This panel session interprets architecture in the broad sense, meaning designing a software system and analyzing its performance before the system is implemented, including the performance analysis of UML based designs. The goal is to have an iterative process whereby the performance analysis results in a software architecture that developers have confidence will meet its performance goals.

Members of the panel will briefly state their position on how to achieve such an iterative design and performance analysis process. The panel will then be opened up to a general discussion with audience participation.

3:00 to 3:30 Break

3:30 to 5:30 Middleware, Web Site, Transaction Processing Systems, and Server Cluster Performance
Chair: D. Menasce, George Mason University, USA

Analytic Modeling of Load Balancing Policies for Tasks with Heavy-tailed Distributions (***)
A. Riska, E. Smirni, G. Ciardo, College of William and Mary, USA

An Analytic Model of Web Servers in Distributed Computing Environments (***)
P. Reeser, AT&T, USA

Performance Analysis of a Transaction Based Software System with Shutdown (***)
H. Okamura, T. Dohi, and S. Osaki, Hiroshima University, Japan

Performance Evaluation of an Enterprise JavaBean Server Implementation (***)
C. Llado and P. Harrison, Imperial College, UK

Using Analytic Models for Predicting Middleware Performance (**)
D. Petriu, Carleton University; H. Amer, Nortel Networks; S. Majumdar, Carleton University; I. Abdull-Fatah, Nortel Networks, Canada

General Discussion

WEDNESDAY, September 20, 2000

7:30 onwards: Continental breakfast in the Laurier Room Alcove (ground floor)

8:30 to 10:00 Panel and Discussion: Burning Issues in Software and Performance
Panelists: M. Woodside (Chair), P. Garg, D Menasce, E. Parsons, C. Smith

This panel session will debate issues that are considered crucial to the field of software performance, including an assessment of the present state-of-the-art, recommended future directions, and how WOSP can contribute to furthering these goals.

Members of the panel will briefly state their views on the Burning Issues in Software and Performance. The panel will then be opened up to a general discussion with audience participation.

10:00 to 10:30 Coffee break

10:30 to 11:45 Scalability and Program Performance Prediction
Chair: P. Hughes, Modicum, UK

Characteristics of Scalability and Their Impact on Performance (***)
A. Bondi, AT&T Labs, USA

Execution-Driven Performance Analysis for Distributed and Parallel Systems (***)
T. Fahringer, University of Vienna; B. Scholz, Vienna University of Technology, Austria; X. Sun, Illinois Institute of Technology, USA

Static Performance Prediction of Data-Dependent Programs (***)
H. Gautama and A. J. C. van Gemund,
Delft University of Technology, The Netherlands

General Discussion

11:45 to 12:00 Concluding Remarks

TUTORIAL ABSTRACTS

Tutorial A1. 9:00 to 12:00 Designing Concurrent, Distributed, and Real-Time Applications with UML
Hassan Gomaa, Department of Information and Software Engineering, George Mason University.

Tutorial Outline: Introduction to Software Design. Software design concepts: object-oriented concepts, information hiding, classes, inheritance. Concurrent tasking concepts. Overview of UML notation. Object-Oriented Software Life Cycle; Object-Oriented Requirements Modeling; Object-Oriented Analysis Modeling, Object-Oriented Design Modeling. Incremental software construction and integration.

Requirements modeling. The use case modeling. Use cases, actors, use case relationships. Use case descriptions. Special considerations for real-time and distributed applications. Analysis modeling. Static modeling. Objects, classes, and relationships. Static modeling of problem domain.

Object and class structuring. Object structuring; class categorization using stereotypes. Finite state machines. State transition diagrams. Hierarchical statecharts. Developing statecharts from state dependent use cases. Dynamic modeling. Object interaction, message sequencing. Dynamic analysis - from the use case model to the object interaction model.

Design modeling for real-time and distributed applications: Subsystem Architecture Design. Developing the overall software architecture from the analysis model. Separation of concerns in subsystem design. Subsystem structuring. Design of Distributed Applications. Distributed subsystem structuring criteria. Defining subsystem interfaces - synchronous, asynchronous, brokered, and group communication.

Client/server applications. Subsystem design. Distributed system configuration. Concurrent Task Structuring. Task structuring criteria. Mapping the analysis model to a task architecture. Task Interfaces - message communication, event synchronization, information hiding classes. Task behavior specifications. Information hiding class design. Refining the static model. Designing class operations. Use of inheritance in software design. Class interface specifications. Detailed concurrent design. Integrating tasks and information hiding object. Access synchronization. Design of connector classes. Performance Analysis of Real-Time Designs. Real-time scheduling; rate monotonic analysis, scheduling of periodic and aperiodic tasks, priority ceiling protocol; event sequence analysis; application to real world problems.

The speaker: Hassan Gomaa is Professor in the Department of Information and Software Engineering at George Mason University. He has over 25 years experience in software engineering, both in industry and academia, and has published over 90 technical papers and two text books. His latest book is "Designing Concurrent, Distributed, and Real-Time Applications with UML", published by Addison Wesley in August 2000 and upon which this tutorial is based.

He has developed real-time and distributed applications in industry, developed software design methods and applied them to real-world problems, and taught short courses on the subject for professional software engineers. He received his Ph.D. in Computer Science from Imperial College, London.

Tutorial A2. 1:30 to 3:00 Deriving Performance Models from UML Models by Graph Transformations

Dorina C. Petriu, Carleton University, Ottawa, Canada

Abstract: The Unified Modeling Language (UML) is the industry-standard language for specifying, visualizing, constructing and documenting software designs. The tutorial presents current research that uses graph transformations based on the graph-rewriting formalism to derive a Layered Queueing Network (LQN) performance model from a UML model.

Both the UML and LQN models have an underlying graphical representation. A "graph rewriting system" contains a schema that defines a class of graphs, and a set of rules that transforms a given graph instance (i.e., the input graph) into another graph instance (i.e., the output graph). In our case, the graph schema contains two parts: one describes the UML model and is derived from the UML meta-model, whereas the other describes the LQN model.

The transformation from UML to LQN follows the general Software Performance Engineering approach: first a software execution model is built, then it is mapped to a system performance model. In this research, the software execution model is represented as UML activity diagrams describing the most important use cases. The activity diagrams may be either contained in the UML model, or derived by graph transformations from interaction (sequence and/or collaboration) diagrams. Also, UML component and deployment diagrams are used to generate the LQN nodes that represent software components and hardware devices. The activity diagrams are automatically partitioned in swimlanes corresponding to different software components responsible for various activities. In order to complete the mapping from UML to LQN, the user must provide additional information regarding the resource demands per activity (such as CPU execution times). These values are used to generate the parameters for the LQN model by aggregating the resource demands for each activity diagram partition.

The speaker: Dorina Petriu is an Associate Professor in the Department of Systems and Computer Engineering at Carleton University, Ottawa, Canada. She received the Dipl. Eng. degree in computer engineering from Polytechnic University of Timisoara, Romania, and the Ph.D. degree in electrical engineering from Carleton University, Ottawa. Her research interests are in the areas of performance modelling and software engineering, with emphasis on integrating performance analysis techniques into the software development process. Current research projects include the automatic derivation of software performance models from UML design specifications for early performance predictions, and scalability analysis of Virtual Private Networks applications. D. Petriu is a member of the I.E.E.E and A.C.M. She was the chair of IEEE Ottawa Computer Chapter for five years, and in 1998 received the award "In Recognition of Outstanding Contributions to the IEEE Ottawa Section".

Tutorial A3. 3:30 to 5:00 UML Performance Data Model

J.-L. Anciano, A. Llamosí, R. Puigjaner, University of the Balearic Islands, Mallorca, Spain.

Abstract: The purpose of this tutorial is to propose a data model that allows the elaboration of performance models for real-time systems. This means the characterisation of the data about a system that is required for specifying its temporal behaviour and resource constraints, as well as for analysing and predicting whether a proposed design may satisfy the specified requirements for that system.

Although the scope of systems that have been considered is not universal, the most common architectures for real-time systems have been taken into account. This tutorial is addressed to only ever lasting systems. But this is the case in most real-time systems, from traffic controllers to bank telling machines. However, we cope with implementations based both in permanent cyclic tasks (hard real-time systems) and in transaction monitors (soft real-time systems) that will create a new terminating task for each request.

The resulting data model is quite complex but it is described step to step, showing in isolation, the different scopes of the proposed model, together with their rationale. The next steps until arriving to the performance models are also suggested.

Tutorial B1. 9:00 to 12:00

A Practical Approach to Capacity Modeling

Tim Norton, Simalytic Solutions LLC, Colorado Springs.

Abstract: Have you ever spent days or weeks modeling a system only to find no one wants to read the results? Does your boss think modeling takes too long and doesn't provide anything useful? Do you think the 'right' modeling tool will fix your problems if you can just find it? A Practical Approach to Capacity Modeling takes a step back and looks at modeling as a business problem to be solved with technical tools rather than a technical problem. Several application modeling approaches are presented with the focus on predicting the required computer system capacity based on business projections. The techniques presented use tools readily available to the attendees to solve significant business problems.

The tutorial outlines a spiral approach to any modeling effort that starts with the simplest of tools. A distinction is made between an isolated modeling effort and establishing an organization to provide modeling services. A few classic failures (modeling efforts that got the right answer, but failed) will be presented and discussed.

The tutorial concludes with an open discussion of where modeling is going, some of the problems modeling new applications (client/server, CORBA, and the Web), a brief overview of new techniques such as Simalytic Modeling and Business Simulation (System Dynamics) and some of the issues for validating models with end-to-end response time metrics.

Tutorial agenda: - How do you identify the problem? - Where to focus the modeling effort? - Tools: Start simple and work up. - One shot effort or service provider? - Failures: What can we learn? - Futures: Where does modeling go from here?

The speaker: Dr. Norton is employed as a Senior Staff Member of Technology Planning at MCI Worldcom in Colorado Springs, CO. He also does consulting in the area of application modeling with Simalytic Solutions, LLC. In addition, he is a adjunct professor of Computer Science at Colorado Technical University. His research is focused on hybrid modeling of client/server systems. He has worked with a variety of computer systems and has 25+ years MVS, PC and Distributed Systems experience, including applications design, support, and systems programming. He has been doing capacity planning and modeling for almost 15 years at a variety of companies, including Covia/United Airlines, ARCO Oil and Gas, Colorado Interstate Gas, and Texas Instruments. He holds a Master's Degree in Computer Science and a Doctor of Computer Science Degree. He is a member of the Association for Computing Machinery, the Society for Computer Simulation, the IEEE Computer Society, and the Computer Measurement Group.

Tutorial B2. 1:30 to 3:00

Reliable software performance modelling using stochastic process algebra

Nigel Thomas and Jeremy Bradley, Research Institute in Software Evolution, University of Durham, UK

Abstract: The aim of any modelling exercise is to develop a better understanding of the system that is being studied, however it is unfortunately not always easy to understand abstract performance models or the metrics that are derived from them. In this tutorial we focus on the facilitation of more reliable performance prediction through better understanding of models and solutions in the context of a stochastic process algebra. The tutorial will cover a basic introduction to performance modelling with stochastic process algebra, aimed at experienced performance engineers with little or no previous knowledge of stochastic process algebra. We will use as many

examples as possible and cover translation of models from stochastic Petri nets and queueing networks to stochastic process algebra. Visual techniques from program comprehension are applied to performance models to aid understanding and improve confidence that the model accurately reflects design. Some visual representations are also used to derive design views, such as UML collaboration diagrams, to provide meaningful feedback in a modelling cycle. As well as being confident that the model is accurate, the designer must also have confidence in the measures derived. Partly this is a matter of ensuring that the correct measures are derived, but also that those measures are sufficiently bounded to be applicable.

The speakers: Dr Nigel Thomas is a lecture in Computer Science at the University of Durham. He holds a PhD from the University of Newcastle-upon-Tyne and has published a number of papers on different aspects of modelling with stochastic process algebra. Recently he has been working on visualising performance models and considering non-product form model decomposition.

Dr Jeremy Bradley is a research assistant at the University of Durham working on the EPSRC funded QWiD project investigating considering non-product form model decomposition. He recently completed his PhD at the University of Bristol in reliable performance modelling

Tutorial B3. Reversed Agents in Markovian Process Algebra

Peter Harrison, Dept. of Computing, Imperial College of Science, Technology and Medicine, London

Abstract: Stochastic process algebra (SPA) is a formalism developed over the last decade that can describe rigorously both the qualitative (functional) and quantitative (performance-related) behaviour of systems of interacting processes. The principal advantage of this algebraic approach to modelling is the property of compositionality possessed by all SPAs. This means that two or more fully specified systems can be combined together (as subsystems) into a more complex system in a simple way|both syntactically and semantically. The behaviour of the subsystems is not affected, except where they are explicitly connected to each other. Compositionality allows the properties of complex systems to be analysed in a hierarchical, inductive way and offers the prospect of efficient performance models, although the latter does not come automatically. A new approach to deriving efficient, product-form solutions in Markovian process algebra (MPA) using properties of reversed processes will be presented. The compositionality of MPAs is directly exploited, allowing a large class of hierarchically constructed systems to be solved for their state probabilities at equilibrium. New results on both reversed stationary Markov processes and MPA itself are included, resulting in a mechanisable proof in MPA notation of Jackson's theorem for product-form queueing networks. Several examples are used to illustrate the approach.

The speaker: Peter Harrison is currently a Professor of Computing Science at Imperial College, London where he became a lecturer in 1983. He graduated at Christ's College Cambridge as a Wrangler in Mathematics in 1972 and went on to gain Distinction in Part III of the Mathematical Tripos in 1979, winning the Mayhew prize for Applied Mathematics. He obtained his Ph.D. in Computing Science at Imperial College in 1979. He has researched into analytical performance modelling techniques and algebraic program transformation for some twenty years, visiting IBM Research Centers for two summers in the last decade. He has written two books, had over 100 research papers published in his research areas and held a series of research grants, both national and international. The results of his research have been exploited extensively in industry, forming an integral part of commercial products such as Metron's Athene Client-Server capacity planning tool. He has taught a range of subjects at undergraduate and graduate level, including Operating

Systems: Theory and Practice, Functional Programming, Parallel Algorithms and Performance Analysis.

WOSP2000 CONFERENCE COMMITTEE

General Chair: Murray Woodside, Carleton University (including registration and publicity)
Program Co-Chairs: Daniel Menasce, Hassan Gomaa, both of George Mason University
Tutorials Chair: Rob Pooley (Herriot-Watt University, Edinburg, Scotland)
Treasurer: Andre Bondi (ATT)

PROGRAM COMMITTEE

Heinz Beilner, Univ. of Dortmund
Gianfranco Ciardo, College of William and Mary
Rich Friedrich, Hewlett-Packard Labs
Pankaj Garg, Hewlett-Packard Labs
Guenter Haring, Univ. of Vienna
Peter Harrison, Imperial College
Peter Hughes, Modicum
Paola Inverardi, Univ. dell'Aquila
Ravi Iyer, Univ. of Illinois
Anant Jalnapurkar, Nortel
Mark Klein, SEI
John Knight, University of Virginia
Paul Kogut, Lockheed-Martin
Doug Lea, SUNY
Ross A. Lumley, Mitretek and GWU
Jeff Magee, Imperial College
Shikharesh Majumdar,Carleton University
Kevin Mills, NIST
Richard Muntz, UCLA
Rob Nord, Siemens
Harry Perros, North Carolina State
Dorina Petriu, Carleton University
Mauro Pezze, Politechnico di Milano
Rob Pooley, Herriot-Watt University
Jerry Rolia, Carleton University
Bill Sanders, Univ. of Illinois
Giuseppe Serazzi, Politecnico di Milano
Ken Sevcik, Univ. of Toronto
Connie Smith, Performance Engineering Services
Alex Thomasian, U. Connecticut
Lloyd Williams, Software Engineering Research
Dave Zubrow, SEI

WOSP Steering Committee

Guenter Haring, Austria,
Danny Menasce, USA,
Gail Murphy, Canada,
Dorina Petriu, Canada,
Rob Pooley, UK,
Connie Smith, USA,
Elaine Weyuker, USA
Murray Woodside, Canada,
Dave Zubrow, USA.