The South African Institute for Computer Scientists and
Information Technologists

ANNUAL RESEARCH AND DEVELOPMENT
SYMPOSIUM

23-24 NOVEMBER 1998
CAPE TOWN
Van Riebeek Hotel in Gordons Bay

Hosted by the University of Cape Town in association with the CSSA,
Fort Hare/University of Zululand and
The University of Natal

PROCEEDINGS

EDITED BY
D. PEIKOV AND L. VENTER

SPONSORED BY

ABSA Group
The South African Institute for Computer Scientists and Information Technologists

ANNUAL RESEARCH AND DEVELOPMENT SYMPOSIUM

23-24 NOVEMBER 1998
CAPE TOWN
Van Riebeeck hotel in Gordons Bay

Hosted by the University of Cape Town in association with the CSSA,
Potchefstroom University for CHE and
The University of Natal

GENERAL CHAIR: PROF G. HATTINGH, PU CHE
PROGRAMME CO-CHAIRS:
PROF. L VENTER, PU CHE (Vaal Triangle), PROF. D. PETKOV, UN-PMB
LOCAL ORGANISING CHAIR: PROF. P. LICKER, UCT - IS

PROCEEDINGS

EDITED BY
D. PETKOV AND L. VENTER

SYMPOSIUM THEME:
Development of a quality academic CS/IS infrastructure in South Africa

SPONSORED BY
FOREWORD

The South African Institute for Computer Scientists and Information Technologists (SAICSIT) promotes the cooperation of academics and industry in the area of research and development in Computer Science, Information Systems and Technology and Software Engineering. The culmination of its activities throughout the year is the annual research symposium. This book is a collection of papers presented at the 1998 such event taking place on the 23rd and 24th of November in Gordons Bay, Cape Town. The Conference is hosted by the Department of Information Systems, University of Cape Town in cooperation with the Department of Computer Science, Potchefstroom University for CHE and and Department of Computer Science and Information Systems of the University of Natal, Pietermaritzburg.

There are a total of 46 papers. The speakers represent practitioners and academics from all the major Universities and Technikons in the country. The number of industry based authors has increased compared to previous years.

We would like to express our gratitude to the referees and the paper contributors for their hard work on the papers included in this volume. The Organising and Programme Committees would like to thank the keynote speaker, Prof M.C Jackson, Dean, University of Lincolnshire and Humberside, United Kingdom, President of the International Federation for Systems Research as well as the Computer Society of South Africa and The University of Cape Town for the cooperation as well as the management and staff of the Potchefstroom University for CHE and the University of Natal for their support and for making this event a success.

Giel Hattingh, Paul Licker, Lucas Venter and Don Petkov
Table of Contents

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lynette Drevin: Activities of IFIP wg 11.8 (computer security education) &amp; IT related ethics education in Southern Africa</td>
<td>1</td>
</tr>
<tr>
<td>Reinhardt A. Botha and Jan H.P. Elof: exA Security Interpretation of the Workflow Reference Model</td>
<td>3</td>
</tr>
<tr>
<td>Willem Krige and Rossouw von Solms: Effective information security monitoring using data logs</td>
<td>9</td>
</tr>
<tr>
<td>Carl Papenfus and Reinhardt A. Botha: A shell-based approach to information security</td>
<td>15</td>
</tr>
<tr>
<td>Walter Smuts: A 6-Dimensional Security Classification for Information</td>
<td>20</td>
</tr>
<tr>
<td>Philip Machanick and Pierre Salverda: Implications of emerging DRAM technologies for the RAM page Memory hierarchy</td>
<td>27</td>
</tr>
<tr>
<td>Susan Brown: Practical Experience in Running a Virtual Class to Facilitate On-Campus Under Graduate Teaching</td>
<td>41</td>
</tr>
<tr>
<td>H.D. Masethe, T.A Dandadzi: Quality Academic Development of CS/IS Infrastructure in South Africa</td>
<td>49</td>
</tr>
<tr>
<td>Philip Machanick: The Skills Hierarchy and Curriculum</td>
<td>54</td>
</tr>
<tr>
<td>Theda Thomas: Handling diversity in Information Systems and Computer Science Students: A social Constructivist Perspective</td>
<td>63</td>
</tr>
<tr>
<td>Udo Averweg and G J Erwin: Critical success factors for implementation of Decision support systems</td>
<td>70</td>
</tr>
<tr>
<td>Magda Huisman: A conceptual model for the adoption and use of case technology</td>
<td>78</td>
</tr>
<tr>
<td>Paul S. Licker: A Framework for Information Systems and National Development Research</td>
<td>79</td>
</tr>
<tr>
<td>K. Niki Kunene and Don Petkov: On problem structuring in an Electronic Brainstorming (EBS) environment</td>
<td>89</td>
</tr>
</tbody>
</table>
Derek Smith: Characteristics of high-performing Information Systems Project Managers and Project Teams

Lucas Venêr: INSTAP: Experiences in building a multimedia application

Scott Hazelhurst, Anton Fatti, and Andrew Henwood: Binary Decision Diagram Representations of Firewall and Router Access Lists

Andre Joubert and Annelie Jordaan: Hardware System interfacing with Delphi 3 to achieve quality academic integration between the fields of Computer Systems and Software Engineering

Borislav Roussev: Experience with Java in an Advanced Operating Systems Module

Conrad Mueller: A Static Programming Paradigm

Sipho Langa: Management Aspects of Client/Server Computing

T Nepal and T Andrew: An Integrated Research Programme in AI applied to Telecommunications at ML Sultan Technikon

Yuri Velinov: Electronic lectures for the mathematical subjects in Computer Science

Philip Machanick: Disk delay lines

D Petkov and O Petkova: One way to make better decisions related to IT Outsourcing

Jay van Zyl: Quality Learning, Learning Quality

Matthew O Adigun: A Case for Reuse Technology as a CS/IS Training Infrastructure

Andy Bytheway and Grant Hearn: Academic CS/IS Infrastructure in South Africa: An exploratory stakeholder perspective

Chantel van Niekerk: The Academic Institution and Software Vendor Partnership

Christopher Chalmers: Quality aspects of the development of a rule-based architecture

Rudi Harmse: Managing large programming classes using computer mediated communication and cognitive modelling techniques
Michael Muller: How to gain Quality when developing a Repository Driven User Interface

Elsabe Cloete and Lucas Venter: Reducing Fractal Encoding Complexities

Jean Bilbrough and Ian Sanders: Partial Edge Visibility in Linear Time

Philip Machanick: Design of a scalable Video on Demand architecture

Freddie Janssen: Quality considerations of Real Time access to Multidimensional Matrices

Machiel Kruger and Giel Hattingh: A Partitioning Scheme for Solving the Exact k-item 0-1 Knapsack Problem

Ian Sanders: Non-orthogonal Ray Guarding

Fanie Terblanche and Giel Hattingh: Response surface analysis as a technique for the visualization of linear models and data

Olga Petkova and Dewald Roode: A pluralist systemic framework for the evaluation of factors affecting software development productivity

Peter Warren and Marcel Viljoen: Design patterns for user interfaces

Andre de Waal and Giel Hattingh: Refuting conjectures in first order theories

Edna Randiki: Error analysis in Selected Medical Devices and Information Systems

184

193

200

211

218

229

230

236

243

252

261

262
Experience with Java in an Advanced Operating Systems Module

Borislav Roussev  
Computer Science Department  
University of the Witwatersrand, Johannesburg  
South Africa  
(borislav@cs.wits.ac.za)

Abstract

The Advanced Operating Systems (AOS) module is one of the courses offered by the Computer Science Department in its honours program. The module aims are twofold: to enhance students' knowledge in single processor operating systems and to introduce the fundamental concepts of distributed operating systems. The workload consists of nineteen hours of work delivered over a 6 weeks period. Contact time is four lecture hours reinforced by a one-hour lab session; the remaining time is private study. Material pertinent to the course is available on-line at an internal departmental web site (http://www.cs.wits.ac.za/~borislav/courses/aos.html). Besides the prescribed textbook each student had to read carefully and critically a core set of twenty papers and to give at least two presentations. All papers are available from the course web site. The students had had no experience in networking.

In the previous version of the module C/C++ was used as the implementation language. The assignment tests showed clearly that except for the brightest students everyone else struggled to implement even simple tasks such as process synchronization, signal handling, client/server systems, and lightweight threads. Students acquired only theoretical understanding of remote procedure calling, group communication, and distributed computing as a whole. It was not possible to reinforce this material with lab exercises. In conclusion it became clear that the laboratory exercises were not achieving their aim of reinforcing the material presented in class.

Fortunately, all features required for the AOS laboratory exercises: UDP sockets, TCP sockets, multicast sockets, multithreading, thread synchronization, platform independence, and object persistence and serialization are part of the Java programming language. The simple and elegant interface to most of Java API made it a natural choice for the course. The major disadvantage of using Java instead of C/C++ was that the students had to learn the language and the extensive library by themselves.

All students managed the initial exercises and were able to write and test a simple multithreaded concurrent server using a pool of worker threads and monitors to synchronize the access to the shared resources. Then the students successfully implemented a communication protocol of moderate complexity. Part of the exercise involved object serialization. The final project of the course was to implement remote method invocation. The students had to write: (1) a portmapper (binder), providing the following services: register, whereis, withdraw, and dump; (2) a client stub and a server stub exporting the following API: listening on an advertised port (the port registered with the portmapper), extracting the control information from a client or a server message, marshalling and unmarshalling parameters and results; (3) a server providing the remote services using a pool of worker threads and appropriate synchronization mechanisms; (4) finally, an application making use of the former three modules. The students were divided into groups of three. All groups managed to complete their projects on time. The switch from C/C++ to Java seems to pass almost unnoticed. Feedback at the conclusion of the module was very positive of the use of Java on this course.

In conclusion, Java proved to be an ideal language for the use in the AOS course. The simplicity of its code and API helped the students to grasp the important concepts quickly without wasting time in technical details. The course ran more smoothly than the previous year when C/C++ was used.