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,
Forchepstream University for CHE and
The University of Natal

PROCEEDINGS

EDITED BY
D. Petkov 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 Venter**: INST AP: 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
Abstract

This paper is intended as an opinion paper regarding information security concerns in the Workflow Reference Model (WfRM) as defined by the Workflow Management Coalition (WfMC). After an introduction into the workflow environment, the WfRM is described. The security services, identification and authentication, authorization, confidentiality, integrity and non-repudiation are briefly defined in order to serve as a framework for further discussion. The main functional areas in the model are discussed in terms of the mentioned security services. The paper concludes by identifying areas for further research.

1. INTRODUCTION

The large number of commercial workflow products that have appeared in the last few years are proof of the increasing interest in workflow technology. Although workflow is in no way ubiquitous yet, many organizations view it as a crucial technology to get a strategic advantage [3, 2].

Within the small to medium size enterprises the true value of workflow will only be realized if the workflow process could be extended to include several of the business partners, suppliers and customers, thereby forming virtual companies. However, as pointed out by [3], current products incorporate different and very concrete interpretations of the real world, thus making it very difficult or practically impossible to federate various systems.

The Workflow Management Coalition (WfMC) [6] is a grouping of companies trying to establish standards that will facilitate the interoperability between workflow systems. The WfMC recognizes the fact that workflow management systems all share certain common characteristics, thus enabling them to potentially achieve a level of interoperability through the use of common standards for various functions.

Authors in the field of workflow management are referring to the WfRM (see for example [1],[3],[5] and [8]), but no references that put emphasis on the relevant security issues could be found.

2. THE WORKFLOW REFERENCE MODEL

The Workflow Reference Model is the result of an effort to standardize workflow management products. It describes the basic concepts of workflow management, a reference architecture and interfaces between the architecture components. This article only discusses the basics of the WfRM, please refer to the WfMC documents (as indicated in the references) for more detailed information.

2.1 Terminology

Workflow is concerned with the facilitation or automation of a business process, either in part or whole [6]. The business process, what is intended to happen, is defined in a process definition. The process definition is a representation of what should happen and can be modeled as consisting of activities (manual or automated) and/or sub-processes.

The Workflow Management System controls the execution of the workflow via a process instance created from a process definition. The process instance includes work items (tasks allocated to participants)
and/or invoked applications (computer tools or applications used to support an activity). The work items are communicated to the responsible user(s) through an interface accessing a worklist.

Interoperability between different workflow environments is dependent on adherence to a general architecture. In the WfRM this general architecture is defined in terms of three functional areas consisting of different components, as well as the interfaces between these functional components.

The following sub-section is concerned with describing the three functional areas that form part of the WfRM.

2.2 Functional areas

The build-time functions are concerned with the definition of a real-world business process into a formal, computer processable definition through the use of various modelling and analysis techniques.

The run-time process control functions are concerned with the interpretation of the process definition and the creation and control of the individual process instances. The responsibility for scheduling the various activity instances that form part of the process instances, as well as other operational issues are also the concern of the run-time control functions. The workflow engine largely represents the run-time control functions.

The run-time interactions with human users and IT application tools represent another functional area. Interaction with the process control software is needed in order to ascertain the status of processes, to invoke application tools and to pass the appropriate data.

These three levels of functionality are realized in the WfRM through various components. Figure 1 gives an overview of the components as defined in the WfRM, as well as how they interface. The next section defines the interfaces in more detail.

2.3 The WfRM-Interfaces

The WfRM distinguishes between five different interface areas. These interfaces (indicated in Figure 1) are specified as functions with input and output parameters as well as a return type and include:

- Interface 1 specifies the interface for the specification of process data and its interchange.
- Interface 2 stipulates interfaces to support interaction with user interface desktop functions.
- Interface 3 defines interfaces to support interaction with various application types.
- Interface 4 is concerned with the interoperability between workflow systems.
3. SECURITY SERVICES

This section will give brief definitions of the security services needed to secure a computing environment.

All of the services are reliant on the identification of user identities. The *identification and authentication service* is responsible for confirming the claimed identity of a user. *Authorization* mechanisms are responsible for controlling access rights of users, i.e. who can do what with which data. *Confidentiality* related services are responsible for the non-disclosure of information to unauthorized parties. *Integrity* services are responsible for keeping the information in a sound state. *Non-repudiation* services are concerned with preventing denial of service to properly authenticated and authorized users.

The rest of this paper discusses the mentioned security services with reference to their applicability in the WfRM.

4. SECURITY AND THE WORKFLOW REFERENCE MODEL

This section identifies some of the security issues surrounding the WfRM, in an attempt to highlight the need to consider information security when designing a system according to the WfRM. The arguments are organized according to the functional areas identified within the model.

4.1 Build-time functions

The WfRM refers to organizational role based data when establishing the process definition. The role-based access control mechanism, as traditionally used within the database environment, suffers from a lack of fluidity which is very necessary in a workflow environment. Access rights and rules must be defined at design time of the process according to an access control policy. The access control policy must support the level of confidentiality required in the environment, i.e. how strict the need-to-know policy must be enforced.

Businesses and their way of doing business change almost continuously to keep up with the agile demands of the market. This has the effect that process definition must be revisited from time to time. Build-time tools must support this evolution of workflow processes in one, or both, of two ways. Firstly changes might need to be immediate, i.e. a change in the execution of currently active process instances, or it could be delayed, i.e. the new process definition that is only used in new process instances. Particular care should be exercised to ensure information integrity throughout this process. It is therefore also important that the authorization functionality of the build-time functions supports this notion.

4.2 Run-time functions

The workflow enactment service forms the run-time core of a workflow environment. It is responsible for "routing" the information needed to perform a task between the participants in the workflow. The WfRM does stipulate expected (or typical) functionality for the workflow engine. However, issues regarding security are left out of the model, except for identifying that the workflow engine may use role based organizational data to determine recipients of messages. The same concerns as identified for the build-time functions still exist.

A proper access control mechanism needs to be designed to ensure that a need-to-know policy can be enforced. The rights of any individual should not be fixed. This means that if a user needs to change a document as part of a specific process instance it should be allowed, whereas if that same person is to try and alter the same document in another process instance, where it is only needed for him to view the
contents, access should be forbidden. It can thus be seen that the traditional view of discretionary access control, being that a subject has a certain right on a certain object, may not be sufficient for the workflow environment as it may involve groups and group related rights within a specific context. The propagation of access rights can also be done in different ways, e.g. per process instance (just for now) or per process definition (for a kind of workflow). These access control (authorization) needs primarily ensure confidentiality, but also assist in ensuring that unauthorized changes to information can not take place (integrity).

Workflow systems may be critical to the success of a business and non-repudiation of service can be extremely important. In a centralized homogeneous environment service could be denied should the single point of control (the workflow engine) fail. This may, however, not be the only reason for service denial, since the human factor also needs to be considered.

The centralized workflow model suffers the same problem as any centralized architecture, that is one point of failure, being the workflow engine. Availability of information can therefore be lost completely, should the workflow engine fail. This immediately brings to mind issues of recoverability from backup mechanisms [3]. For any backup mechanism to be effective it should be possible to return to a recoverable state. In a workflow environment, where the duration of “transactions” may be measured in days, weeks or even months, this could present problems if the whole workflow is considered one transaction.

The database notion of well-formed transactions therefore potentially needs revisiting in the workflow environment. A workflow transaction could be seen as a meta-transaction, incorporating traditional transactions as part of its execution. Well-formed transaction properties, like failure atomicity, may only apply at the individual transaction level and not for a meta-transaction. Similarly a meta-transaction may not be serializable in the traditional sense of the word, i.e. its outcomes may differ depending on the parallelisms and timing involved.

In distributed workflow systems the availability issues become less serious provided that the workflow engines can “cover” for one another and that the shared data is still available. In heterogeneous workflow environments the workflow enactment services need a trust relationship with one another. It is therefore important that the workflow engines must be able to mutually authenticate.

4.3 Run-time interactions

The workflow client application is the point of contact with the user. It is therefore important that user authentication is done. Different scenarios regarding the distribution of the worklist handlers correspond to the different client/server partitions as discussed in [4]. As such the security needs and mechanisms will be dependant on the precise implementation. The following comments, however, can be considered as being of general interest to all different partitions.

To perform a task it may be necessary to invoke certain applications. These can range form a complex financial package to a word processor. Although the specifications of how this invocation should be done is not formal yet it can be seen that this has considerable impact on the security services. The workflow environment may have limited control over the invoked application. This could, for example, in a word processor allow the user to misuse the cut-copy-paste functionality to circumvent the confidentiality service.

The availability of applications on the client machines could hamper the support for non-repudiation. If the application that should be invoked is not available in the operational environment, access to certain information may be denied.

This section showed that the run-time interactions with users indeed highlight certain security concerns within the framework of the WfRM.
5. CONCLUSION

The Workflow Reference Model is only concerned with standardizing the interfaces between the various components. Security is not addresses in the WfRM per se. The implementation of security features in various products is presumably left as a differentiating factor between products.

The Workflow Management Coalition's vision of heterogeneous workflow management systems (or components thereof) cooperating seamlessly in a ubiquitous fashion can only be fulfilled if the different workflow systems (or components) can trust each other. Trusting a system implies that the trusted system provides at least as much protection to my information as I do. A potential trusted party would thus be evaluated in terms of the security services it has implemented.

Security considerations will therefore have to form an integral part of establishing a relationship between heterogeneous distributed workflow management systems. This paper has highlighted some of the security concerns regarding the main functional areas within the WfRM.

In this light many research projects can be identified. This work can be focussed on any of the security services and its implementation in a workflow environment. The authors will particularly address the modeling of access control in a workflow environment in future research.

BIBLIOGRAPHY


