HARDWARE,
SOFTWARE
AND PEOPLEWARE

SAICSIT 2001

Edited by
Karen Renaud
Paula Kotzé
Andries Barnard
HARDWARE,
SOFTWARE
AND PEOPLEWARE

South African Institute of Computer Scientists and Information Technologists
Annual Conference
25 – 28 September 2001
Pretoria, South Africa

SAICSIT 2001

Edited by Karen Renaud, Paula Kotzé & Andries Barnard
University of South Africa, Pretoria
### Table of Contents

**Message from the SAICSIT President** ................................................ iv  
**Message from the Chairs** .................................................................... vi  
**Conference Organisation** ..................................................................... vii  
**Referees** ................................................................................................ viii  

**Keynote Speakers**

- *Cyber-economies and the Real World* ................................................ xi  
  Alan Dix  
- *Computer-aided Instruction with Emphasis on Language Learning* ........ xiv  
  Lut Baten  
- *Internet and Security Trends* .............................................................. xv  
  Arthur Goldstuck  
- *The Future of Data Compression in E-technology* .............................. xvi  
  Nigel Horspool  
- *Strategic Planning for E-Commerce Systems: Towards an Inspirational Focus* ........ xvii  
  Raymond Hackney

**Research Papers**

**Human-Computer Interaction / Virtual Reality**
- *The Development of a User Classification Model for a Multi-cultural Society* .......... 1  
  M Streicher, J Wesson & A Calitz  
- *Real-Time Facial Animation for Virtual Characters* ............................... 11  
  D Burford & E Blake  
- *The Effects of Avatars on Co-presence in a Collaborative Virtual Environment* ........ 19  
  J Casanueva & E Blake

**Education**
- *Structured Mapping of Digital Learning Systems* ..................................... 29  
  E Cloete & L Miller

**Formal Methods**
- *The specification of a multi-level marketing business* ........................... 35  
  A van der Poll & P Kotzé  
- *Finite state computational morphology - the case of the Zulu noun* ............. 45  
  L Pretorius & S Bosch  
- *Combining context provisions with graph grammar rewriting rules: the three-dimensional case* ......................................................... 54  
  A Barnard & E Ehlers

**Human-Computer Interaction / Web Usability**
- *Web Site Readability and Navigation Techniques: An Empirical Study* ........... 64  
  P Licker, R Anderson, C Macintosh & A van Kets  
- *Jiminy: Helping Users to Remember Their Passwords* ............................ 73  
  K Renaud & E Smith

**Information Security**
- *Computer Security: Hacking Tendencies, Criteria and Solutions* ................ 81  
  M Botha & R von Solms  
- *An access control architecture for XML documents in workflow environments* .... 88  
  R Botha & J Elof
Graphics and Ethics
Model-based Segmentation of CT Images .......................................................... 96
O Marte & P Marais
Towards Teaching Computer Ethics................................................................. 102
C de Ridder, L Pretorius & A Barnard

Human-Computer Interaction / Mobile Devices
Ubiquitous Computing and Cellular Handset Interfaces – are menus the best way forward? ............................................................... 111
G Marsden & M Jones
A Comparison of the Interface Effect on the Use of Mobile Devices............... 120
J Franken, A Stander, Z Booley, Z Isaacs & R Rose
The Effect of Colour, Luminance, Contrast, Icons, Forgiveness and Closure on
ATM Interface Efficiency .................................................................................. 129
A Stander, P van der Zee, & Y Wang

Object Orientation
JavaCloak - Considering the Limitations of Proxies for Facilitating Java
Runtime Specialisation ..................................................................................... 139
K Renaud

Hardware
Hierarchical Level of Detail Optimization for Constant Frame Rate Rendering..... 147
S Nirenstein, E Blake, S Windberg & A Mason
A Proposal for Dynamic Access Lists for TCP/IP Packet Filtering ................. 156
S Hazelhurst

Information Systems
The Use of Technology to Support Group Decision-Making in South Africa .......... 165
J Nash, D Gwilt, A Ludwig & K Shaw
Creating high Performance I.S. Teams ............................................................. 172
D C Smith, M Becker, J Burns-Howell & J Kyriakides
Issues Affecting the Adoption of Data Mining in South Africa ......................... 182
M Hart, E Barker-Goldie, K Davies & A Theron

Information Systems / Management
Knowledge management: do we do what we preach? .................................... 191
M Handzic, C Van Toorn, & P Parkin
Information Systems Strategic Planning and IS Function Performance:
An Empirical Study ......................................................................................... 197
J Cohen

Formal Methods
Implication in three-valued logics of partial information .................................. 207
A Britz
Optimal Multi-splitting of Numeric value ranges for Decision Tree Induction .... 212
P Lutu
Abstracts of Electronic Papers

Lessons learnt from an action research project running groupwork activities on the Internet: Lecturers' experiences .............................................................. 221
T Thomas & S Brown

A conceptual model for tracking a learners' progress in an outcomes-based environment
R Harmse & T Thomas

Introductory IT at a Tertiary Level – Is ICDL the Answer? .................... 222
C Dixie & J Wesson

Formal usability testing – Informing design ................................................ 222
D van Greunen & J Wesson

Effectively Exploiting Server Log Information for Large Scale Web Sites ........ 223
B Wong & G Marsden

E von Solms & J Eloff

A Pattern Architecture, Using patterns to define an overall systems architecture ........................................................................................................ 224
J van Zyl & A Walker

Real-time performance of OPC ................................................................. 224
S Kew, & B Dwolatzky

The Case for a Multiprocessor on a Die: MoaD ........................................ 225
P Machanick

Further Cache and TLB Investigation of the RAMpage Memory Hierarchy .... 225
P Machanick & Z Patel

The Influence of Facilitation in a Group Decision Support Systems Environment ...... 226
T Nepal & D Petkov

Managing the operational implications of Information Systems ................ 226
B Potgieter

Finding Adjacencies in Non-Overlapping Polygons ..................................... 226
J Adler, GD Christelis, JA Deneys, GD Konidarlis, G Lewis, AG Lipson, RL Phillips, DK Scott-Dawkins, DA Shell, BV Strydom, WM Trakman & LD Van Gool
Message from the SAICSIT President

The South African Institute of Computer Scientists and Information Technologists (SAICSIT) was formed in 1982 and focuses on research and development in all fields of computing and information technology in South Africa. Now in the 20th year of its existence, SAICSIT has come of age, and through its flagship series of annual conferences provides a showcase of not only the best research from the Southern-African region, but also of international research, attracting contributions from far afield. SAICSIT does, however, not exist or operate in isolation.

More than 50 years have passed since the first electronic computer appeared in our society. In the intervening years technological development has been exponential. Over the last 20 years there has been a vast growth and pervasiveness of computing and information technology throughout the world. This has led into the expansion and consolidation of research into a diversity of new technologies and applications in diverse cultural environments. During this period huge strides have also been made in the development of computing devices. The processing speed of computers has increased thousand-fold and memory capacity from megabytes to gigabytes in the last decade alone. The Southern African region did not miss out on these developments.

It is hardly possible for such quantitative expansion not to bring a change in quality. Initially computers had been developed mainly for purposes such as automation for the improvement of processing, labour-reduction in production and automation control of machinery, with artificial intelligence, which made great strides in the 1980s, seen as the ultimate field to which computers could be applied. As we moved into the 1990s it was recognized that such an automation route was not the only direction in the improvement of computers. The expansion of processing power has enabled image data to be incorporated into computer systems, mainly for the purpose of improving human utilisation. For most computer technologies of the 1990s, including the Internet and virtual reality, automation was not the ultimate purpose. Humans were increasingly actively involved in the information-processing loop. This involvement has gradually increased as we move into the 21st century. Development of computer technology based not on automation, but on interaction, is now fully established.

The method of interaction has significantly changed as well. The expansion of computer ability means that the same function can be performed far more cheaply and on smaller computers than ever before. The advent of portable and mobile computers and pervasive computing devices is ample evidence of this. The need for users to be at the same location as a computer in order to reap the benefits of software installed on that computer is becoming an obsolete notion. Time and space are no longer constraints. One of the most discussed impacts of computing and information technology is communication and the easy accessibility of information. This changes the emphasis for research and development – issues such as cultural, political, and economic differences must, for example, be accommodated in ways that researchers have not previously considered. Our goal should be to enable users to benefit from technological advances, hence matching the skills, needs, and expectations of users of available technologies to their immense possibilities.
The conference theme for the SAICSIT 2001 Conference – *Hardware, Software and Peopleware: The Reality in the Real Millennium* – aims to reflect technological developments in all aspects related to computerised systems or computing devices, and especially reflect the fact that each influences the others.

Not only has SAICSIT come of age in the 21st century, but so has the research and development community in Southern Africa. The outstanding quality of papers submitted to SAICSIT 2001, of which only a small selection is published in this collection, illustrates both the exciting and developing nature of the field in our region. I hope that you will enjoy SAICSIT 2001 and that it will provide opportunities to cultivate and grow the seeds of discussion on innovative and new developments in computing and information technology.

Paula Kotzé  
SAICSIT President
Message from the Chairs

Running this conference has been rewarding, exciting and exhausting. The response to the call for papers we sent out in March was overwhelming. We received 64 paper submissions for our main conference and twelve for the postgraduate symposium. We had a panel of internationally recognized reviewers, both local and international. The response from the reviewers was impressive – accepting a variety of papers and mostly returning the reviews long before the due date. We were struck, once again, by the sheer magnanimity of academia – as busy as we all are, we still manage to contribute fully to a conference such as SAICSIT.

After an exhaustive review process, where each paper was reviewed by at least three reviewers, the program committee accepted 26 full research papers and 14 electronic papers. Five papers were referred to the postgraduate symposium, since they represented work in progress – not yet ready for presentation to a full conference but which nevertheless represented sound and relevant research. The papers published in this volume therefore represent research of an internationally high standard and we are proud to publish it. Full electronic papers will be available on the conference web site (http://www.cs.unisa.ac.za/saicsit2001/).

Computer Science and Information Systems academics in South Africa labour under difficult circumstances. The popularity of IT courses stems from the fact that IT qualifications are in high demand in industry, which leads in turn to a shortage of IT academic staff to teach the courses, even when posts are available. The net result is that fewer people teach more courses to more students. IT departments thus rake in ever-increasing amounts of state subsidy for their universities. These profits, euphemistically labelled “contribution to overhead costs”, are deployed in various ways: cross-subsidization of non-profitable departments; maintenance of general facilities; salaries for administrative personnel, etc. Sweeteners of generous physical resources for the IT departments may be provided. We have yet to hear of a University in South Africa where significant concessions have been made in terms of industry-related remuneration. At best, small subventions are provided. As a result, shortages of quality staff remain acute in most IT departments – especially at senior teaching levels. What is even worse is that academics in these departments have to motivate the value of their conference contributions and other IT outputs to selection committees, often dominated by sceptical academic power-brokers from the more traditional departments whose continued survival is underwritten by IT’s contribution to overhead costs.¹

The papers published in this volume are conclusive evidence of the indefatigability and pertinacity of Computer Science and Information Systems academics and technologists in South Africa. We are proud to be part of such a prestigious and innovative group of people.

In conclusion, we would like to thank the conference chair, Prof Paula Kotzé, for her support. We also specially thank Prof Derrick Kourie for his substantial contribution. Finally, to all of you, contributors, presenters, reviewers and organisers – a big thank you – without you this conference could not be successful.

Enjoy the Conference!
Karen Renaud & Andries Barnard

¹ This taken almost verbatim from Professor Derrick Kourie’s SACLA 2001 paper titled: “The Benefits of Bad Teaching”. 
Conference Organisation

General Chair
Paula Kotze

Programme Chairs
Karen Renaud
Andries Barnard

Organising Committee
Chairs
Lucas Venter, Alta van der Merwe

Art and Design
Tersia Parsons

Sponsor Liaison
Paula Kotze, Chris Bornman

Secretarial & Finances
Christa Prinsloo, Elmarie Havenga

Marketing & Public Relations
Klarissa Engelbrecht, Elmarie van Solms, Adriaan Pottas, Mac van der Merwe

Audio Visual
Tobie van Dyk, Andre van der Poll, Mac van der Merwe

Program Committee

Bob Baber – McMaster University, Canada
Andries Barnard – University of South Africa
Judy Bishop – University of Pretoria
Andy Bytheway – University of the Western Cape
Andre Calitz – University of Port Elizabeth
Elsabe Cloete – University of South Africa
Carina de Villiers – University of Pretoria
Alan Dix – Lancaster University, United Kingdom
Jan Eloff – Rand Afrikaans University
Andries Engelbrecht – University of Pretoria
Chris Johnson – University of Glasgow, United Kingdom
Paul Licker – University of Cape Town
Paula Kotze – University of South Africa
Derrick Kourie – University of Pretoria
Philip Machanick – University of the Witwatersrand
Gary Marsden – University of Cape Town
Don Petkov – University of Natal Pietermaritzburg
Karen Renaud – University of South Africa
Ian Sanders – University of the Witwatersrand
Derrick Smith – University of Cape Town
Harold Thimbleby – Middlesex University, United Kingdom
Theda Thomas – Port Elizabeth Technikon
Herna Viktor – University of Pretoria, South Africa
Bruce Watson – Universities of Pretoria and Eindhoven
Janet Wesson – University of Port Elizabeth
Referees

Molla Alemayehu
Trish Alexander
Adi Attar
Bob Baber
Andries Barnard
John Barrow
Judy Bishop
Gordon Blair
Arina Britz
Andy Bytheway
André Calitz
Charmain Cilliers
Elsabe Cloete
Gordon Cooper
Richard Cooper
Annemieke Craig
Thad Crews
Quintin Cutts
Michael Dales
Carina de Villiers
Alan Dix
Dunlop Mark
Elize Ehlers
Jan Eloff
Andries Engelbrecht
Klarisssa Engelbrecht
David Forsyth
John Galletly
Vashti Galpin
Wayne Goddard
Alexandre Hardy
Scott Hazelhurst
Johannes Heidema
Tersia Hörne
Chris Johnson
Bob Jolliffe
Paula Kotzé
Derrick Kourie
Les Labuschagne
Paul Licker
Philip Machanick
Anthony Maeder
David Manlove
Gary Marsden
Thomas Meyer
Elsa Naudé
Martin Olivier
Don Petkov
Pekka Pihlajasaari
Nelisha Pillay
Laurette Pretorius
Karen Renaud
Ingrid Rewitzky
Sheila Rock
Markus Roggenbach
Ian Sanders
Justin Schoeman
Martie Schoeman
Elsje Scott
Derek Smith
Elmé Smith
Adrie Stander
Harold Thimbleby
Theda Thomas
Judy Van Biljon
Alta Van der Merwe
André van der Poll
Tobias Van Dyk
Lynette van Zijl
Lucas Venter
Herna Viktor
Bruce Watson
Janet Wesson

Conference
Sponsors

UNISA
ABSA Group
Microsoft
ORACLE
NRF
Keynote Abstracts
An Access Control Architecture for XML documents in workflow environments

Reinhardt Botha\textsuperscript{a} Jan Eloff\textsuperscript{b}

\textsuperscript{a}Faculty of Computer Studies, Port Elizabeth Technikon, Port Elizabeth
rbotha@computer.org
\textsuperscript{b}Department of Computer Science, Rand Afrikaans University, Johannesburg
eloff@rka.rau.ac.za

Abstract

The eXtensible Markup Language (XML) are being upheld as having the potential to change the way business is conducted. This will be effected by changing the way in which information is shared. However, with the sharing of information, information security becomes a concern. This paper presents an access control architecture that allows for the sharing of XML documents in workflow environments. The architecture addresses the issue of access control from two perspectives. On the one hand, issues regarding the confidentiality of information are addressed. On the other hand, the semantic integrity of information is attended to. The paper shows how the access control services, provided as part of the architecture, achieve these objectives.

Keywords: systems architecture, information security, access control, workflow systems

Computing Review Categories: D2.11, D4.6, H4.1, I7.2, K6.3, K6.5

1 Introduction

The eXtensible Markup Language (XML) has had a profound impact on the computer industry since its proposal by the World Wide Web Consortium (W3C) in February 1998 [15]. XML is often mistaken, by the uninformed, as a technology. However, XML presents a standard [4] with respect to information representation. As such, other tools and technologies make use of the XML standard for a uniform way to transfer and manipulate data.

XML, as its name suggests, is a markup language. It is often discussed in the context of content markup, content management, search engines and meta languages. XML differs from the HTML markup language in one fundamental way - it is extensible. With HTML, you are given a set of tags (markups) and you are bound by the given set as you markup a document. With XML, you “invent” the tags, so you can use XML to markup different types of documents for very specific purposes. Adjunct languages (e.g. XSL [1], XPointer [9] and XLink [10]) can be used to associate these elements to some rendering or linking semantics for their display on different media types. This allows a definite separation between the description of the structure of the document and the description of its representation on the respective media types. The structure of an XML document can be dictated by a specification in the form of a DTD [3] or an XML Schema [11]. The resultant document can be viewed and parsed as a hierarchical structure, referred to as the document tree.

Since sensitive content may be contained in an XML document, organizations that wish to utilize XML documents must consider the information security aspects associated with XML.

2 Information Security Aspects of XML Documents

Information Security is often described in terms of the five services (authentication, access control, confidentiality, integrity and non-repudiation) [13] that is employed to ensure that information maintains certain attributes, i.e. that the information stays available and confidential while maintaining a state of integrity.

For XML documents, as with any other information, these attributes of secure information need to be maintained. A distinction can be made between the protection of information while in transit and the protection of information while at the end-points, i.e. when it resides on the client or server machines. This paper is not concerned with the information security issues while the document is in transit – standard encryption and other techniques that are available within the communication technology field can be employed in this respect. This paper will only address the protection of XML documents while at the end-points.

At the end-points information security requirements can be found in questions such as: “Who may read this document?”, “May Sue read this part of the document?” and “Should Tom be able to edit this
3 Access Control Requirements for XML Documents

XML documents have a structure as specified by an XML Schema [11] or a DTD [3]. An XML Schema provides a richer way of expressing structure than a DTD, but, in essence, both ways specify rules regarding the relation between parts of the document. In the light of the fact that an XML document can be perceived as a tree structure [19], we can see this as restrictions on the child-nodes of parent-nodes.

Consider, for example, the representation of a personnel record, depicted as an XML document and its accompanying DTD in Figure 1. The DTD show that elements consist out of other elements. For example,
shows that pers_details consists of a surname, first_name, other_initials and a home_address. Note that home_address, in turn, consist out of several elements. Elements can also have attributes, such as staff_member who has a personnel_number. The DTD furthermore controls the number of instances of an element. For example, other_initials indicates that it is an optional element, after all, not everybody has second names. Similarly leave_period* shows that the element may occur zero, one or many times, while a + would indicate that the element could occur one or more times.

A document, such as the personnel record, contains parts which are more sensitive than other parts. An employee’s salary can be considered more sensitive than the employee’s address, which, in turn, could be more sensitive than his personnel number. Different parts of the document must thus be protected in different ways. We can therefore state that XML documents require a fine-grained access control mechanism.

The permissions associated with access control documents will be dependent on the semantics of the document (or the component of the document) to which it applies. However, permissions can be seen as abstractions of the general actions that can be observed with XML documents. All of these will be connected to the creation, inspection and modification of nodes and elements in the XML document tree. In the personnel record example, the permission “update salary” reflects the ability to replace the values in /salary_details/©basic_pay as well as writing historic information in the relevant part of the documents, say (among others)

\{/salary_details/salary_history/©basic_pay\}.

In a workflow environment these actions may not be applicable to the same user all of the time. The context of the task in the bigger business process will influence the access control decision [2, 17]. For example, after a leave application has been approved the dates may not be changed.

We need to consider the role that XML documents can play in a workflow environment. The next section thus describes the environment in which we operate.

4 XML documents in the workflow environment

XML documents may traverse the organization for numerous reasons. However, in order to provide a working context, this paper concentrates on XML documents that are propagated through a workflow management system.

Figure 2 graphically depicts the execution of a leave application process for employee 1. It assumes that workflow definition tools were used to define the leave application process as input to the workflow enactment service. The workflow enactment service will, on request of a user, start a new instance of a specific workflow. Thereafter the request must be approved by employee 1’s Manager and the Human Resource department. In practice this would be more complex as provision must be made for negotiating the leave or rejecting the leave. The example is sufficient, however, to show the principles that will be described in this paper.

For the purposes of this paper we will furthermore assume that the workflow environment subscribes to the general idea behind role-based access control [16]. Therefore users are associated with roles, which, in turn, are associated with permissions. A role may also be related to other roles according to a partial order. Roles inherit the permissions of the roles that are inferior to them in the partial order, whereas users may also assume the roles that are inferior in the partial order to their assigned roles. The partial order is referred to as the role hierarchy.

Each of the tasks may involve a number of XML documents. The task definition will specify which role may perform the task. In environments where strict least privilege is a requirement [2, 5] this may involve the introduction of special roles in the role hierarchy. Since access to these documents is restricted at a fine granularity, permissions will be defined in terms of the constituent components of the XML document.

Figure 2 shows how the personnel record defined in Figure 1 is propagated, together with a leave form during a leave application process, in order to assist the Manager and Human Resource department with their decision. The access control requirements to the two XML documents change as their role in the workflow changes. This is considered in more detail in the following section.

5 Controlling access to XML documents

In the leave application example, two documents were encountered: a leave application and a personnel record. Consider the access control requirements of the two documents in turn.

Firstly, the personnel record. At the “Manager Approval” task in the workflow the personnel record should not be edited. Even so, not all of the information is relevant to making the decision and should, therefore, not be viewed by the manager. An example of this is the salary – the salary details should not play a role in the leave decision. Since it is sensitive information it should not be displayed at inappropriate times. A similar argument holds at the “HR Approval” task in the workflow. However, at the “HR Approval” task a portion of the personnel record that deals with leave recording might need to be updated.

1 In practice, this would probably only happen once both approval tasks have been completed. Furthermore, it will most
At that stage, however, other details such as the salary may not be updated.

The leave form is subject to similar arguments. When the employee requests leave, the leave form can be updated by him. However, when the document arrives at the Manager or HR, they should not be able to edit the leave application details. They should only be able to complete the portions that they need to complete to grant approval. Once the leave is approved the leave details should not be edited at all.

The question of how we specify what is allowed, i.e. how we specify the permissions then comes to mind.

5.1 Specifying permissions

A permission indicates the ability to perform some action. With XML documents these abilities relate to the actions that may be performed on the nodes of the document tree. A number of observations regarding the specification of access permissions are thus in order.

Firstly note that due to the hierarchical nature of the document tree, permissions can be filtered down the document tree. For example, the fact that someone must have read permission on the \(<home_address>\) node of the XML document, implies that the person must have read access on the child nodes of \(<home_address>\). However we must be able to specify that a person may read \(<home_address>\) but may not read the \(<street>\) node. We call this permission the "view person living area" permission. Explicit negative permissions must therefore be included.

A permission can therefore be generalized to be a set of actions that can be performed by a role. The "view person living area" permission can thus be expressed as:

\[\{(/pers_details/home_address,read,+), (/pers_details/home_address//street,read,-)\}\]

In general, a permission \(P\) is 
\[P = \bigcup\{\text{node, action, sign}\} \text{ where node is the XPath expression}\]
identifying any node in the document tree, action is an action that is allowable when the sign is positive and disallowable when sign is negative. The actions represented by action are interdependent. For example, the action to change implies the ability to read the current value. Table 1 summarizes possible actions on nodes in the document tree and their relationships. Damiana et al. [8] refined this concept to include eight authorization types which is based on whether their is inheritance and the type of inheritance that that permission have.

The permissions are specified in terms of the node of the resultant tree. However, the specification will happen once for every type of document, i.e. in XML terms, it will happen once for each DTD or Schema definition. XML schemas may allow the further refinement of these permissions by allowing the specification of "types" of nodes, e.g. an address type. Investigation as to the specification of permission based on
the type of the node, rather than its position in the document tree warrants further investigation. For the purpose of this paper, however, this issue is considered future work.

5.2 Administration of Access Control

If permissions are generalized as above then the access control administration problem can easily be addressed through standard role-based access control mechanisms such as those described in [16].

This would then involve the association of permissions with roles and roles with users. Since the permissions to a document change in terms of the context of the document, the fact that a user is associated with a permission is not a sufficient condition for access control. Specific tasks will also be associated with roles [5]. A user will have to belong, therefore, to a role equal or superior to the role required for the task to be able to perform the task. The user will only receive the permissions associated with the task.

The granting of permissions will be governed by several architectural components in the workflow environment. The next section discusses the role that the various architectural components play in securing XML documents.

6 Architectural components

Figure 3 depicts the components present in the proposed architecture. In order to see what the different components’ purpose is, it is best to look at the various activities that can take place. The activities fall in one of two categories: reading or changing. These two categories are discussed in paragraphs 6.2 and 6.3 respectively.

Within the architecture we can observe that XML documents can reside either on the client or on the server. The security mechanisms devised are based on the premise that we cannot trust the client [18].

6.1 Building the worklist

Within a workflow environment the worklist handler would be responsible for determining which users should do what. The workflow server will, based on the process definition and the completed tasks, determine the tasks that should occur next. These tasks will be handed off to the worklist handler who has the responsibility of determining who may perform those tasks. The Access Control Service must, therefore, interpret the defined Access Control Rules. These include separation of duty specifications [2] such as that “an employee may not approve his own leave”.

The result of this access decision must be communicated to the users. If a “push” paradigm is followed, the worklist handler will decide who must do the task and inform the specific user through the worklist. If a “pull” paradigm is followed then the worklist handler will inform all users that may possibly do the task of the presence of the task in their worklists. As soon as one of the users commits to perform the task, the notification of the task will be removed from the other users’ worklists.

The first level of access control is thus inherent to the workflow environment in that only users who should perform the task will know of the existence of the task. However, when a user performs the task, i.e. acts on the item in the worklist a further level of access control is required. The decisions that must be made can be stated as the questions “What may the user see?” and “What may the user change?”. These two questions are addressed in paragraphs 6.2 and 6.3 respectively.

6.2 Restricting what the user may see

Since an XML document may contain information of various degrees of sensitivity, the complete document will seldomly be used as is. For example, in the leave application cited, the salary details may not play a role in granting/disallowing the leave.

Figure 3 shows that the Access Control Service within the worklist handler is going to be responsible for retrieving the XML document from the XML document store. The Access Control Service will dynamically create an XSL style sheet [6] (based on the task definition) that would allow for the pruning of the document tree by a parser application. Damiani et al. [8] explain the “pruning” of an XML document tree in detail. The pruned document will only contain the information which is relevant to that step in the workflow. The document can then be transferred to the client, whereafter the client will display the document in its browser interface, possibly based on an existing style sheet. At that stage no sensitive information resides on the client as the document was pruned appropriately on the server. In this respect

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
<th>Implied</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>The content of the node can be viewed</td>
<td>Read</td>
</tr>
<tr>
<td>Edit</td>
<td>Change the content of node</td>
<td>Read, Edit</td>
</tr>
<tr>
<td>Add</td>
<td>Create a node</td>
<td>Edit, Read</td>
</tr>
<tr>
<td>Append</td>
<td>Create child nodes</td>
<td>Read, Add(children nodes)</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete the node</td>
<td>Edit, Read</td>
</tr>
</tbody>
</table>

Table 1: Actions on XML documents
Process definition tools are used to define the workflow.

Access Control Rules Expression

Workflow Server is responsible for determining the routing of information between participants in the organization

Web-enabled Worklist Handler

Server-side technologies

Client-side technologies

XML Document Store

DTD or XML Schema definitions

Figure 3: Architecture for controlling access to XML documents

<?xml version="1.0"?>  
<staffmember personnel_number="001">  
  <pers_details>  
    <surname>Jones</surname>  
    <first_name>Ben</first_name>  
    <other_inits>GF</other_inits>  
  </pers_details>  
  <old_leave_details>  
    <leave_period>  
      <start>10-May-2000</start>  
      <end>12-May-2000</end>  
    </leave_period>  
    <leave_period>  
      <start>16-Dec-2000</start>  
      <end>7-Jan-2001</end>  
    </leave_period>  
  </old_leave_details>  
</staffmember>

Figure 4: The XML personnel record after transformation

we therefore don’t have to worry about trusting the client application and execution environment for not making sensitive information available.

6.3 Updating the documents

The contents of the XML document need to change during the workflow. In the discussed example the leave form must be completed by the employee, then it must be approved (or not) by the Manager and HR. All of these activities require the documents to be updated. However, different parts of the document are being edited at different times. For example, when the document is being approved by the Manager, he should not be able to edit the leave application dates.

Updating the information on the server requires the client to send the data to the server. The client will render the HTML either as a HTML form [12] or display it through an applet. The applet option is depicted in Figure 3. The applet approach allows client side validations to occur. However, due to the untrustworthy nature of clients, these checks will have to be repeated on the server. Either way, the information will be sent to the server.

As mentioned the server will have to check that the XML document that arrives adheres to the DTD or Schema definition. Not only should it conform to the structure, but the resultant document should be compared with the original document to confirm that only nodes which should have been updated were updated, i.e. that the client application sent expected information. Experienced hackers will find it easy to alter messages on the client side [18]. Once this has been confirmed, the XML document that arrived at the server (which represents a pruned version of the original) must be merged with the original document.
and the updated version stored in the persistent document store for future use. This logic is embedded in the architecture in the “Servlet responsible for modifications” component. Although the term servlet is used, it may also be achieved through other server side technologies such as CGI scripting or ASP scripting.

The updating of information requires careful consideration as to the likelihood of information being altered on the client. The server side component that is responsible for validating the updates is thus of principle importance.

7 Conclusion

The architecture that was proposed here aims to ensure the availability, confidentiality and integrity of the XML documents. In particular, it addresses these needs from an access control perspective.

Availability is addressed through the inclusion of a persistent XML document store. However, the access control service will limit access to the documents based on the task to be performed. From a practical perspective it will occasionally be difficult to predict which document will be referred to during a specific task. It is, however, reasonable to assume that those documents will only be read and not updated.

It was shown that confidentiality can be achieved through pruning the XML document tree on the server. This ensures that no unnecessary and possibly sensitive information is displayed. For documents which must be available for continual reference at unpredictable times, a default pruning process that removes all possible sensitive information may thus take place. The architecture is therefore quite capable of ensuring that information is readily available, yet in a confidential manner.

Information will retain its semantic integrity. This is partially due to the reliance on the workflow systems to make the information available for update only when it should be updated. The server side checking that updates were in line with expectations adds a further degree of confidence that semantic integrity will remain intact.

Semantic integrity can still be at risk if the data that were expected to be edited is edited inappropriately. Technology solutions to this is not possible as human error will always play a role where information is being captured. This of course allows for experienced hackers to introduce errors that appear to be human errors. This is possible since messages can still be altered by client-based methods. However, the likelihood of this is greatly reduced through applying the proposed architectural principles.

Consider, for example, the inappropriate changes of salaries. A client-side attack is only possible when the salary must actually be edited, for example, during a salary review. At all other times any message that contains an updated XML document will be validated by the server against what should have been edited and inappropriate changes will be rejected.

In the proposed architecture the worklist handler has more functionality than what would typically be the case in a workflow environment where the documents are non-XML. This is due to the fact that the worklist handler does not (and, in most cases, cannot) concern itself with the contents of the document objects. However, considering the much finer granularity at which protection occurs the extra cost incurred is not significant.

The architecture presents several opportunities for further investigation and refinement of techniques employed. The following aspects require further attention.

The work reported on in this paper only relied on DTDs to express structure. The influence of the more powerful XML Schemas on the specification of access permissions must be considered. XML Schemas allow, for example, the specification of custom types which may influence how we wish to express access permissions.

The development of tools to support this type of architecture must also receive attention. As future work a generic forms handler that subscribes to the techniques required will be developed. Furthermore, a toolset to allow for the easy specification of access permissions is required.

It is believed that this architecture provides an excellent foundation for further work. The architecture contributes to the understanding of information security and, in particular, the dynamic access control needs in an internet-based workflow environment.

References


