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 Eloff
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 .............................................................. 221
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 Konidaris, 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
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”.

v1
Conference Organisation

General Chair
Paula Kotzé

Programme Chairs
Karen Renaud
Andries Barnard

Organising Committee Chairs
Lucas Venter, Alta van der Merwe

Art and Design
Tersia Parsons

Sponsor Liaison
Paula Kotzé, Chris Bornman

Secretarial & Finances
Christa Prinsloo, Elmarine Havenga

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

Conference
Sponsors

UNISA

ORACLE

Microsoft
Keynote Abstracts
The specification of a multi-level marketing business

J.A. Van der Poll\textsuperscript{a} \hspace{1cm} P. Kotzé\textsuperscript{b}

University of South Africa
Department of Computer Science and Information Systems
\textsuperscript{a}vdpolja@unisa.ac.za \hspace{1cm} \textsuperscript{b}kotzep@unisa.ac.za

Abstract

A formal specification of a multi-level marketing (MLM) business is presented. Specifying a MLM business boils down to specifying properties of and operations on mathematical forests and trees. The usefulness of the model-based specification language, Z, is investigated as a vehicle for a formal specification of these. Proof obligations aimed at corroborating the aptness of the specification are stated and discharged using the resolution-based, first-order theorem prover OTTER. Reasoning about two simple properties of the specification illustrates the utility of two automated reasoning strategies in the literature, namely avoiding equality and applying resonance.

Keywords: automated reasoning, formal specification, multi-level marketing, OTTER, resolution, set theory, Z

Computing Review Categories: D.2.4, F.3.1, F.4.1, I.2.3

1 Introduction

A formal specification of a multi-level marketing (MLM) business is presented using Z (e.g. Spivey [12]). A MLM business may be modelled by a mathematical forest consisting of various trees. The study of forests and trees is well established in Computer Science (see e.g. Baase et al. [2]). These structures have been used to describe various entities, for example, the specification of a text editor (Scheurer [11]) in ordinary Zermelo-Fraenkel set theory (Enderton [5]), but as far as we are aware, such structures have never been used to specify a MLM business and its various intrinsic operations in Z.

A lucrative feature of a formal specification is that the specifier can reason about the properties of the specification. One can show that the specification has certain characteristics or that certain undesirable consequences are absent. Various reasoning mechanisms are available to a specifier. Some well-known systems are the term-rewriting theorem provers like CaDiZ (Toyn [14]) and Z-Eves (Saaltink [10]). These systems are all interactive reasoning assistants. Alternatively, one can use a fully automatic reasoning assistant like OTTER (McCune [7]) or Gandalf (Tammet [13]). In this paper we use OTTER to discharge proof obligations arising from the specification. OTTER is a first-order, resolution-based theorem prover developed by William C. McCune at the Argonne National Laboratory in Illinois, USA. Two important sections often present in the input to OTTER are the \textit{usable list} and the \textit{set-of-support (sos) list}. One way in which to use OTTER is to place the relevant facts known to be true in the proof attempt in the usable list and the negation of what we want to prove in the sos.

Our MLM specification below gives rise to a rather large state space which in turn poses demanding challenges to a resolution-based automated reasoning assistant when reasoning about the properties of state components. We show that proof attempts of two rather simple properties call for the application of two important reasoning strategies, namely avoiding equality and using resonance. These strategies are only two of many. Other strategies for aiding the operation of a resolution-based reasoning assistant have been developed and presented in [16].

2 Background to MLM systems

Traditional multi-level marketing (MLM) businesses (e.g. GNLD [6]) have been around for a number of years and many conventional businesses are now starting to add some form of MLM to their existing operation. A multi-level marketing business normally markets consumable products\footnote{One of the most common consumable products in such a business is soap, since it normally has a high turnover. Alternatively, one can trade with luxury items like motor cars or houses, resulting in lower turnover and often lower profits.} and operates as follows:

A new distributor (also called a member) pays a registration fee and joins the business either as a direct associate of the company, or under an existing distributor called a sponsor. The sponsor does not sponsor the new distributor with money, but rather with knowledge and advice about the business. Both the sponsor and the new distributor then go on to each sponsor more new distributors, and so on. In this way a network of distributors of the products of the company is built. The sponsor is also called the upline of the new distributor, while the new distributor is generally known as the downline of the sponsor.
distributors have to renew their registration annually to remain in the business. An example of a MLM network is shown in Figure 1.

The distributors A1, A2, and A3 in Figure 1 are associated with the company directly are called the roots of the forest (or network in MLM terms).

The 'work' in a MLM business involves a number of steps:

- **Step 1**: Become a distributor of the consumable products available from the company.
- **Step 2**: Sponsor others to become product distributors as well.

Each product has a point value (pv for short) as well as a business value (bv for short) associated with it. Both the points and the business values are accumulated per distributor throughout a calendar month. At the end of the month the total business value (called a turnover in MLM terms) in the network for each distributor is calculated, and the distributor is paid (in the appropriate currency) a certain percentage of the total business value for his or her group. This is called a bonus.

The point value determines, on a sliding scale, the percentage to be used in the calculation of a bonus — a lower percentage for a lower turnover and a higher percentage for a higher turnover. Bonuses are the main source of income for distributors in such a network.

A distributor qualifies for a bonus at the end of a month only after having accumulated a certain number of points through personal product consumption and a (possibly different) number of points generated by downline distributors.

Distributors may also buy products from the company and sell these to customers at a profit. A customer is somebody who uses the products, but did not join the MLM business to become a distributor. A distributor may furthermore sell her or his business (or part of it) to an existing distributor, or somebody outside the MLM business who then automatically becomes a distributor. Multi-level marketing companies normally have fixed guidelines regarding such a transaction. (See e.g. the section The marketing plan, pages 20 — 23 in [6].)

We start with the following natural language requirements definition:

2The business value is an amount which is some indication of the price of the product.

3These multi-level marketing businesses are sometimes called pyramids. In a typical pyramid, the sponsor always earns more than any of his or her downline. While this might be true for some MLM businesses, it is not the case in general. Often the amount allocated to a downline is subtracted from the gross income of the sponsor (i.e. the upline). In this way it is quite possible for a downline to earn more per month than the upline. The use of a sliding scale furthermore ensures that the amount subtracted from the gross bonus of an upline is not more than the gross bonus itself.

Specify a system where new distributors may join a MLM business and become consumers of products available from the company. A distributor is allocated a unique identity code upon enrollment. Other information to be maintained for a distributor include the name, address, personal pv, and personal bv. Every product carries a point value as well as a business value. The system must allow for operations like enrolling a new distributor, ordering of products, calculation of bonuses at the end of a month and removing a distributor from the network of distributors.

Our specification is presented in the Z specification language (e.g. Spivey [12], Potter et al. [8]). From the above requirements definition we identify the following basic types:

\[ [ID, Name, Address, PV, BV, Bonus, Message] \]

The basic type Message above represents the set of all possible messages generated by the system. Its sole purpose is to provide feedback to users.

In our model we adopt a global viewpoint taken by the company that creates the environment within which distributors can build their businesses.

One possibility is that the company views the businesses as a group of top-level upline distributors, each with their own network of downline distributors. This viewpoint corresponds to the idea of a forest, where the roots of the trees in the forest represent the top-level upline distributors, and the network of downline distributors for each top-level upline is represented by a corresponding tree. This viewpoint fits the generic model for forests and trees put forward by Scheurer [11].

In this case, the state space is given by:

\[
\begin{align*}
\text{MLM} & : \text{known} : \mathbb{P}ID \\
\text{NRoots} & : \mathbb{P}ID \\
\text{NUplines} & : ID \leftrightarrow ID \\
\text{NDist} & : ID \rightarrow \text{Name} \times \text{Address} \times PV \times BV \times \text{Bonus} \\
\text{known} & = \text{dom} \text{NDist} \\
\text{dom} \text{NUplines} & \cup \text{ran} \text{NUplines} \subseteq \text{known} \\
\text{NRoots} & = \text{known} \setminus \text{ran} \text{NUplines} \\
\text{Inj}(\text{NUplines}) & : (\forall \text{Netw}) \\
& (\text{Netw} \subseteq \text{known} \land \text{NRoots} \subseteq \text{Netw} \land \\
& \text{ClosedBy} (\text{Netw}, \text{NUplines}) \\
& \rightarrow \text{Netw} = \text{known})
\end{align*}
\]

Schema MLM introduces 4 components and 2 auxiliary predicates:

- The set known contains the identity codes of all distributors known to the system.
• \textit{NRoots} represents the distributors who do not have an upline in the business but joined the company directly.

• The network of distributors is represented by relation \textit{NUplines}.

• The function \textit{NDist} represents a mapping from a unique identity code to the particulars for that distributor. Note that \textit{NDist} is partial, since for a given value of \textit{NDist} it need not be the case that \( \text{dom NDist} = \text{ID} \); neither is it injective since two (or more) distributors may have the same particulars (i.e. name, address, etc.).

• Relation \textit{NUplines} is injective since every distributor in the business has at most one upline. This property is captured by the predicate \textit{Inj(NUplines)}. The definition of \textit{Inj} is:

\begin{equation}
(\forall R)(\text{Inj}(R) \leftrightarrow \\
(\forall i)(\forall j)(\forall k)((i, k) \in R \land (j, k) \in R)
\rightarrow (i = j))
\end{equation}

Note that the first coordinate of a pair in \textit{NUplines} represents the upline of a \textit{downline} which is the second coordinate of the pair.

• The purpose of the last predicate in \textit{MLM} is to ensure that \textit{known}, which represents the set of all known identity codes, is the smallest inductive set of identity codes generated by relation \textit{NUplines}, starting from the root nodes, \textit{NRoots}. Predicate \textit{ClosedBy} is given by:

\begin{equation}
(\forall \text{Netw})(\forall R)
\
(\text{ClosedBy}(\text{Netw}, R) \leftrightarrow \\
(\forall i)(\forall j)((i \in \text{Netw} \land (i, j) \in R)
\rightarrow j \in \text{Netw}))
\end{equation}

3 System operations

Our first operation defined on the state is to register a new distributor. A new distributor (say \( p \)) may register either directly with the company (i.e. becomes a root node) or under an existing distributor (say \( q \)) in the network. As is customary in \textit{Z}, an unused identity code for the new distributor is generated by the system.

The following schema registers a new distributor under an existing one:
Finally, a robust operation to register a distributor is given by:

\[
\text{Register} \equiv \text{Register\_with\_upline} \lor \text{Register\_no\_upline} \lor \text{No\_more\_codes}
\]

Next, we specify an operation to capture a product order placed by a distributor.

\[
\text{Order} \equiv \text{Order\_product} \lor \text{Unknown\_distributor}
\]

Schema \textit{Order} introduces some functions:

- The functions \textit{first}, \textit{second}, and so forth project out an element at the appropriate position in the tuple.
- The symbol \(\oplus\) is the relational overriding operator and its effect is to replace a tuple in a relation. Therefore, \(\text{NDist}'\) is obtained from \(\text{NDist}\) by replacing the tuple with first coordinate \(id\) as specified above.

Supplying an unknown identity code results in:

\[
\text{Unknown\_distributor} \equiv \text{unknown} \lor \text{ID}
\]

A robust operation for placing an order is:

\[
\text{Order\_product} \equiv \text{Order} \lor \text{Unknown\_distributor}
\]

An important operation is to calculate the nett bonus for a distributor. The first step is to specify an operation to calculate the gross bonus to be paid to a distributor. For any distributor, say \(id\), the gross bonus is kept as the fifth coordinate of \(\text{NDist}(id)\).
of the gross bonuses of all the immediate downlines of id?.

A robust bonus calculation is specified as either a gross bonus calculation followed by the calculation of a nett bonus, or an appropriate error condition:

\[ \text{Bonus} = (\text{Gross Bonus \& Nett Bonus}) \lor \text{Unknown_distributor} \]

Our last operation deals with the scenario where a distributor decides to quit the business. Suppose a root distributor, say A1 in Figure 1, decides to drop out of the network. This has the following effect:

1. Remove A1 as a root element, i.e.
   \[ \text{NRoots}' = \text{NRoots} \setminus \{A1\} \quad (3) \]

2. Remove the pairs \( A1 \rightarrow B1 \) and \( A1 \rightarrow B2 \) from the network, i.e.
   \[ \text{NUplines}' = \{A1\} \triangleleft \text{NUplines} \quad (4) \]

The symbol \( \triangleleft \) is known as the domain anti-restriction operator. Its effect is to remove all tuples with the first coordinate being an element of the set on the left of the operator from a relation.

3. Make \( B1 \) and \( B2 \) root elements, i.e.
   \[ \text{NRoots}'' = \text{NRoots}' \cup \{B1, B2\} \quad (5) \]

As a schema operation we have:

\[ \text{DelRoot} \]

\[ \Delta \text{MLM} \]

\[ \text{id}?: \text{ID} \]

\[ \text{mes}!: \text{Message} \]

\[ \text{id}? \in \text{NRoots} \wedge \]

\[ \text{known}' = \text{known} \setminus \{\text{id}?\} \wedge \]

\[ \text{NDist}' = \{\text{ndist}\} \triangleleft \text{NDist} \wedge \]

\[ \text{NUplines}' = \{\text{id}?\} \triangleleft \text{NUplines} \wedge \]

\[ (\exists \text{FRoots} : \text{P ID} \bullet \]

\[ \text{FRoots} = \{r | (\text{ndist}, r) \in \text{NUplines}\} \wedge \]

\[ \text{NRoots}' = (\text{NRoots} \setminus \{\text{id}?\}) \cup \text{FRoots} \wedge \]

\[ \text{mes}!' = \text{Distributor_deleted} \]

Note that appropriate after state values are also specified for \text{known} and \text{NDist}.

Alternatively, if a non-root distributor, say B2, decides to quit the business then the following actions are applicable:

1. Node B2 is deleted from the forest. Deleting B2 from the forest amounts to removing the pairs \( B2 \rightarrow C1, B2 \rightarrow C2 \) and \( B2 \rightarrow C3 \) from the network, i.e.
   \[ \text{NUplines}' = \text{NUplines} \triangleright \{C1, C2, C3\} \quad (6) \]
as well as removing the pair $A1 \rightarrow B2$ from the network, i.e.
\[
N\text{Uplines}'' = N\text{Uplines}' \triangleright \{B2\} \quad (7)
\]
The symbol $\triangleright$ is known as the range anti-restriction operator. Its effect is to remove all tuples with the second coordinate being an element of the set on the right of the operator from a relation. Note that equation (6) can also be written as $\{B2\} \triangleleft N\text{Uplines}$. The reason for using the format in (6) is to simplify the definition of the after state value of $N\text{Uplines}$ in schema $\text{Del}_\text{-} \text{Not}_\text{-} \text{Root}$ below.

(2) Nodes $C1$, $C2$ and $C3$ become immediate downlines of $A1$:
\[
N\text{Uplines}''' = N\text{Uplines}'' \cup \{A1 \rightarrow C1, A1 \rightarrow C2, A1 \rightarrow C3\}
\]
As a schema we have ($\sim$ denotes an inverse):

\[
\text{Del}_\text{-} \text{Not}_\text{-} \text{Root}
\]
\[
\begin{array}{|l|}
\hline
\Delta \text{MLM} \\
\text{id}? : \text{ID} \\
\text{mes!} : \text{Message} \\
\text{id}? \in \text{ran} N\text{Uplines} \land \\
\text{known}' = \text{known} \setminus \{\text{id}?\} \land \\
\text{NDist}' = \{\text{id}?\} \triangleleft \text{NDist} \land \\
(\exists \text{id}; \text{FRoots} : \mathbb{P} \text{ID} \bullet \\
\text{q} = N\text{Uplines}''(\text{id}?)) \land \\
\text{FRoots} = \{\text{r} \mid (\text{id}?, \text{r}) \in N\text{Uplines}\} \land \\
N\text{Uplines}' = \\
(N\text{Uplines} \triangleright (\text{FRoots} \cup \{\text{id}?\})) \cup \\
\{(q, \text{r}) \mid \text{r} \in \text{FRoots}\}
\end{array}
\]
A robust operation for deleting a distributor is given by:
\[
\text{DelDist} \triangleq \\
\text{Del}_\text{-} \text{Root} \lor \text{Del}_\text{-} \text{Not}_\text{-} \text{Root} \lor \text{Unknown}_\text{-} \text{distributor}
\]
Summarising the MLM operations in this section gives:

\[
\text{Register} \triangleq \\
\text{Register}_\text{-} \text{with}_\text{-} \text{upline} \lor \text{Register}_\text{-} \text{no}_\text{-} \text{upline} \lor \\
\text{No}_\text{-} \text{more}_\text{-} \text{codes}
\]
\[
\text{Order}_\text{-} \text{Product} \triangleq \text{Order} \lor \text{Unknown}_\text{-} \text{distributor}
\]
\[
\text{Bonus} \triangleq (\text{Gross}_\text{-} \text{Bonus} \triangleright \text{Nett}_\text{-} \text{Bonus}) \lor \\
\text{Unknown}_\text{-} \text{distributor}
\]
\[
\text{Del}_\text{-} \text{Dist} \triangleq \text{Del}_\text{-} \text{Root} \lor \text{Del}_\text{-} \text{Not}_\text{-} \text{Root} \lor \\
\text{Unknown}_\text{-} \text{distributor}
\]

\section{4 Reasoning about the specification}

In \emph{Z}, system operations are implicitly assumed to preserve the state invariant (see e.g. Diller et al. \cite{4}), hence proving that the state invariant is preserved by an operation is not considered a necessary proof obligation in \emph{Z}. Nevertheless, it is a good idea to check that an operation does not incur unpredictable results as far as the invariant is concerned. In systems like the \emph{B} method (Abrial \cite{1}, Waeselynck \cite{17}) ensuring that an operation preserves the invariant is considered a proof obligation.

For example, suppose a specifier wants to verify that the property
\[
N\text{Roots}' = \text{known}' \setminus \text{ran} N\text{Uplines}'
\]
which is a conjunct of the invariant, is preserved by operation $\text{Register}$. Suppose a new distributor $p$ is registered below an existing distributor, $q$. Consider two restrictions specified in $\text{Register}_\text{-} \text{with}_\text{-} \text{upline}$:

\[
\text{known}' = \text{known} \cup \{p\} \quad (9)
\]
\[
N\text{Uplines}' = N\text{Uplines} \cup \{q \mapsto p\} \quad (10)
\]
Suppose we define:
\[
\text{NewRoots} = \text{known}' \setminus \text{ran} N\text{Uplines}'
\]
and pose the negation of the following equality in the OTTER set-of-support list
\[
(N\text{Roots} = \text{NewRoots})
\]
then the theorem prover fails to find any proof for (12) in 20 minutes. It is well known that equality reasoning poses demanding challenges to automated reasoning assistants (e.g. Boyer et al. \cite{3}, Quaife \cite{9}). Therefore, if we apply the first axiom of Zermelo-Fraenkel set theory (see e.g. Enderton \cite{5}), namely Extensionality, and simply replace (12) with (13) below
\[
(\forall x)(x \in N\text{Roots} \leftrightarrow x \in \text{NewRoots})
\]
then OTTER finds a short proof in 0.49 seconds.\footnote{Proof performed on Linux Red Hat 6.2 on a Pentium III running at 600 MHz.}
Note that this result also implies that the set of root nodes (i.e. $N\text{Roots}$) is not affected by operation $\text{Register}_\text{-} \text{with}_\text{-} \text{upline}$.

Another example of a proof is to show that the cardinality of a set is preserved, e.g.
\[
\#\text{NDist}' = \#\text{NDist}
\]
after operation $\text{Order}_\text{-} \text{product}$. Suppose the precondition $\text{id}? \in \text{known}$ is satisfied and the after state value
of $\text{NDist}$ is as reflected in operation $\text{Order}$:

$$\text{NDist'} =$$

$$\text{NDist } \oplus$$

$$\{ \text{id} \mapsto$$

$$\text{(first}(\text{NDist}(\text{id})), \text{second}(\text{NDist}(\text{id})), \text{pv}, \text{bv}, \text{bonus})\}$$

Possible definitions for reasoning about the cardinality of a set are (e.g. [15]):

$$(\forall A)(\#A = 0 \leftrightarrow A = \emptyset)$$

$$(\forall A)(\forall n)(\#A = n + 1 \leftrightarrow (\exists x)(x \in A \land \#(A \setminus \{x\}) = n))$$

In a more procedural fashion relational overriding, $\oplus$, for any two relations $R$ and $S$ is defined as:

$$R \oplus S = (\text{dom } S \sqsubseteq R) \cup S$$

Hence, definition (15) is unfolded into the two formulæ (19) and (20):

$$(\forall u)(\forall v)(\forall w)(\forall x)(\forall y)(\forall z)$$

$$6\text{TUP}(u, v, w, x, y, z) \in \text{NDist}' \leftrightarrow$$

$$6\text{TUP}(u, v, w, x, y, z) \in \text{NDist} \land u \neq \text{id}?$$

The set $\text{NDist''}$ represents the entity (dom $S \sqsubseteq R$) in definition (18) above.

$$(\forall u)(\forall v)(\forall w)(\forall x)(\forall y)(\forall z)$$

$$6\text{TUP}(u, v, w, x, y, z) \in \text{NDist''} \leftrightarrow$$

$$6\text{TUP}(u, v, w, x, y, z) =$$

$$6\text{TUP}(\text{id?', name, addr, pv, bv, bonus}))$$

where name, addr, pv, bv and bonus represent the last 5 coordinates of a tuple from $\text{NDist}$.

Proof obligation (14) above boils down to showing that if we start by claiming that $\#\text{NDist} = k + 1$ (say), then $\#\text{NDist'} = k + 1$. OTTER fails to find any proof of this property using the standard definitions (16) and (17). One of the reasons for failure is that these definitions are not suitable for a proof of (14), especially not when we are dealing with sets of tuples instead of simple sets.

We start our attack on the problem by employing a reasoning strategy called resonance described in Wos [18]. The aim of the resonance strategy is to give preference (for directing a program’s reasoning) to formulæ that have the same syntactic shape (ignoring variables) as one or more of the patterns supplied by the specifier. Such formulæ are called resonators.

We, therefore, rewrite the cardinality definitions in terms of a tuple being removed or added to a set.

First, we give definition for removing a tuple from a set:

$$(\forall A)(\forall n)(\forall u)(\forall v)(\forall w)(\forall x)(\forall y)(\forall z)(\forall B)$$

$$(\text{Fun}(A) \land \# A = n + 1 \land$$

$$6\text{TUP}(u, v, w, x, y, z) \in A \land$$

$$(\forall u1)(\forall v1)(\forall w1)(\forall x1)(\forall y1)(\forall z1)$$

$$6\text{TUP}(u1, v1, w1, x1, y1, z1) \in B \leftrightarrow$$

$$6\text{TUP}(u1, v1, w1, x1, y1, z1) \in A \land u1 \neq u)$$

$$\rightarrow \# B = n)$$

Definition (21) could be explained as follows. For any given 6-tuple relations $A$ and $B$, if $A$ is a function and $\# A = n + 1$ then $\# B = n$, provided that $B$ is obtained from $A$ by performing a domain subtraction of a single tuple from $A$.

A resonance definition for adding a new tuple to a set is:

$$(\forall A)(\forall n)(\forall x)(\forall B)$$

$$(\# A = n \land x \notin \text{dom } A \land$$

$$\#(\text{Fun}(A) \land \# A = n \land$$

$$6\text{TUP}(u, v, w, x, y, z) \in A$$

$$\rightarrow \# B = n + 1)$$

Definition (22) states that for any sets $A$ and $B$ and any $x$ where $x \notin \text{dom } A$, if $\# A = n$ and $B$ is obtained from $A$ by adding a tuple with $x$ as the first coordinate, then $\# B = n + 1$.

We also need a resonance definition of a domain:

$$(\forall R)(\forall u)$$

$$(u \in \text{dom } R \leftrightarrow$$

$$(\exists v)(\exists w)(\exists x)(\exists y)(\exists z)$$

$$6\text{TUP}(u, v, w, x, y, z) \in R))$$

These changes allow OTTER to find a short proof for (14) in just 1.78 seconds. The input to the theorem prover appears in Appendix A.

5 Enhancements to the specification

A small enhancement can be made to the above specification.

Distributors sometimes pose queries which need not go through their uplines, but which may be addressed to the company directly. The company appoints some 'distributor relations coordinators' for this purpose and these coordinators are assigned to the top-level
uplines. If the group of distributors belonging to a top-level upline is not too large (according to the judgement of the company), then only one coordinator is assigned to such a group of distributors, but larger groups normally have more than one coordinator assigned to them.

One way of incorporating coordinators in our specification is to define a new function

\[ N\text{Coord} : ID \rightarrow \mathcal{P}(\text{Coord}) \]

where \( \text{Coord} \) represents the set of coordinators employed by the company. For any distributor identity code \( d \), if \( d \notin \text{dom}(N\text{Coord}) \) then \( d \) is not a top-level upline and therefore \( d \)'s top-level upline has to be determined first, whereafter the coordinator set for \( d \) is obtained. The set could be a singleton for a small network as mentioned above. Note that the variable \( N\text{Coord} \) would become a fifth component of the state, possibly further complicating proof attempts when reasoning about properties of the state.

6 Summary and future research

This paper presented a \( Z \) specification of part of a multi-level marketing business. In essence, specifying such a business boils down to specifying properties of trees and forests and operations on these. \( Z \) appears to be a suitable vehicle for this task. We illustrated how some proof obligations arising from the specification may be stated and discharged using an appropriate reasoning assistant. In the process we demonstrated the utility of two reasoning strategies from the literature, namely avoiding equality and using resonance.

The bonus calculation defined in schema \( \text{Gross\_bonus} \) is incomplete. Normally in a MLM business there are additional, more advanced bonus structures. For example, one such structure involves the idea of a '4000 point unit': A distributor is paid a bonus of 5\% of the business volume\(^7\) of a 4000 point unit on his or her first level, 2\% on the business volume of such a unit on the second level, and 1\% on the business volume of a similar unit on the third level. These percentages advance on a sliding scale with the top values fixed at 7\%, 5\%, and 2\% respectively. (See e.g. the section \( \text{The Marketing Plan} \), pages 5 — 6 in [6].)

Additional proof obligations arising from the specification can be identified and the suitability of using OTTER to discharge these can be investigated. For example, the following proof obligation can be formulated:

\[ \text{The sum of the gross bonuses allocated to first-level downline distributors is subtracted from the gross bonus allocated to their parent distributor in the network. Show, therefore, that the sum of the gross bonuses allocated to first-level downline distributors is less than or equal to the gross bonus allocated to their immediate upline.} \]

Formally, consider a distributor, say \( q \), with a number of first-level downlines, represented by a set, say \( D \), such that

\[ (\forall d)(d \in D \leftrightarrow (q, d) \in NU\text{plines}) \]

for \( NU\text{plines} \) a component of the MLM state space. Suppose the gross bonus allocated to a distributor, say \( i \), is indicated by \( \text{Bonus}_i \). The proof obligation is to show that

\[ \text{Bonus}_{q_i} = \sum_{(q, d) \in NU\text{plines}} \text{Bonus}_d \geq 0 \]

OTTER fails to find any proof of the above property. Therefore, an attempt at discharging this proof obligation with OTTER calls for the development of a set of heuristics for reasoning about arithmetic expressions, reminiscent of the heuristics developed in [16] for reasoning about expressions in set theory. Of course, one can also investigate the feasibility of using some of the well-known interactive term-rewriting reasoners like CaDiZ (see [14]) or Z-Eves, [10] to discharge the above proof obligation.

A rather simple way to solve this problem in one swell swoop in \( Z \) is to include additional conditions in the state invariant in schema \( MLM \) requiring that the value of every distributor’s bonus be non-negative and greater than or equal to the sum of the bonuses of that distributor’s immediate downlines. Nevertheless, the sensible specifier will always check (using some proof mechanism) that all relevant operations uphold this property.

7 Acknowledgement

The authors would like to thank the referees for making valuable suggestions regarding this article.
Appendix A

The complete input to OTTER for a proof of

$\#\text{NDist}' = \#\text{NDist}$

after operation $\text{Order}$ is given below.

```
%%%%%%Y.Y.Y.%%%%Y.Y.%%%%%% %%%%%%%%%%%%%
%% Prove $\#\text{NDist}' = \#\text{NDist}$ after $\%$
%% overriding an element. $\%$
%%%%%%Y.Y.Y.Y.Y.Y.Y.Y.%%%%%%%Y.%Y.%%%Y.Y.Y. Y.Y.Y.
%% set (neg_hyper_res).
%% set (factor).
%% set (ur_res).
%%
%% Paramodulation.
%%
%% set (para_from).
%% set (para_into).
%%
%% set (order_eq).
%%
%% Restriction strategies:
%%
%% set (process_input).
%% clear (para_from_right).
%% clear (para_into_right).
%% set (dynamic_demod_all).
%% set (back_demod).
%%
%% weight_list (pick_and_purge).

weight (x, 5).

end_of_list.
%
%% clear (print_given).
%% clear (print_kept).
%% clear (print_back_sub).

formula_list (usable).
%
%% Reflexivity.
%%
%% (all x (x = x)).
%
%% Resonance domain definition.
%%
%% (all R u
%% ( El (u, dom (R)) <-
%% ( exists v w x y z
%% El (6TUP (u, v, w, x, y, z), R) ))).
%
%% #NDist1 = DOMDIFF (NDist, {id}).
%%
%% (all u v w x y z
%% ( El (6TUP (u, v, w, x, y, z), NDist1) <-
%% ( El (6TUP (u, v, w, x, y, z), NDist) &
%% (6TUP (u, v, w, x, y, z) =
%% 6TUP (id, name, addr, pv, bv, bonus)) ))).

%% NDist is functional.
%%
%% Fun (NDist).
%
%% #Dist = k + 1.
%%
%% Card (NDist, $\sum (k, 1))$.
%%
%% Fact about NDist1.
%%
%% - El (id, dom (NDist1)).

end_of_list.

formula_list(sos).

%%%% \texttt{#NDist2 = k + 1.}
%%%% \texttt{-------------}
-Card(NDist2,$\sum(k,1))$.

end_of_list.

References
