Australian Safety Critical Systems Association

A National Special Interest Group of the AUSTRALIAN COMPUTER SOCIETY

Newsletter

June 2006

From the Chair

Since the last newsletter, the aSCSa Committee has been very active arranging the 2006 Conference and hosting a five day York University course - *Introduction to System Safety Engineering and Management* in association with the Australian National University (ANU). The course was presented by Dr David Pumfrey and Mark Nicholson of the University of York. I thank the ANU again for co-hosting this event through the Masters of Software Engineering program.

The aSCSa also participated in the ACS Queensland initiative *Technologists In the Public Interest (TIPI)*. As aSCSa chairman, I was invited to prepare a paper for the initiative to raise awareness and stimulate debate. I also participated in two workshops.

More information of the TIPI initiative and my paper are included in this newsletter.

I thank my aSCSa committee colleagues Chris Edwards, Clive Boughton and Kevin Anderson in reviewing my paper for TIPI.

George Nikandros
National Chairman

Association Matters

Annual General Meeting

The 2006/07 Annual General Meeting will be held on Thursday, 31 August 2006 in conjunction with the 2006 Conference at Eden on the Park, 6 Queens Road, Melbourne VIC 3004.

The meeting will commence 4:10pm.

The meeting will be asked to vote on the proposed change to the Association's Constitution Mission Statement. The proposed statement is:

> *Raise the awareness of the engineering and wider community of safety issues specific to software-intensive systems and to provide leadership and guidance in matters of safety.*

Continues Page 2

11th Australian Conference

MELBOURNE, 31 August – 01 September 2006

Eden on the Park, 6 Queens Road

Safe Software Architectures

The Australian Safety Critical Systems Association announces its 11th National Conference on Safety Related Systems. The 2006 conference will be held in Melbourne and will have a software architecture theme.

Critical functions for example safety, security, mission success and financial transactions are often entrusted to software intensive systems. Software architecture is the key to whether such systems can really be trusted.

Continuing the very successful format of recent annual conferences, a number of international Keynote Speakers will address this topical issue. International speakers include:

Klaus Marius Hansen
University of Aarhus, Denmark

David Garlan
Carnegie Mellon University, USA

Tim Kelly
University of York, UK

Jakob Gärtner
Esterel Technologies, Germany

Questions? More Information?

Dr Tony Cant (Program Chair)
Trusted Computer Systems Group
Information Networks Division
Defence Science and Technology Organisation
PO Box 1500, Edinburgh SA 5111 Australia
Phone: +61 8 8259 6700, Fax: +61 8 8259 5589
Mobile: (0412) 348 367,
Email: Tony.Cant@dsto.defence.gov.au

Mr Kevin Anderson (Conference Chair)
Kevin J. Anderson & Associates Pty Ltd
218 Danks Street
Albert Park VIC 3206 Australia
Phone: +61 3 8623 4091
Fax: +61 3 8623 4111
Mobile: (0412) 297 822
Email: kevin.anderson@hyderconsulting.com

See Page 2 for more details

This is a newsletter of the Australian Safety Critical Systems Association. The opinions expressed within are not necessarily those of the Association or of the Editor. Copyright for material included in this Newsletter remains with the Association and authors unless otherwise indicated.

Association Matters (continued)

Membership

Membership has grown significantly over 2005/06. It now stands at 107 (an increase of 29). However 45 members failed to pay their 2005/06 membership fees. There have been 2 resignations over the period.

The increase is largely due to the policy of offering an attractive discount to members participating in the Association's events and allowing participants to become members when registering for these events.

Membership renewal notices for 2006/07 have been issued.

National Committee

George Nikandros	Chairman
Kevin Anderson	Secretary
Chris Edwards	Treasurer
Tony Cant	Workshop Program Chair
Clive Boughton	Certification & Canberra Chapter Chairman
Robert Worthington	
Peter Hartfield	
Allan Coxson	
Alex Moffatt	
David Goedecke	
Web Site	www.safety-club.org.au

The term of the current committee expires 30 June 2006. As per the constitution the 2006/07 chairman is elected by the outgoing committee and all other committee positions are declared vacant. George Nikandros will continue as chairman for 2006/07 and all current committee members have agreed to continue.

Anyone interested in being a committee member is invited to contact the Asscoiation's Chairman by 31 July 2006.

Website

As previously reported the Association's website is limited to 10MB and hence is not sufficient to publish workshop presentations. In the December 2005 newsletter we advised that the larger resource items e.g. past conference presentations have been loaded on the aSCSa's resource repository hosted by the Australian National University.

Some links to the resource repository have now been provided. These links have been embedded within the program for the particular event. We expect to have a more direct link via resources webpage before the end of 2006.

Continues Page 4

Contents

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

School of Information Technology & Electrical Engineering

Development of Safety Critical Systems

3-day course

Next offering: 7-9 August 2006

Safety is a whole life cycle issue that relates to all aspects of the system. Hardware, software, operating procedures, planning, development, testing, maintenance, installation, commissioning, decommissioning, disposal and other aspects are considered in a safety program.

For most safety-critical systems, it is insufficient to simply develop a safe system; the system must be shown to be acceptably safe. The lecture component of this course explains the principles and practice of safety management and engineering and the unique challenges of computer-based systems. The content blends discussion of management and development issues with practical experience in safety analysis techniques. Topics covered include: hazard identification and risk analysis, safe system design, safety analysis techniques, safe software engineering, system hazard analysis, safety cases, safety management and human factors, and formal methods for system specification. Techniques covered include: Hazard and Operability Studies (HAZOP) and Computer Hazard and Operability Studies (CHAZOP), Fault Tree Analysis (FTA), Event Tree Analysis (ETA), Failure Modes and Effects Analysis (FMEA) and Failure Modes Effects and Criticality Analysis (FMECA), and Goal Structured Notation (GSN).

Assumed Background

It is recommended that participants have taken ENGG7000 or have had other experience of systems development and the system lifecycle. Familiarity with software engineering principles is desirable but not essential.

Cost & Venue:

$2200 incl. GST, course notes, lunch & refreshments

GP South (Bldg 78), The University of Queensland, St Lucia

To register:

contact Virginia Garton (07 3365 1003,
email virginia@itee.uq.edu.au)

Registration deadline: Monday, 1st August, 2006

For further information:

www.itee.uq.edu.au/~engg7020/DSCScourse.htm

2006 Conference

Once again the aSCSa will be hosting a conference along the same lines as the acclaimed conferences of Adelaide (2002), Brisbane (2004) and Sydney (2005).

The 2006 (11th) conference – see advert Page 1- has a software architecture theme. The two day programme will include four invited internationally renowned speakers:

- ***Klaus Marius Hansen*** – Dr Hansen is an Associate Professor at the Computer Science Department, University of Aarhus and Deputy Manager of the software area of the ISIS Katrinebjerg competency centre. Furthermore, he is scientific manager of the infrastructure group of the Danish national network for pervasive communication. His research areas include software architecture design and analysis, object-oriented modelling, techniques and tools for experimental object-oriented system development, and pervasive computing. Klaus received a Ph.D. in Computer Science from the University of Aarhaus in 2002.
- ***David Garlan*** – Dr Garlan is a Professor in the School of Computer Science at Carnegie Mellon University, where he leads several research projects and is the Director of the Master in Software Engineering Programs. He received his Ph.D. from Carnegie Mellon University in 1987. His research interests include software architecture, ubiquitous computing, self-adaptive systems, formal methods, and software development environments.
- ***Tim Kelly*** – Dr Kelly is a Lecturer within the Department of Computer Science at the University of York. He is also Deputy Director of the Rolls-Royce Systems and Software Engineering University Technology Centre funded at York. His research interests include safety case management, software safety analysis and justification, software architecture safety, certification of adaptive and learning systems, and the dependability of "Systems of Systems". He has published over 70 papers on high integrity systems development and justification in international journals and conferences. He is also Managing Director of Origin Consulting (York) Limited – a consultancy company specialising in safety critical systems development and assurance.
- ***Jakob Gärtner*** – Jakob is the Technical Director of Esterel Technologies (Germany). He specialises in formal methodologies, automatic certified code generation, and open system architectures. His work builds on experience in aerospace, rail and marine projects, and solid computer science.

Included in the programme is a dinner function on the Thursday evening. This will be an excellent opportunity to network in a relaxed atmosphere.

Also included is a post-conference event to allow visitors to experience the Melbourne hinterland.

For registration and programme details, please visit the Association's website at www.safety-club.org.au. Unlike recent conferences this year's conference registration does have a provision to join the aSCSa to claim the member discount.

Southwest Airlines Flight 1248 crash

Source: Risks-Forum Digest Volume 24, Issues 15 and 16 located at:

http://catless.ncl.ac.uk/Risks/24.15.html and http://catless.ncl.ac.uk/Risks/24.16.html

Post Date: Fri, 27 Jan 2006
Posted by: Joe Thompson

Post Date: Tue, 31 Jan 2006
Posted by: Peter Ladkin

The NTSB has issued an advisory A-06-16 as a consequence of the Southwest Airlines Flight 1248 crash in Chicago on December 8, 2005. The Southwest airline flight 1248 on landing at Midway Airport in a snowstorm rolled off the end of the snow-contaminated runway -- where it tore through two fences and stopped in an intersection, hitting two cars. A 6-year-old boy in one of the cars was killed.

According to Joe Thompson the NTSB reported that the thrust reversers did not deploy properly, causing the plane to overshoot the end of the runway.

The posting by Peter Ladkin gives more insight into the mishap. According to Peter Ladkin, the pilots had used an "on-board laptop performance computer (OPC)" to calculate landing distances to determine whether they could land at Midway in the snow-stormy conditions. The crew input weather data and entered runway braking conditions as "WET-FAIR" in the OPC. The OPC calculated that the airplane would be able to land and completely stop with 560 feet of runway remaining. However, "the OPC is programmed to assume that the engine thrust reversers will be deployed on touchdown" and they were not so deployed. They deployed 18 seconds after touchdown. "If the reverse thrust credit had not been factored into the stopping distance calculations made by the OPC, it would have indicated that a safe landing on runway 31C was not possible under a braking condition of either fair or poor". A point of contention right after the accident was that the pilots had apparently activated the automatic brake system in violation of Southwest policy, but the NTSB concluded the crucial factor was the unanticipated 18-second delay in the thrust-reversers deploying. As a result, NTSB is urging the FAA to prohibit allowing for thrust-reversers in onboard stopping-distance calculations. (Before landing, the crew had used the onboard computer to calculate stopping distance for "wet-poor" conditions; those calculations assumed the thrust reversers would deploy normally.)

In other words, an implicit assumption made by the OPC program led to the OPC indicating to the pilots that they could land safely on runway 31C when, under the conditions that actually obtained during the landing, the OPC program would have indicated that they could not do so without overrunning.

The reasons for the delayed deployment of reverse thrust have not yet been publicly determined by the Board.

Pilot reliance on the OPC seems to be a key causal factor in the mishap. Without any knowledge of the OPC, one can only speculate as to the rigour of its development or the understanding by its developers of the consequences of reliance upon it.

Association Matters (continued)

Finance

As of June 30, 2006, the Association has some $70K in accumulated funds. These funds are not the profit made over the year, but are the funds that have been accumulated over the last 10 years; 7 of those years as the ACS National Technical Committee on Safety-Critical Systems.

The aSCSa essentially broke even over this financial year.

The committee intends to continue to use these funds for conferences and educational activities. The association's administration costs will rise as the reliance on the committee members' employer organisations reduce.

Affiliations

The aSCSa Committee has joined the System Safety Society as a corporate member. The System Safety Society is a non-profit organisation supporting safety professionals worldwide. With a wide range of individual and corporate members, the Society is affiliated with major corporations, educational institutions and other agencies.

As a corporate member, we get our logo and a link to our website, hopefully raising our profile internationally.

Education - Safety Critical Systems

Again this year aSCSa and the ANU organised the High Integrity Systems unit from the University of York to present a 5 day intensive course on Systems Safety Management. The course is an elective within the ANU Masters of Software Engineering program and industry participants are encouraged to attend through advertising by aSCSa. This year there were 7 MSE students and 20 industry participants from various organisations such as Boeing, CSC, CSR, Dept. of Defence, SAAB, Thales, Union Switch and Signal, and Westinghouse. University of York sent two presenters in David Pumfrey and Mark Nicholson. Workshops were run with the help of Clive Boughton, Brian Molinari, Malcolm Newey, and Gordon Stone.

As with last year, the course was well received by all. A couple of the industry participants decided to attempt the assignment that is used to assess the masters students. They have 6 months to hand in their attempts if they want them assessed. All industry participants will be receiving certificates in the mail.

The course will be running again next year (see advert) at around the same time at ANU. Once the date has been finalised, aSCSa will advertise for industry participants. There were no prerequisites for participation.

Technologists in the public interest

Source: Information Age, April/May 2006 and June/July 2006.

In response to a sustained decline in interest in ICT in schools and universities and acknowledging the critical importance of ICT for economic growth, the Queensland State Premier, Peter Beattie announced a national skills summit, an initiative of the Queensland Minister for ICT Policy, Chris Cummins. Sharing these concerns, the Premier also provided the ACS with an opportunity to lead a separate event *Technologists in the public interest (TIPI)*, a phrase coined by ACS President Philip Argy and initiated by the ACS Queensland Chairman, Mark Lloyd.

TIPI seeks to promote debate in the ICT community about what should be the minimum codes of conduct, standards and training for ICT workers. The outcome expected is:

- A cohesive identity – something better than geek, nerd, propeller-head, computer professional, system analyst;
- A competence framework – what are the base skills, what are the skills required for particular specialisations?
- An ethical framework – is there a need for an enforceable code of ethics?
- A governance structure – many associations represent the ICT industry; there is no peak body to ensure competence and ethics.

Research papers have also been completed and these act as resource material and thought starters for attendees at various workshops. The papers cover Safety Critical Systems, Security Certification, Ethics, and how traditional professions (Engineering, Financial Planners, Law and Teaching) have addressed the issues that TIPI is attempting to address.

The Premier of Queensland has planned to deliver his annual address to the ICT industry on August 11, with the National Skills Summit and TIPI featuring prominently.

2005 Workshop

The papers for the 10th Australian Workshop on Safety Critical Systems held in Sydney on 25-26 August 2005 have been published and are available at the ACS Conferences in Research and Practice in Information Technology website (http://crpit.com/Vol55.html).

Hard copies will be distributed to members.

Introduction to System Safety Engineering and Management
(Content to be confirmed)

Day 1	• Introduction and Safety Concepts • Development for Safety • Preliminary Hazard Identification & Case Study • Modelling Event Sequences • Case Study: Chemical Containment Fault Tree • Risk Assessment
Day 2	• Functional Hazard Assessment • Case Study: ARP4761 WBS FHA • HAZOP • Case Study: Process Plant HAZOP • Systematic failure • Safety Integrity levels
Day 3	• Safety Analysis techniques 1 • Case Study: AGV Fault Tree and FMEA • Safety Cases 1 • Case Study: Safety Case Construction • Safety Cases 2
Day 4	• Safety Analysis Techniques 2 • Preliminary System Safety Assessment • Case Study: ARP 4761 WBS PSSA and SSA review • Common Cause Analysis • Safety case: Common Causes • Introduction to Software Safety
Day 5	• Safety Management • Case Study: AGV Safety Management • Human factors • Safety Culture • Conclusions • Bibliography • Glossary

Australian National University

April 2007

Registration

(to be advised)

Contact aSCSa Secretary to register interest and for more information

Early bird and group discounts

Bulletin Boards

ACM Risk Forum On Risks To The Public In Computers and Related Systems – http://catless.ncl.ac.uk/Risks.

Safety-Critical Mailing List Forum hosted by the University of York. Need to join using the form located at www.cs.york.ac.uk/hise/text/sclist/form.php for access.

ICT When it really has to matter!

This paper was prepared for reference material for the ACS Technology in the Public Interest initiative.

Author: G. Nikandros, aSCSa, submitted April 2006

Information and Communication Technology - When it really has to matter!

Abstract

In this paper I discuss the ever-increasing reliance of Information and Communication Technology (ICT) for mission critical and safety-related systems. The flexibility of the technology entices its use for applications never before contemplated. However, can or should ICT be trusted for such applications and if so, how?

Examples are cited where ICT was a significant contributor in the loss of life and/or mission failure and considers the effectiveness of regulation within Australia in the control of such risks.

Keywords: information, communication, technology, ICT, mission, critical, safety, regulation

Introduction

> *If a builder has built a house for a man and his work is not strong and the house falls in and kills the man, then the builder shall be slain.*
>
> Code of Hammurabi, 2150 BC (Underwood, 1996)

If such a code applied to ICT systems today then perhaps the enthusiasm to exploit the technology would be somewhat curbed.

ICT has moved beyond simple administrative functions; it now runs the family car, microwave oven, washing machine and even amusement park rides, to name but a few. The train you travel on or the plane you fly both depend on ICT to get you to your destination safely. You even rely on ICT to correctly process your “000” emergency call (Nikandros, 1998).

In fact, ICT systems are now controlling many complex processes in industry. Industry examples include chemical processing, manufacturing, transport, power generation and distribution, medical devices, telecommunications, mining machinery, and fire and protection.

Yet despite this ever-increasing reliance “bugs” are still regarded as being synonymous with ICT – often up to 10 per thousand lines of code. In no other “product”, is the community more tolerant of defects - so much so that terms like “good enough software” are often used.

What can go wrong?

There have been numerous instances where ICT was considered to have been the significant contributor to the failure directly leading to substantial loss. Two of the more publicised examples are the Therac-25 radiation therapy machine and the Ariane 5 rocket launch vehicle.

Therac-25

Between 1985 and 1987 the Therac-25 massively overdosed 6 people. The direct cause was the re-use of software from an earlier model, the Therac-20. However, unlike the earlier model, there were no hardware safety interlocks. These safety interlocks effectively masked the software errors in the Therac-20, consequently the software was assumed to have had a proven safe history and therefore was accepted as is, without the safety interlocks. (Standards Australia HB220-2000)

Ariane 5

In 1996 an Arianne 5 exploded during the launch phase and resulted in the loss of a communications satellite. The explosion was blamed on a complete loss of guidance and altitude information 30 seconds after lift-off. The guidance system gave a wrong command to the boosters. The cause according to the official report was that "the loss of information was due to specification and design errors in the software of the inertial reference system. The extensive reviews and tests carried out .... did not include adequate analysis and testing of the inertial reference system or the complete flight control system, which could have detected the potential failure." (Standards Australia HB220-2000)

Software from Ariane-4 had been used in Ariane-5 without testing. When subjected to higher accelerations, the software (calibrated for Ariane-4) ordered an "abrupt turn 30 seconds after lift-off", causing the airframe to fail. Apparently, conversion from a 64-bit floating representation to a 16-bit signed representation caused an Operand Error. (Neumann – New Material)

Other Examples

The following paragraphs are further examples of ICT failures:

- On January 23, 2003, a Singapore Airlines (SIA) Boeing 747-400 experienced a complete loss of information on all six integrated display units (IDUs) while in cruise flight from Singapore to Sydney, Australia. The pilots flew the airplane for 45 minutes using standby flight instruments, namely an altimeter, airspeed indicator, and artificial horizon/attitude indicator i.e. no traffic alert and collision avoidance system, enhanced ground proximity warning system, or weather radar. (National Transportation Safety Board, 2003)
- Der Spiegel Issue 12/1999, page 226 gives an analysis of an accident in which a baby was killed by a deploying air bag in a Volkswagen Golf after having been disabled in a certified garage. Deactivation is a software control function. However if the system self-checking detects an error, the current configuration is automatically replaced by back-up software held in ROM. The back-up only knows simple rules e.g. deploy all air bags on impact. (Risk Digest Volume 20: Issue 28)
- A woman in Düsseldorf, Germany, told the court that she had been erroneously informed of having incurable syphilis and had passed it on to her daughter and son. As a result, she strangled her 15-year-old daughter and attempted to kill her son and herself. She was acquitted. The insurance company said the medical information had been based on a computer error – which could have been a lame excuse for human error. (Neumann)
- As many as 20 deaths may have been attributable to the London Ambulance Service's inability to dispatch ambulances in response to emergencies. After severe difficulties in system development, including repeated test failures, the system was finally placed in operation. The system then collapsed completely, including worst-case delays of 11 hours. "An overcomplicated system and incomplete training for control staff and ambulance crews are the likely causes...." (Neumann)
- In late Spring of 1983, there was serious flooding from the Colorado River resulting in six deaths and damages costing millions. The problem was traced to a bug in the computer program that had modelled the flood process and predicted how much water should be stored. (Neumann)
- A payroll blunder left the Brisbane City Council responsible for bus drivers incurring late mortgage fee and bank fee penalties as the result of 1400 bus drivers not being paid on time. The council has blamed a computer payroll error for the mistake. Brisbane ratepayers will be required to cover the cost of the late payment penalties incurred. This was the second major payroll problem in seven months. (Courier Mail, 2006)
- The Tokyo Stock Exchange suffered its worst ever outage on 02-Nov-2005 when trading was suspended for four and a half hours due to a software problem. The glitch appeared to be connected to the decision to expand the trading system's capacity the previous month in response to high trading volumes. The modified system had worked well, but crashed when the automatic monthly clean-up of the software was implemented. A back-up system also failed because it also used the same software. (Risk Digest Volume 24, Issue 9)

Software Flexibility

Software is an "intangible" - by itself it cannot cause harm. Not being a "physical" artefact, software developed for one purpose can be used for other purposes, beyond which the designer originally envisaged. These other purposes may have much higher safety risk compared to the purpose for which it was originally designed. (Standards Australia HB220-2000)

To illustrate this point, who would have envisaged that the Microsoft® Access data base application would be the underlying software platform for a legal euthanasia machine?

In 1995, the Northern Territory amended the *Rights of the Terminally Ill Act* to allow a terminally ill person experiencing pain and suffering to request their doctor to assist them in terminating their life.

Dr Philip Nitschke, assisted by a computer technician, developed such a device after Australia's Northern Territory passed that law. It consisted of a laptop computer loaded with the *Deliverance* software, a syringe driver and other standard medical components. The patient operated it via the keypad. Answering 'Yes' to a series of questions led to the release of a fatal injection. Between 1996 and 1997 four people were legally allowed to use the machine before Australia's Federal Parliament overturned the controversial law.

According to Uhlig and Martin, 1996, the computer program, *Deliverance*, checked that a patient realises what he or she was doing before administering a lethal dose of barbiturate. It used an adaptation of Microsoft® Access, a database program. Once all the questions are answered affirmatively, a signal goes from the computer's parallel port - normally used to connect a printer - via a relay switch to an air compressor that pushes the plunger of a syringe containing the appropriate drugs.

One can only wonder what would have happened if the "Print Screen" key was pressed (accidentally or deliberately). Would this have compromised the safeguard provided by the question sequence? What about the integrity of the electronic interface between the parallel printer port and the powered syringe delivering the deadly drug cocktail? What if all the questions were answered and the syringe failed to operate? Imagine the psychological stress on the patient.

Software – the case for regulation?

Leveson (1992) provides a strong case as to why the development of software should be regulated. Her argument parallels the development and use of steam power to that of software highlighting the similarities. The following paragraphs in italics are from Leveson, 1992.

James Watt, through his research and development of steam engines, had patented several important ideas which prevented others from building rotating steam engines.

The Watt engines used low pressure steam which limited both their efficiency and economy. High pressure would have permitted more powerful engines, but Watt opposed it on the grounds that it increased the danger of explosion and thus constituted an unacceptable risk.

Watt's patents expired in 1800 and such high pressure engines soon made their appearance. Steam power was transforming industry and therefore very important to the economy and national growth. Until 1800, such growth had been constrained by Watt's patents.

Death and injury increased significantly following the increasing spread of the use of high pressure steam engines in steamboats and industrial plants. These often resulted in disastrous explosions.

Boiler technology lagged the development of the engines. Engineers quickly amassed scientific information about thermodynamics, the action of steam in the cylinder, the strength of materials in the engine, but had little scientific understanding about the build up of steam pressure in the boiler, the effect of corrosion and decay and the causes of boiler explosion.

The early steam engines used inferior materials; they had low standards of workmanship; the mechanics lacked proper training and skills; and there were serious problems with quality control.

In England, Watt's campaign against high pressure engines supported by well publicised accidents slowed their adoption.

In the USA between 1816 - 1848, a total of 233 steamboat explosions had occurred in which 2562 persons had been killed and 2097 injured with property losses in excess of $3M (1840s value).

Watt and others were correct in their beliefs, that standards were essential in the design, manufacture, and operation of steam engines.

These high standards were finally enforced in Britain in the latter period of the nineteenth century. Boiler explosions dropped to 14 deaths per year (in 1905) as compared to 383 in the USA. Eventually USA enforced such standards.

Society is now in the computer age and it is now again faced with a technology for which there are great economic incentives to push the state of the art and to use this technology to control dangerous systems.

Computers, like steam engines and electrical systems, give users the ability to accomplish things we could not accomplish before. And again, it appears that the risks could increase over time as computers take over more and more functions.

Like boilers, the scientific foundations of the software engineering field are still being developed. Changing from an art to a science requires accumulating and classifying knowledge. Although this is happening, more effort is being expended on new inventions and building tools for unproven techniques.

The paper continues to discuss the need for those involved in the computer industry to understand the hazards associated with the systems they are building and the need for appropriate skills and knowledge to manage the development and use of these systems.

Software – legal issues

The following paragraphs are from the December 2004 edition of the Australian Safety Critical Systems Club's newsletter article titled *Software and the law*.

Society's tolerance towards "buggy" software is very much contrary to its tolerance to faulty products and services in general. When it comes to software it seems, society is prepared to tolerate defects as long as the software generally provides the functionality expected.

As a consequence of society's complacency in relation to software, lawmakers have tended to shy away from the complex issue of software liability.

In an article Beware of Faulty Software published in Engineers Australia Magazine (July 2004), David Neiger, a mechanical engineer and lawyer, provides some valuable insight into the peculiarities of software in relation to the Australian Trade Practices Act and the various State Fair Trading Acts.

Whilst statute law in relation to negligence is clear, in that designers of faulty products are liable for any *reasonably foreseeable loss or damage that arises for the use of the products, these laws are not so clear in relation to software – is software a product?*

> *"Most cases involving computer software rely upon intellectual property rights such as copyright, patents and design. No one thought of software as a good….."*
>
> *"At first, object code (the 1s and 0s stored in a ROM or on floppy disks) was not thought of as a property at all because the judges did not consider the electrical charges in a ROM or magnetic pulses on a disk to be a "literary work" worthy of protection. However, source code, which could be read by a human was considered to be a literary work and was protected under copyright law. In 1984, after an appeal, the judges were finally convinced that both source and object (machine) code were literary works that could be protected under copyright. This was enshrined in legislation with the Copyright Amendment (Digital Agenda) Act 2000."*
>
> *"While computer programs are considered literary works, they are not covered by the same rules of product liability as physical goods or services. The way the law is presently, the vendors license you to use the computer program in accordance with the conditions of the End User Licensing Agreement (EULA) contract."*

By making the use of the software conditional on 'voluntary' acceptance of a EULA, liability is effectively transferred from the software vendor to the software end-user. However goods and services involving the use of the software are very much subject to the Trade Practices Act. Further, engineering software tools, being 'literary works' would not be regarded as consumer goods, and as such would not have the statutory warranty protection provided by the Trade Practices Act.

David Neiger sums up the issue thus:

> *Ultimately, as engineers, we are responsible for the output of any computer programs, so if your CAD package or machine control software fails and your designs are faulty, you, rather than the software vendor will be held liable.*
>
> *If you write software, you impose any conditions you want in your EULA, so you might as well exclude everything. And be particularly careful if you supply goods that rely on software as you may still be liable, even if the software is at fault unless you have negotiated a different contract with the software vendor.*

It follows of course, those procurers who specify (mandate) software packages to be used in relation to the goods and services to be supplied incur liability arising from the use of the software.

Standards Australia (HB220-2000) gives a concise and clear explanation of the laws in relation to liability, namely the Common Law Tort of Negligence and the Commonwealth's Trade Practices Act – Product Liability.

The Common Law Tort of Negligence

The Common Law Tort of Negligence is available to anyone at anytime. However it can only be applied when:

- harm has occurred; ***and***
- a "duty of care" existed towards the injured party; ***and***
- the expected "standard of care" was not met; ***and***
- the harm was a foreseeable consequence of the breach of care.

A "duty of care" is owed to anyone who might have been foreseen as affected by the consequences of your act or omission. The "standard of care" is the standard that should be exercised by an ordinarily competent and diligent member of the profession.

In such actions, the courts will look to industry and professional standards to determine the "standard of care".

Trade Practices Act – Product Liability

Like Common Law, contracts cannot exclude liability under the Trade Practices Act. The developer/supplier and procurer have obligations under the act. Under the act, in terms of safety, *"....goods are defective if they do not provide the degree of safety which persons generally are entitled to expect [in all the circumstances]....*". It includes not only the specified use of the product, but also reasonably foreseeable misuse.

The goods covered by the Act include systems containing software. This means that developers and procurers of such systems are legally liable for their safe operation throughout their life. It should be noted that if the goods were defective solely because they complied with a mandatory standard, or that the state of scientific knowledge was such that the defect could not be discovered at the time of the sale, then the supplier has a strong case against a claim under the Trade Practices Act.

Standards

Standards and their compliance are essential in minimising the risk of legal action.

Numerous standards (international, some adopted by Australia) now exist, both for general and industry specific ICT mission-critical and safety-related applications. Because the state-of-the-art in relation to ICT is still very much evolving, the standards somewhat differ in their requirements. This proliferation of conflicting standards has created a situation where developers cannot be certain that the particular standards adopted will satisfy a future court that the standard of care was appropriate at the time. After all Courts are required to adjudicate after some loss has occurred; this may be long after the critical decisions were made. They also have the benefit of hindsight.

All standards acknowledge that rigorous testing alone is insufficient due to the complexity of ICT systems. The nature of ICT is such that defects are difficult to detect; they lie dormant until the conditions arise to reveal them, often with serious consequences. Civil and mechanical engineers, for example, use well-established continuous models supported by extensive data that enable failure predictions with some accuracy.

Standards for ICT systems are generally process focussed. The theory is that the application of robust and rigorous processes correctly applied to all phases of the system life cycle will strongly support a claim that the ICT system is appropriate for the intended application.

However processes alone are not sufficient; there needs to be the proper framework to support those processes. Key is the definitions of roles and responsibilities and ensuring that those so assigned have the necessary competence and required independence.

Competency requirements for ICT practitioners

Many in the ICT industry today remain largely unaware that standards exist; they don't know what they don't know. This goes for both organisations as well as individual ICT practitioners.

The underlying premise of standards relating to safety/ mission critical ICT, is that a system being developed has to be considered critical until justified otherwise. Also, today's practitioners usually consider only the functions that a system has to perform; not the functions that it must not.

Software functionality is virtually limitless – it is the processing hardware which limits functionality – and software is often made complex so as to allow its use for many varied applications. Much of today's software is able to be customised through e.g. configuration parameters, add-ins etc. This flexibility substantially increases the state space so as to make complete testing impossible for all but trivial applications.

Much of today's software development (customisation/ integration) involves the use of commercial-off-the-shelf (COTS) ICT. However the intended use may be for an application not originally considered by the COTS product developer. The user of the COTS product is most likely not aware of its limitations. More disturbing is the increasing use of software/system-of-unknown-pedigree (SOUP); as not only are the limitations unknown, but there is no evidence of its quality.

Legislative requirements

In Australia, there is currently little regulation in relation to the safety/mission criticality of ICT. Whilst there are laws covering safety, these laws are very much focussed towards workplace health & safety (WH&S). Apart from the laws governing drugs, poisons and therapeutic goods, there is little in relation to public safety.

Of those practitioners involved in the ICT industry only professional engineers are subject to regulation and that regulation only seems to apply in the state of Queensland. Queensland has the Professional Engineers Act 2002, which is supported by the Professional Engineers Regulations 2003. The Act and the Regulations are controlled by the Board of Professional Engineers, Queensland. The Act and supporting regulations require those so registered to work within their nominated engineering area of competence. However no competency requirements are specified in order to satisfy registration. All that is required is that applicants meet the qualifications for Membership of the Institution of Engineers Australia (Engineers Australia) and have the designated years of experience.

Engineers Australia operates a National Professional Engineers Registration (NPER) scheme. This is a voluntary scheme and there is no legislation which requires NPER for those engineers involved in the ICT industry. In any case NPER does not define competency requirements.

Professional engineers would only be a small proportion of the ICT practitioners within Australia.

In fact there is little by way of definition of competency requirements for safety/mission critical related ICT worldwide. The UK Health & Safety Executive (HSE) commissioned the UK Institution of Electrical Engineers (IEE) to undertake preliminary work in relation to determining competency requirements. This led to the development of the Competency Guidelines for the Safety-Related System Practitioners. These guidelines, published in 1999, were the result of collaboration between the IEE and the British Computer Society (BCS). These guidelines were developed so that UK Industry is able to demonstrate competence of individuals involved in the development and operation of safety-related systems.

Establishing competency requirements is one thing, there needs to be the training programmes to enable gaining of these requirements and certification that these have been attained.

Training for ICT practitioners

In 2001, the Australian Computer Society (ACS) through its National Technical Committee on Safety-Critical Systems[1] (ACS-SCSC) commissioned a study to determine the requirements for an introductory course for developers and procurers of safety-related systems. The intent of this course was to provide practitioners with key basic skills in relation to the existence of standards and the ability to understand the requirements. Basically, so that those involved in ICT will know what they don't know and where to go for information. Boughton, 2002 provides us with the results of the study. In essence there were no courses available that addressed the topics considered necessary. In fact only the UK, USA and Germany had appropriate learning centres. However, most safety-critical specialists and courses reside in the UK and USA; hence the available courses are not practical for those residing in Australia.

The ACS-SCSC attempted to establish an elective module within the ACS CMACS (Certified Member of the Australian Computer Society) programme for safety / mission critical related systems. However it was not considered to be commercially viable due to the likely limited interest.

To fill this void, the Australian Safety Critical Systems Association (aSCSa) in association with the Australian National University (ANU) are offering the course *Introduction to System Safety Engineering and Management*. This is a 5 day course developed and

[1] The National Technical Committee on Safety Critical was established under the ACS Software Engineering and Computer Science Board in 1991. In 2002, the national committee evolved into the Australian Safety-Critical Systems Club, as a National Special Interest Group of the ACS. In 2005, the Club amended its name to that of the Australian Safety Critical Systems Association (aSCSa).

delivered by the High Integrity Systems Engineering (HISE) group at University of York. This course is held at ANU and is part of an ANU master's degree program. It is available to industry participants.

In relation to continuing professional development (CPD) and to raise awareness, the aSCSa has conducted annual workshops, the 11th to be held in Melbourne later this year. For the more recent workshops, the aSCSa has had a number of international experts in a related field as invited speakers. Information on these workshops and other activities can be found on the aSCSa's website www.safety-club.org.au.

Conclusions

Boughton, 2002 opined:

> *Within industry world wide there seems to be a great deal of ignorance about evaluating, constructing, managing and maintaining, software-based safety critical systems. This is especially so in Australia.*

Given the frequent media reports relating to ICT system failures, it's an opinion that has some justification. One merely has to visit the The Risk Digest.

Unfortunately within Australia, ICT practitioners don't know what they don't know. The ACS through the aSCSa is endeavouring to raise awareness so that ICT practitioners at least know what they don't know.

Fortunately, Australia has been spared the serious accidents cited in the paper. This however has very much contributed to the high level of ignorance within the ICT industry.

The legal status of software and the limited legislative framework will continue to frustrate the implementation of appropriate practices and ensuring that only those having the appropriate competence are associated with safety/mission critical related systems.

The phrase *fly-by-wire* emerged from within the air industry to refer to the replacement of the mechanical controls in an aircraft with distributed controllers interconnected electrically. Such commercial aircraft have been flying some 10 years now without a major mishap. That does not mean of course that critical failures have not happened; it means that no such aircraft has fallen out of the sky.

If the use of ICT in the car continues to evolve unabated it won't be long before there is a *drive-by-wire* model available. Imagine driving a car where the steering wheel is no longer mechanically connected to the front wheels; all there is a joy-stick like device with an input to a computer. It is this computer that points the front wheels in the required direction. Bear in mind that the software that does the control is considered to be "artwork".

References

Underwood, A. (1996): Computers and Safety. Seminar -*Computers and Safety* - ACS Technical Committee on Safety-Critical Systems and Software Verification Research Centre, The University of Queensland.

Nikandros, G (1998): ACS Article "Raise the Standard on Safe Software", The Australian, 17-Nov-1998.

Standards Australia (2000): Handbook *Safety issues for software*, HB220-2000

Neumann, PG Computer-Related Risks: New Material http://www.csl.sri.com/users/neumann/risks-new.html

National Transportation Safety Board, USA: Safety Recommendation A-03-55 and –56, 02-Dec-2003

The Risk Digest, Forum on risks to the public in computers and related systems, moderated by Neumann, PG, http://catless.ncl.ac.uk/Risks

Neumann, PG: Computer Related Risks, ACM Press / Addison Wesley

Corkill, M, Courier Mail (Australia), News article *Bus drivers pay dearly for errant computer*, published 06-Apr-2006.

Uhlig R, and Martin G, Telegraph (UK), News article *The man and his machine*, published 19-Apr-1996.

Leveson, N (1992): High-Pressure Steam Engines and Computer Software. *Proc. International Conference on Software Engineering*, Melbourne Australia, **14**:2-14, ACM Press.

Australian Safety Critical Systems Club, Newsletter article *Software and the law*, Dec-2004

Commonwealth of Australia, Trade Practices Act 1974

The Institution of Electrical Engineers, UK: Competency Guidelines for Safety-Related System Practitioners, 1999.

Boughton, C (2002): Beginning to Define a Body of Knowledge for Safety Practitioners. *Proc. Seventh Australian Workshop on Industrial Experience with Safety Critical Systems and Software*, Adelaide Australia, October 2002, Ed Lindsay, P, pages 31-40, *Conferences in Research and Practice in Information Technology, Volume 15*, Australian Computer Society Inc.