2ND WORKSHOP ON

Complexity in Design and Engineering

EDITOR: CHRIS JOHNSON

GIST TECHNICAL REPORT G2005-1,
DEPARTMENT OF COMPUTING SCIENCE, UNIVERSITY OF GLASGOW,
SCOTLAND.

ACKNOWLEDGEMENTS

We wish to thank the European Office of Aerospace Research and Development, Airforce Office of Scientific Research, United States Air Force Laboratory for their contribution to the success of this conference.

The workshop has also been supported by the EC ADVISES Research Training Network.

We also acknowledge the support of the following organisations:

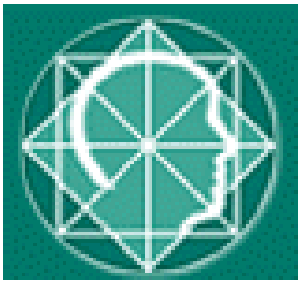


TABLE OF CONTENTS

WORKSHOP TIMETABLE

THURSDAY 10TH MARCH

09.00-09.30 C. Johnson	Welcome and Introduction.
09.30-11.00 Chair: Peter Johnson, Dept of Computing Science, Univ. of Bath, UK.	**Paper Session 1: Change, Complexity and Planning** *Abstracting Complexity for Design Planning* David Wynn, Claudia Eckert and P John Clarkson, EDC, University of Cambridge. *Design Change and Complexity* Chris Earl, Claudia Eckert*, John Clarkson*, Department of Design and Innovation, Open University. *Engineering Design Centre, University of Cambridge. *Complexity of Design in Safety Critical Interactive Systems* Sandra Basnyat, David Navarre, Philippe Palanque, LIIHS-IRIT, France.
11.00-11.30	*Coffee*
11.30-13.00 Chair: Philippe Palanque, LIIHS-IRIT, France.	**Paper Session 2: Creativity, Diversity and Design** *Creativity in the Design of Complex Systems* N. Maiden and S. Jones, Centre for HCI Design, City University, UK. *Diversity as a Determinant of System Complexity* B. Sherwood-Jones and P. Anderson, Digital Design Studio, Glasgow School of Art. *Design of the ICT Infrastructure of an Educational System* P. Baquero, R.M. Aguilar, A. Ayala, Univ. of La Laguna, Spain.
13.00-14.30	*Lunch*
14:30-16:00 Chair: Alistair Sutcliffe, School of Informatics, University of Manchester, UK	**Paper Session 3: The Human Factor in Design Complexity** *Uncovering the Information Needs in Complex Aerospace Systems* Iya Solodilova and Peter Johnson, Department of Computer Science, Univ of Bath. *Validating a Process for Understanding Human Error Probabilities* Richard Maguire, SE Validation Ltd, Salisbury, UK. *Design of Complete Systems: Developing Human Factors Guidance for COTS* Anne Bruseberg, Systems Engineering and Assessment Ltd. Somerset, UK.
16:00-16:15	*Tea*
16:15-17:45 Chair: Nick Chozos, Dept of Computing Science, University of Glasgow.	**Paper Session 4: Applications** *Sources of Complexity in the Design of Healthcare Systems* T.-B. Azzelarabe, D. England, P. Misedine, K. Murphy, M. Randles School of Computing and Mathematical Sciences, Liverpool John Moores University *Automation, Interaction and Complexity* Robert L. Wears, Dept of Emergency Medicine, University of Florida. *What Makes Emergency Ambulance Command and Control Complex?* B.L. William Wong, Jared Hayes*, Tony Moore*, Dept. of Computing Science, Middlesex University, *Department of Informatics, University of Otago, New Zealand.
18:00-20:00	**Informal Reception**

FRIDAY 11TH MARCH

09.00-09.30 C. Johnson	*Welcome and Coffee*,
09.30-11.00 Chair: Iya Solodilova, Dept of Computing Science, Univ. of Bath, UK.	**Paper Session 5: Emergent Properties?** *Understanding the Root Causes of Complex Security Incidents* Chris Johnson, Glasgow Accident Analysis Group, University of Glasgow, UK. *Complexities of Multi-Organisation Error Management* John Dobson, Simon Lock and David Martin, University of Lancaster. *Emerging Complex Interactions: Safety Analysis in Air Traffic Managememt* Massimo Felici, LFCS, University of Edinburgh.
11.00-11.30	*Coffee*
11.30-13.00 Chair: Claudia Eckert, Engineering Design Centre, University of Cambridge, UK	**Paper Session 6: Stakeholders, Conflict and Complexity** *Extending Small Group Theory for Understanding Complexity* Alistair Sutcliffe, School of Informatics, University of Manchester, UK. *Systems Failures: Analysing Stakeholder Influence in Complex Case Histories* John Donaldson, Software Forensic Centre, Middlesex University, UK. A *Systems Approach for Resolving Complex Issues in Design Processes* Emad Marashi and John P. Davis, Dept. of Civil Engineering, University of Bristol,
13.00-14.30	*Lunch*
14:30-15:30 Chair: Bob Wears, (tbc), University of Florida, USA.	**Paper Session 7: Errors and Accidents** *Communication Tool Between Designers and Accidentologists for In-car Safetys* Walid Ben Ahmed, Mounib Mekhilef, Michel Bigand*, Yves Page** Ecole Centrale de Paris, * Ecole Centrale de Lille, ** PSA-Renault, *Barriers, Complexity and Human Reliability* B. Schuup and P. Wright, Dept of Computing Science, University of York.
15.30-15:45	*Tea*
15:45-17:15 Chair: Peter Wright, Dept of Computing Science, Univ. of York	**Paper Session 8: Complexity, Abstraction and Viewpoints** *Adapting Interface Representations for Mobile Support in Safety-Critical Contexts* Fabio Paterno', Carmen Santoro and David Touzet, ISTI-CNR, Italy. *Viewpoints and Views in Engineering Change Management* R.Keller, C.M. Eckert & John Clarkson, EDC, Univ of Cambridge. *Applyng Task Analysis to Facilitate the Design of Context Aware Technologies* Yun-Maw Cheng and Chris Johnson*, Acedmia Sinica, Taiwan, * University of Glasgow
17:15-17:30	*Close and hand-over*.

SATURDAY, 12TH MARCH

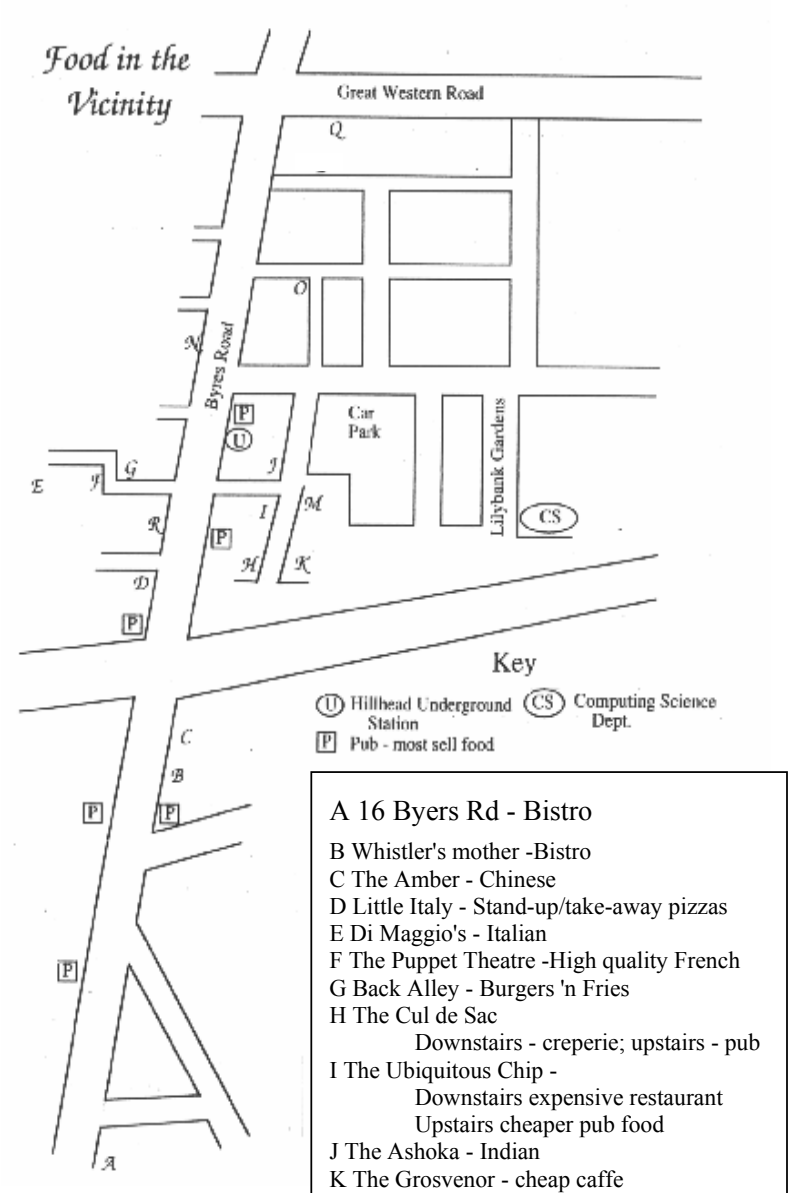
This will provide the opportunity for informal discussions about the issues raised during the workshop. The day will be spent on the Isle of Arran, off the west Coast of Scotland. The intention is to meet outside the Department of Computer Science at 07:30. We will be taking the train because this connects directly with the CalMac (http://www.calmac.co.uk) ferry onto the Island. Anyone who misses the first rendez-vous can meet us underneath the large clock at Central Station for 08:00 (Buchanan Street is the nearest Underground station). Trains depart from Glasgow Central station at 08:33, arrives at Ardrossan harbour at 09:25. The ferry leaves for Arran at 09:45. Ferry arrives at Brodick on Arran at 10:40. The ferry departs Brodick at 16:40, arrives Ardrossan 17:35. The train arrives at Glasgow Central 18:52. There is an additional service departing Brodick at 19:20, arriving at Ardrossan to connect with the 20:30 that arrives into Glasgow at 21:22.

If anyone misses this departure then they will have to spend the night on the Island (there are lots of hotels and bed & breakfast places). Arran Tourist Office can be contacted on 01770-302140 or 01292 678100 (http://www.ayrshire-arran.com/arran.htm) for hotel accommodation and other enquiries. The whiskey distillery is open for visits from 10.00-18.00 and can be contacted on 01292 678100.

Out	Monday to Saturday					Sunday			
Glasgow Central dep	**0833**	1115	1415	1650	1915	0840	1115	1405	1655
Ardrossan dep	**0945**	1230	1515	1800	2030	0945	1230	1515	1800
Brodick arr.	**1040**	1325	1610	1855	2125	1040	1325	1610	1855

Return	Monday to Saturday						Sunday			
Brodick dep	0820	1105	1350	**1640**	1920	2140	1105	1350	1640	1920
Ardrossan arr	0915	1200	1445	**1735**	2015	2235	1200	1445	1735	2015
Glasgow Central arr	1022	1322	1622	**1852**	2122	-	1328	1550	1850	2117

Restaurants in the Local Area



A 16 Byers Rd - Bistro
B Whistler's mother -Bistro
C The Amber - Chinese
D Little Italy - Stand-up/take-away pizzas
E Di Maggio's - Italian
F The Puppet Theatre -High quality French
G Back Alley - Burgers 'n Fries
H The Cul de Sac
Downstairs - creperie; upstairs - pub
I The Ubiquitous Chip -
Downstairs expensive restaurant
Upstairs cheaper pub food
J The Ashoka - Indian
K The Grosvenor - cheap caffe
M Jinty McGinty's - Irish pub & bar food
O The Metro - salads, hot dishes
Q The Parthenon - Greek food
R Burger King

What are Emergent Properties and How Do They Affect the Engineering of Complex Systems?

Christopher W. Johnson,

Department of Computing Science, University of Glasgow,
Glasgow, G12 9QQ. Scotland, UK.

johnson@dcs.gla.ac.uk, http://www.dcs.gla.ac.uk/~johnson

Abstract: 'Emergent properties' represent one of the most significant challenges for the engineering of complex systems. They can be thought of as unexpected behaviors that stem from interaction between the components of an application and their environment. In some contexts, emergent properties can be beneficial; users adapt products to support tasks that designers never intended. They can also be harmful if they undermine important safety requirements. There is, however, considerable disagreement about the nature of 'emergent properties'. Some include almost any unexpected properties exhibited by a complex system. Others refer to emergent properties when an application exhibits behaviors that cannot be identified through functional decomposition. In other words, the system is more than the sum of its component parts. This paper summarizes several alternate views of 'emergence'. The intention is to lend greater clarity and reduce confusion whenever this term is applied to the engineering of complex systems. This paper was motivated by observations of a recent workshop on Complexity in Design and Engineering held in Glasgow, Scotland during March 2005. It builds on the analysis presented by the other papers in a special edition of Elsevier's Reliability Engineering and Systems Safety journal. These other papers are indicated by references that are not accompanied by a numeric index.

Keywords: complexity, emergent properties, philosophy, engineering, layers of abstraction.

Introduction

Complex systems can be characterized in several different ways. At a superficial level, they are hard to design and understand. These conceptual problems often stem from the multiple interactions that occur between many different components. Marashi and Davis define complex systems to 'contain many components and layers of subsystems with multiple non-linear interconnections that are difficult to recognise, manage and predict'. Taleb-Dendiab, England, Randles, Miseldin and Murphy provide an example of this 'face-level' complexity in the design of their decision support system for post operative breast cancer care. The Neptune system combines a 'situation based calculus' with a 'grid architecture' to provide distributed support to a large and diverse user population. However, Solidova and Johnson provide examples of second order complexity where relations between sub-components change over time. They cite the difficulty of predicting the complex temporal flows of interaction in response to rapidly changing properties of both systems and environment. As Wears, Cook and Perry observe in their contribution on healthcare failures 'The incident emerged from the interaction of major and minor faults which were individually insufficient to have produced this incident. The design problem here is that validation of individual device design is an insufficient basis from which to conclude that use in context will attain the design performance levels'.

It seems clear that complexity poses growing challenges to systems engineering. For example, Gott (2005) has recently identified what he calls the 'complexity crisis'. He finds that, "…driven by markets that demand ever-increasing product value and functionality, manufacturers have embarked on continuous product improvement, delivering more features, more innovation and better looking products". He goes on to argue that "coping with the resulting design complexity while maintaining time to market and profitability is the latest challenge to hit the engineering industry". Bruseberg provides examples of this interplay between technical difficulty, project management issues and the economics of systems development in her study on the use of 'Commercial Off The Shelf' (COTS) software components for the UK Ministry of Defence (MoD). In her case, complexity stems from the need to satisfy system safety requirements using components that were not designed for safety-critical applications. Similarly, Felici's paper looks at the socio-technical challenges created by the need to maintain safety margins in European

Air Traffic Management as the number of flights are predicted to double by 2020 without significant increases in infrastructure investment.

Wulf (2000) has identified complexity as one of the most significant "macroethical" questions facing the US National Academy of Engineering "the key point is that we are increasingly building engineered systems that, because of their inherent complexity, have the potential for behaviors that are impossible to predict in advance". A recurring theme in this paper will be the 'surprise' that engineers often express following adverse events. This is illustrated by Basnyat, Chozos and Palanque's analysis of a complex mining accident and by Felici work on the Űberlingen mid-air collision. Both incidents revealed traces of interaction between operators and their systems that arguably could not have been anticipated using current engineering techniques.

Engineering Complexity in a Historical Context

Complexity is not a novel phenomenon. Galileo's Dialogues Concerning Two New Sciences describes how a column of marble was stored on two supports at either end. The masons knew that these columns could break under their own weight and so they placed another support at the mid section of the beam. They were satisfied with their work until the column broke at exactly this mid-point. One of the interlocutors in the Dialogue's makes observations that have a great deal in common with modern descriptions of emergent behaviors in complex systems; the failure of the column was "a very remarkable and thoroughly unexpected accident, especially if caused by placing the support in the middle". One of the end supports had decayed over time while the middle support had remained hard so that half of the beam projected into the air without any support. Petrowski (1994) states that it is 'to this day a model of failure analysis' and goes on to point out that if the supports had been placed at the extreme eights of the column then the maximum bending stress would have been up to double that of the beam resting on its end points alone. Arthur D. Little (1915) one of the pioneers of chemical engineering argued that the basic unit operations in any process are relatively simple. However, "the complexity of chemical engineering results from the variety of conditions as to temperature, pressure, etc., under which the unit operations must be carried out in different processes, and from the limitations as to material of construction and design". Such quotations remind us that environmental and contextual views of engineering complexity have very early roots.

It is possible to distinguish between the elegant 'complexity' of early engineering and the 'messy complexity' of modern systems. As Marashi and Davis note 'An effective and efficient design could not usually be achieved without a proper understanding of the relationship between the whole and its parts as well as the emergent properties of the system. A wicked and messy problem like engineering design has many interlocking issues and consequences which may be unintended. The vast range of stakeholders involved in an engineering design project, e.g. public, client, construction team, designers, financiers, managers, governmental agencies and regulating bodies; and their changing requirements from the system, escalates the complexity of the situations. Furthermore, the objectives change in response to the actions taken and each attempt for a solution changes the problem situation'.

The introduction of information technology and integrated manufacturing techniques has broken down boundaries between subsystems to the point where the behavior of many systems cannot adequately be modelled in terms of individual components. Changes in one area of a system quickly propagate themselves to many other areas in ways that seem impossible to control in a reliable way. The North American electricity black-out of August 14th 2003 provides a good example of this 'messy' complexity. Few could have predicted the way in which deregulation of an industry might combine with the diseased branches of several trees to bring down a vast network of power distribution across two nations. However, not all modern systems possess this 'messy complexity'. Felici argues that 'nuclear or chemical plants are well-confined entities with limited predictable interactions with their surroundings. In nuclear and chemical plants design stresses the separation of safety related components from other systems. This ensures the independence of failures…In contrast; ATM systems operate in open and dynamic environments. Hence, it is difficult to identify the full picture of system interactions in ATM contexts'.

What is Emergence?

Emergent properties are often used to distinguish complex systems from applications that are merely complicated (Johnson, 2003). They can be thought of as unexpected behaviors that stem from interaction between the components of an application and the environment. Emergent properties can be beneficial, for example, if users adapt products to support tasks that designers never intended. They can also be harmful if they undermine important safety requirements. However, there is considerable disagreement about the nature of 'emergent properties'. Some include almost any unexpected properties exhibited by a complex system. Others refer to emergent properties when an application exhibits behaviors that cannot be identified through functional decomposition. In other words, the system is more than the sum of its component parts.

The British Emergentists and Layered Views of Complexity

The recent preoccupation with emergent properties has many strands. One area of research has renewed interest in the parallels between complex technologies and biological systems. For example, Holland (1998) and Gershenfeld (1999) put a new spin on the work of philosophers such as J.S. Mill (1884): "All organised bodies are composed of parts, similar to those composing inorganic nature, and which have even themselves existed in an inorganic state; but the phenomena of life, which result from the juxtaposition of those parts in a certain manner, bear no analogy to any of the effects which would be produced by the action of the component substances considered as mere physical agents. To whatever degree we might imagine our knowledge of the properties of the several ingredients of a living body to be extended and perfected, it is certain that no mere summing up of the separate actions of those elements will ever amount to the action of the living body itself". Mill based his arguments on distinctions between heteropathic and hompathic effects. Homopathic effects arise where causes acting together are identical to the sum of the effects of those causes acting in isolation. For example, forces acting on an object can have the same effect when applied in combination or separately. In contrast, heteropathic effects describe emergent properties seen in complex biological and chemical systems. These conjoint actions cannot be characterised by the sum of any individual causes. For example, the addition of sodium hydroxide to hydrochloric acid produces sodium chloride and water. It is unclear how such a reaction could be characterised as the sum of individual components.

Mill's ideas indirectly led to the development of the 'British Emergentists'. Although their work was not directly intended to guide the engineering of complex systems, many of their ideas have implicitly helped to shape current debates in this area. The emergentists proposed a layered view of complexity in which the world is divided into different strata. At the bottom are fundamental physical laws. On this foundation we can observe chemical, biological, psychological and social interactions at ever increasing levels of organisational complexity. Research in physics, therefore, investigates fundamental properties and laws that are broadly applicable. The remaining 'special sciences' focus on properties that emerge from complex systems. These emergent properties can be influenced by behaviors at lower levels in this layered approach. In engineering, therefore, we can identify behaviors that cannot be understood in terms of the individual observations of underlying physical phenomena. They can only be considered in terms of their collective actions at the higher systems level.

An immediate problem with this layered approach to complexity is that many properties stem from interaction between systems and their environment. Paternò and Santoro make this clear when they analyse the impact of context of use on interaction with safety-critical systems. While standard emergentist approaches look at general relationships between different layers in complex systems. Engineers must, typically, focus on specific relationships between a system and its environment. The underlying ideas in the emergentist approach do not readily suggest a design method or development technique.

Alexander and the Challenge to Functional Decomposition in Risk Assessment

There are several different theories about how relationships are formed between the layers of a complex system. For Mill and later philosophers such as Broad, higher-level emergent properties in complex systems stem from, and are in addition to, lower level causal interactions. Hence we can talk about both the ways in which crowds behave at an organisational level and at the level of individual actions that influence wider patterns of group behavior. The distributed cognition observable in teams of system operators is often quite different from the sum of the individual cognitive resources displayed by each

individual user. Although the sum may be more or less than the individual parts of a complex system, in Mills view, it is still possible to reason about properties of the higher level systems in terms of the properties possessed by their component parts.

The second explanation of interaction between levels in complex systems was proposed by emergentists, such as Alexander (1920). They argue that the appearance of novel qualities and associated, high-level causal patterns cannot be directly expressed in terms of the more fundamental entities and principles. In this view, it makes little sense to talk of human cognition in terms of individual neurons. Consciousness is intrinsically a systems level property quite distinct from the underlying physiology of lower level components. The term emergence is often used to reflect the limitations on our understanding of complex systems. This strand of thought is strongly associated with engineering studies of 'messy' complexity, mentioned in previous paragraphs. Although the work of Alexander and his colleagues is not primarily intended to address the engineering of complex systems, the implications are clear. The idea that there are properties of systems that cannot intrinsically be understood in terms of lower level concepts seems entirely at odds with many contemporary approaches to engineering. For example, this would suggest that there are many risks that cannot be identified by following the functional decomposition that is implicit within techniques such as FMECA.

Some authors, including Pepper (1926), have attacked the concept of emergence in complex systems. For example, it can be argued that unexpected or novel macroscopic patterns of behavior do not reveal special forms of 'emergent' properties. Instead they simply illustrate problems with our understanding of a complex system. The limitations in our knowledge could be addressed if we augmented our theoretical understanding to include the conditions in which novel phenomena occur. It would then be sufficient to construct more complex laws that specify behavior when the new variables are not satisfied and the 'novel' behavior when the variables are satisfied (O'Connor and Wong, 2002). This approach creates a number of problems. For example, the papers of this special edition describe a great range of emergent phenomena in several different application domains ranging from healthcare to aviation. Pepper's would characterize their behavior using of dozens of disjoint laws, each of which would describe system properties in a very small set of circumstances. By extension, we might also argue for the presence of emergent properties whenever there is discontinuity in microscopic behaviors unless we can develop an elegant theory that relies more narrowly on the basic properties of the system.

Predictive Approaches to Emergence in Accident Investigation

The contemporary philosophy of emergence has been heavily influenced by predictive approaches; emergent properties are system level features that could not have been anticipated. For example, Wears, Cook and Perry describe healthcare vulnerabilities that cannot easily be anticipated by designers 'in particular, some forms of failure emerge from the interactions of independently designed and implemented components'. They go on to present a case study 'of such an emergent, unforeseen failure and use it to illustrate some of the problems facing designers of applications in health care'.

Predictive approaches are orthogonal to Alexander's ideas. For instance, there are many systems level properties that are not directly related to system subcomponents but which might be predicted. Alexander might call these properties emergent but this attribution would not fall within the definitions adopted by predictive approaches. For example, 'risky shift' occurs when greater risks are accepted by groups than would have been taken by individual team members. In other words, the level of acceptable risk shifts towards the more risk preferring members of the group. Such behaviors are emergent in Alexander's terms because they are closely entwined with the behavior of groups. However, in a predictive sense they need not be emergent because they can be anticipated. Designers can, therefore, take steps to guard against these behaviors that might otherwise compromise the engineering of complex systems. For example, risky shift can often be detected by external reviews of team-based risk assessments.

The predictive approach to modern theories of emergence raises questions about the perspective of the person making the predictions. Designers and engineers never have 'perfect knowledge' about the systems and environments that they work with. Information about a complex system may be distributed in such a way that some properties may be emergent for a group that lacks key information whereas the same features of a complex system might easily have been anticipated by co-workers with additional data. It is,

therefore, often assumed that emergent properties are those that cannot be predicted by individuals who possess a thorough knowledge of the features of, and laws governing, the parts of a complex system and its environment (O'Connor and Wong, 2002). This relativism has important implications in the aftermath of major accidents. In retrospect, it is easy to argue that any behaviors that were not anticipated during the design of a complex system were emergent simply because they had not been predicted. This interpretation ignores the important caveat that emergent properties must be assessed with respect to a thorough knowledge of the system and environment in question. Under the predictive view, to argue that an accident was caused by an emergent property is to accept that the behavior could not have been anticipated by an individual with "a thorough knowledge of the features of, and laws governing, the parts of a complex system and its environment". All too often, engineers have issued warnings about possible accidents that have been ignored until after an adverse incident has occurred (Johnson, 2003). Such failures cannot be described as emergent properties under this predictive interpretation.

Popper and Eccles (1977) have extended studies into emergence and unpredictability by investigating the non-determinism that often characterizes complex systems. Designers often fail to determine the behavior of application processes. For example, environmental conditions can introduce the non-determinism that prevents accurate predictions about complex systems. In a layered view, non-determinism can also stem from interactions with the layers both above and below a particular level. Apparently random behaviors can arise when we have failed to understand underlying systems. Early 19th century bridge builders such as John Scott Russell, struggled to predict the performance of suspension bridges because they did not sufficiently understand the way in which the positioning of crossbars could affect the modes of vibration that affected particular designs. Without any sufficient theory, the best that could be done was to use experimental methods as a means of mapping out the apparent non-determinism that was observed when some bridges failed whilst others succeeded (Petrowski, 1994). This analysis seems to contradict Anderson's view in which there are emergent properties that cannot be understood in terms of the underlying layers in a complex system. In contrast, if we view emergence as strongly related to non-determinism then it might be possible to 'control' or at least anticipate emergent properties by understanding the source of any non-determinism, for example, by studying the underlying properties of lower level components within a system. The non-deterministic approach to emergence, therefore, need not reject functional decomposition as a primary tool in the engineering of complex systems.

Teleological Approaches to Emergence

Chalmers (2002) identifies a spectrum of approaches to emergence. At one extreme, 'emergence' is used to describe 'semi-magical' properties. This captures an extreme form of surprise where higher-level properties of a system cannot be deduced from lower level attributes no matter how sophisticated the analysis or the analyst. At the other end of the spectrum is a more prosaic view in which emergence means little more than properties that are possessed by a 'whole' and not by its parts. In this view, almost every non-trivial object possesses emergent properties, including filing cabinets and chairs.

Most applications of the term lie between these extremes. When we talk about emergent properties in biological systems or connectionist networks we are usually referring to behaviors that can, in principle, be deduced but only with great difficulty. Bedau (1997) calls this 'weak emergence'. Such properties are identified by the degree of difficulty that an observer has in deducing them from lower level phenomena. However, emergence often carries with it the notion that the underlying phenomena are relatively simple. For example, the behavior of a large-scale computer program can be almost impossible to deduce in terms of the underlying binary signals. However, few people would say that the behavior is emergent. In contrast, many authors describe the computational architecture of connectionist networks as displaying emergent properties. Chalmers views emergence as a largely positive phenomena where these simple combinations of simple components buy you 'something for nothing'. In other words, emergent behaviors in biological systems support behaviors in addition to those provided by individual components. For instance, the visual system supports perception that defies explanation in terms of components such as the retina, cornea etc. Similarly, the genetic mechanisms of evolution are very simple but the results are complex.

Chalmers' analysis approaches a teleological definition of emergence. These phenomena are associated with systems that possess interesting properties that were not included in the goals of the designer. This teleology is significant because for many working on the biological aspects of emergence, the notion of a 'designer' implies some guiding hand that stands at odds with Darwinian views. However, Chalmers

(2002) argues that the psychological and relative approach to emergence also allows a non-telelogical approach; ‘in evolution, for instance, there is no "designer", but it is easy to treat evolutionary processes as processes of design’. It is more straightforward to apply Chalmers’ work in the field of engineering where design objectives can be inferred without reference to divine intervention.

Bedau and Weak and Strong Emergence

As mentioned, Bedau (1977) distinguishes between weak and strong emergence. Weak emergence is a macroscopic state which could be derived from knowledge of the system's micro-dynamics and external conditions but only by simulating or modeling all the interactions of the microstates starting from a set of initial conditions. Bedau’s work contributes to recent research in the area of chaos ‘theory’. His view of weak emergence characterizes situations in which the longer term outcome of non-linear processes is sensitive to very small differences in initial conditions or environmental factors. However, in weak emergence it is possible for engineers to derive these higher level behaviors from lower levels even if this analysis requires considerable modeling resources.

In contrast, Bedau’s work on strong emergence borrows much from the mind-body problems of cognition, mentioned in previous sections. Higher-level behaviors are largely autonomous from underlying layers, just as higher levels of cognition cannot easily be described in terms of neurological processes. These distinctions between weak and strong emergence can also be characterized in terms of causal relationships between the different levels of a complex system. For example, weak emergence can be analyzed using reductionist techniques where complex behaviors at a systems level are caused by properties of underlying components. In contrast, strong emergence relates to a form of ‘downwards causation’ where behaviors at lower levels in a system are constrained by higher level characteristics. One way of thinking about this is in terms of social interaction. The behavior of a crowd can be simulated in terms of the behavior of individual members. This represents a ‘bottom-up’ form of weak emergence. In contrast, crowds also act in conformity with rules that govern their behavior as a whole. For instance, if a crowd enters a narrow alley then it alters its movements. Groups entering the constriction will slow their pace in order to avoid hitting or getting to close to others in front. Locally, the crowd self-organises even though for any individual the movements and buffeting may appear random. As Lemke (2000) observes of this form of strong emergence; “Order forms because there are only relatively few solutions to the problem of correlated motions, and when contrasted with an ideal of randomness in which all possible states of motion are equally likely, those few solutions stand out as orderly”. It is for this reason that many computer-based evacuation simulators enable their users to specify individual behaviors. These tools also provide facilities for users to place constraints onm crowd behaviors, to simulate the flocking that occurs in the immediate aftermath of some adverse events (Johnson, 2005).

These notions of strong and weak emergence can be contrasted with the predictive approaches mentioned earlier. Recall that emergence can be connected to non-determinism and that the term ‘emergent property’ is often used to describe a feature of a complex system that was not anticipated by systems engineers. This creates problems because there are classes of properties that relate to systems level behaviors, which seem to be emergent, but that are also predictable. Further problems arise because emergent properties rely on the subjective experience of people making the predictions. The idea of strong emergence avoids some of the conceptual problems that arise when these emergent behaviors are narrowly tied to predictions about complex behaviors. Strong emergent properties cannot be reduced to the physical laws of causal composition. However, they can still be described in terms of other laws or patterns of behavior. We can still talk about patterns in cognitive behavior even though we cannot explain in detail how those behaviors relate to underlying electrochemical changes in the brain. Clark (2001) argues that emergent phenomena are best understood by observing a ‘pattern resulting from the interactions’ among multiple elements in a system including aspects of the environment.

The Engineering Implications of Emergence

The previous paragraphs have provided an initial overview of emergence. The intention has been to provide a more structured, theoretical basis to the engineering of complex systems. In such an abstract and often theoretical discussion it is easy to lose sight of the importance of these ideas for complex systems engineering. Wears, Cook and Perry make the following statement; ‘emergent vulnerabilities, such as arise from the interaction among disparate, independently designed components, seem almost impossible to

foresee in anything other than the most general terms. Health care seems especially vulnerable to these sorts of threats for several reasons: 1) The relative youth of complex computer application in the field; 2) The general unfamiliarity of health professionals and managers with methods for reducing vulnerabilities; 3) The fragmented nature of health care "organizations"; 4) The potential subversion of risk information into internal, conflicting agendas; and 5) The lack of formal or regulatory frameworks promoting the assessment of many types of new technologies. These factors are as much social-organizational as they are technological'.

It seems clear, therefore, that emergent properties have considerable significance for the design and engineering of many applications. It is less clear that the philosophical ideas on emergence can make a significant contribution to engineering and design. The ideas are interesting but how can they help engineers? Buchli and Costa Santini (2005) have observed that the process of finding unifying principles either at the microscopic or macroscopic levels of complex systems, is hindered both by the divisions between specialised disciplines and by the problems of language where different concepts share overloaded names. Haken (1999) continues that "despite a lot of knowledge about complex systems the application of this knowledge to the engineering domain remains difficult. Efforts are scattered over many scientific and engineering disciplines". Attempts to establish complexity engineering as a discipline are hindered by basic misunderstandings over common terms such as 'emergence'. It is unlikely that the 'concensus making' advocated by Marashi and Davis will be successful while more basic disagreements complicate the use of common terms.

The confusion created by the (ab)use of common terms can be illustrated by two recent papers on engineering with complexity1. The first argued that "emergence is often associated with a 'surprise-factor': local interactions result in something unexpected at the global level. Engineering emergence is about removing this surprise". Such comments illustrate a pragmatism based on the predictive approach to emergence and non-determinism described in previous paragraphs. In contrast, a companion paper went on to demonstrate "…that interacting cell networks are prime candidates to study principles of self-organized pattern formation. In addition, they offer a multitude of possibilities for microscopic interactions that might also be relevant for dynamic communication networks. Examples of interacting cell systems are life cycles of bacteria or social amoebae, embryonic tissue formation, wound healing or tumour growth and metastasis. Then, we show that mathematical modelling of dynamic cell networks (biomathematics) has developed techniques which allow us to analyze how specific microscopic interactions imply the emergence of a particular macroscopic behavior. These techniques might also be applied in the context of dynamic communication networks". The aim of this work is to transfer observations about the macro behavior of biological systems to the engineering of telecommunications networks using a language of 'self-organisation'. This has much in common with the idea of strong emergence, although the author does not use this term and shows no evidence of having read Bedau.

These two papers reflect very different implicit views of emergence. The resulting tensions are most apparent when Zambonelli (2005) argues "It is getting more and more recognized that the exploitation of self-organization and emergent behaviors can be a feasible way to bear the complexities and dynamics of modern systems. However, attempting at defining a practice of engineering such emergent and self-organizing systems in a reliable and repeatable way appears a contradiction in terms". As we have seen, this contradiction arises because engineers freely move from predictive definitions in which emergence is equated to a surprise and definitions of strong emergence where higher-level patterns can be used as design templates. The main aim of this paper is to help future engineers avoid these contradictions. Greater care must be taken when using terms such as 'emergence'. Without this there is little chance of developing the discipline of complexity engineering.

[1] Engineering with Complexity and Emergence (ECE'05), Paris, Satellite workshop of the European Conference on Complex Systems, see http://complexsystems.lri.fr/

Conclusions

Complex systems research has been hindered by a lack of precision when people refer to 'emergent properties'. Contemporary views of emergence in philosophy include Chalmers' spectrum ranging from a mystical property to the whole-part relationships in mundane objects including filing cabinets. They also include Bedau's distinction between 'weak' emergence, based on simulation and modeling, and 'strong' emergence relying on downwards causation. As we have seen, problems arise because engineers combine many different aspects of these ideas when referring to emergence in complex systems. They refer to the surprise implicit in predictive approaches while talking about the design of emergent properties. In contrast, we have attempted to ground recent research into complex systems by surveying different approaches to emergence. The intention has been to help engineers avoid some of the paradoxes that arise when inconsistent definitions are used.

Further work remains to be done. For example, engineers continue to extend the concept of emergence in many directions that are not adequately captured by philosophical discourses on complexity. For instance, Eckert, Keller, Earl and Clarkson refer to emergent changes, 'which arise from problems with the current state of a design proposal in terms of mismatches with requirements and specification…these can be caused by mistakes, supplier constraints and factors internal to the process such as resources, schedules and project priorities across the company'. Although these changes clearly emerge during manufacturing 'from a mistake or a late modification from the supplier, designers often resent it as avoidable'. Further work is required to determine whether such properties are a particular instance of Bedau's weak emergence, only predictable through advanced simulation techniques, or whether they pose a further challenge to the philosophy of emergence as it relates to engineering and design.

Acknowledgements

The workshop upon which this work and the special edition is based was supported by the European Commission's ADVISES Research Training Network (HPRN-CT-002-00288) and by a grant from the European Office of Aerospace Research and Development, Airforce Office of Scientific Research, United States Air Force Laboratory.

References

S. Alexander, Space, Time, and Deity. Macmillan, London, 1920.

U. Beck, Risk Society, Sage, London 1992.

M. Bedau, Weak Emergence, Philosophical Perspectives, 11: Mind, Causation, and World. Blackwell, pp. 375-399, 1997.

J. Buchli and C.C. Santini. Complexity engineering: Harnessing emergent phenomena as opportunities for engineering. In Reports of the Santa Fe Institute's Complex Systems Summer School 2005. Santa Fe Institute, 2005.

D.J. Chalmers, Varieties of Emergence, Technical report/preprint, Department of Philosophy, University of Arizona, USA, 2002.

A. Clark, Mindware, MIT Press, Cambridge, USA, 2001.

B. Gott, The challenge of complexity in product design and engineering, Cambashi Limited, Cambridge UK, 2005.

N. Gershenfeld, When Things Start to Think, Henry Holt, New York, 1999.

H. Haken. Information and Self-Organization A Macroscopic Approach to Complex Systems. Springer, Berlin, 1999.

J.H. Holland, Emergence from chaos to order, Oxford University Press, Oxford, 1998.

C.W. Johnson, A Handbook of Accident and Incident Reporting, Glasgow University Press, Glasgow, 2003.

C.W. Johnson, Applying the Lessons of the Attack on the World Trade Center, 11th Septermber 2001, to the Design and Use of Interactive, Evacuation Simulations, In Proceedings of ACM CHI 2005, 651-660, ACM Press, New York, 2005.

J.L. Lemke, Material Sign Processes and Emergent Ecosocial Organisation. In P.B. Andersen, C. Emmeche, N.O. Finnemann and P.V. Christiansen (eds) Downward Causation, Aarhus University Press, 2000.

A.D. Little, cited from 1915 in W. Pafco (ed.) Struggle for Survival, History of Chemical Engineering: American Institute of Chemical Engineers, 2005.

J.S. Mill, A System of Logic, Bk.III, Ch.6, §1, Longman, 1884.

T. O'Connor and H.Y. Wong, Emergent Properties, The Stanford Encyclopaedia of Philosophy (Summer 2005 Edition), Edward N. Zalta (ed.).

S. Pepper, Emergence, Journal of Philosophy, 23: 241-245, 1926.

K.R. Popper and J.C. Eccles, The Self and Its Brain. Springer International, New York, 1977.

C. Perrow, Normal Accidents, Princeton University Press, 1999.

H. Petrowski, Design Paradigms, Cambridge University Press, 1994.

W.A. Wulf, Great Achievements and Grand Challenges, The Bridge, US National Academy of Engineering, (30)3&4, Fall/Winter 2000.

F. Zambonelli, Tackling the "engineering emergence" oxymoron: some preliminary thoughts, Engineering with Complexity and Emergence (ECE'05), Paris, 2005.

Appendix A: Tom Maibaum's Comments on the First Draft

One of the main motivations behind this paper was to provide an overview of different philosophical views on emergence, as they relate to the engineering of complex systems. It is difficult to envisage any survey being 'complete' given the controversial and changing nature of this subject. I have, therefore, extended the initial version of this paper by including comments and criticisms that have been received sine it was first published. The following email from Tom Maibaum makes an important point with respect to non-determinism and under specification that I failed to consider. In retrospect, this omission is all the more regrettable given that it comes from my own discipline!

From Tom Maibaum [tom@maibaum.org], 16th February 2006.

Dear Chris

I have been reading your recent paper on 'What are Emergent Properties ....'. I have some thoughts about it that your paper helped me to place in context and I wanted to share some of them with you.

I have been working on component based design of systems for almost 20 years, using category theory as a setting for describing how components are bound together to make larger components/systems. The binding is basically describing how interaction between components is defined. (It is an example of a coordination mechanism.) We use temporal logic to specify components. Now, if you take a classical example like the dining philosophers, specified in terms of philosopher components and fork (resource)

components and 'connectors' to make them work together correctly, then if you ascribe to philosophers the property that if they hungry, they will eventually pick up their forks and eat, when you look at the system of dining philosophers (and forks), a philosopher has an emergent property! It is the property that the philosopher eventually puts his forks down! This is because models of the whole system require that all philosophers eventually eat and this requires neighbouring philosophers to eventually release the resources required to do this.

Now, this property was not precluded for philosophers, but not all models/executions of philosophers have this property. So one way of looking at this is that the 'social' interaction of the philosophers (and forks) excludes some potential models of components as not supporting the 'social' behaviour of the components. This relates to what you referred to as 'nondeterminism' in your paper. It is NOT nondeterminism, which is to me an aspect/attribute of a model of computation, like in a Turing machine. What you are referring to is what I call UNDERSPECIFICATION. That is, the model/implementation you are interested in is not fully determined by the specification. (This is what SE is about: reducing underspecification by development steps. This is the classical reference to 'introducing implementation detail' in SE books.) Now, many people get nondeterminism and underspecification mixed up. It is a terrible mistake to make.

Let me give you another, simple, illustration. Suppose you specify sets of somethings, say using first order logic. You use a FUNCTION symbol called 'choose' that, when applied to a non-empty set, gives you back some value in the set. Almost everyone calls this a nondeterministic function. But it is not, because no such animal exists in first order languages. It is a function proper that is underdetermined. So, an implementation might fix on the minimum value in the set as the 'right' implementation. So, if in your implementation you apply choose to the same set at different points during execution, YOU WILL ALWAYS GET THE SAME VALUE! Of course, the implementer might change the implementation on you overnight, and tomorrow you will get a different value applied to the same set, say the maximum. But he is within his rights to do this as it satisfies the specification. The implementation is in no way nondeterministic. A nondeterministic mechanism (of computation) is used in languages like Prolog, where choose would be defined as a relation and Prolog would generate for you ALL THE VALUES IN THE SET in some sequential and apparently nondeterministic order.

Nondeterminism as an internal computational mechanism is to do with randomness and is an external mechanism of choice for users (that may not be truly random).

Why have I gone into the distinction between nondeterminism and underspecification in such detail? Well, it is because I believe that emergent behaviour is intimately related to the latitude for deciding properties for components provided by underspecification. Underspecification is an inherent property of engineered artefacts (as we cannot fully describe everything and all aspects of our artefacts). It is also an inherent aspect of natural phenomena to the extent that we cannot fully describe entities in nature, so we 'underspecify' them.

Now back to the dining philosophers example. Clearly the property of a philosopher always eventually putting its forks down is valid only in some of the models/implementations. The problem of engineering of complex systems is that, at the level we are now discussing it, the 'choosing' of the subclass of models determined by the context in which a component is put IS ITSELF UNDERSPECIFIED. There are a multitude of possible choices for such emergent properties we could observe when we put systems together. The problem for the engineer is to find standard ways of putting components together so as to reduce the chaos of choosing these emergent properties. The whole area of feature interaction in telephony was essentially an example of the chaos of emerging phenomena emerging! :-) Here we had these various components/features, that had emergent properties that resulted exactly because of underspecified interactions with the rest of the system. These people then spent a decade sorting out what this meant for the discipline of component based telephony. Of course, systems of components also have properties that were not exhibited by individual components. The obvious example of this is deadlock. This is inherently a global property and cannot be reduced to a component property. (It corresponds to the idea that the context of the component eliminates ALL models of the component.) So there are properties that cannot be studied only at the component level and could be characterised as emergent properties of conglomerates not predictable by studying only component parts. (The emergent properties discussed in the examples above

were properties of individuals/components.) However, these global properties still depend crucially on underspecification (and clearly not on nondeterminism).

As a conclusion, one might say that the problem of emergent properties in engineered artefacts is centred on the nature of underspecification of components, environment, and the nature of interconnection and coordination and how these forms of underspecification are handled in engineering. (Of course, INCORRECT specification is also a problem!), Regards, Tom

Abstracting Complexity for Design Planning

David Wynn, Claudia Eckert and P John Clarkson,

Engineering Design Centre, Engineering Department, Trumpington St, Cambridge, UK.
http://www-edc.eng.cam.ac.uk/people/dcw24

Abstract: This short paper is based on research carried out during an eight month case study at a large UK aerospace company. The focus of the study was to develop a more effective technique for planning the complex design processes found in the company, in order to support the development of more detailed and more accurate schedules. The paper outlines some key findings of this research, concluding with an important challenge for future work: once an appropriate level of abstraction has been identified for modelling complex design plans, how can the distributed, subjective planning knowledge be synthesized to form a coherent perspective?

Keywords: complexity, planning.

Introduction

The design of complex products such as jet engines or helicopters requires the co-ordination of hundreds or even thousands of individuals performing tasks which span disciplines, companies, and locations. These tasks form an evolving network of interrelated activities; a network which companies strive to understand in their efforts to meet the increasingly tight time, budget and quality constraints placed on their design processes. Planning methods currently available to industry are ill-equipped to cope with the complexity and unpredictability of engineering design; as a result, design managers are forced to make decisions based on a limited overview of the development process. Additionally, although increasing pressure is placed on design managers to assess process risk in addition to product risk, this can be a difficult task without a coherent understanding of the plan of work.

This paper describes design planning as a complex process and outlines the author's approach to supporting industrial practice through plan modelling. Iteration is briefly described as the key driver of uncertainty in these models.

Complexity in planning

A plan may be viewed as a framework for reaching process goals, usually displaying both descriptive and prescriptive aspects: descriptive, due its use in predicting such quantities as programme cost, delivery dates or resourcing requirements; and prescriptive, due to its use in setting and communicating deadlines and deliverables. The ideal plan, as sometimes envisaged by design managers, would take the form of a simple model which determined the company's collective behaviour with regards to the context of the design process. As the process unfolded, the plan would be continuously modified to remain effective and up-to-date.

A more realistic picture of planning in industry is provided by Eckert *et al* [1], who carried out empirical studies in a number of design companies. They describe how many planning documents were used in parallel, and how these representations took on a variety of forms - including individuals' activity checklists, the ubiquitous Gantt charts, and even bills of materials. The information content of these documents was found to exhibit high overlap and a low degree of coherence. Eckert concluded that global consistency in planning is achieved through an ongoing process in which many individuals reason about and maintain overlapping sets of representations, and that the corresponding lack of overview led to avoidable mistakes and inefficiencies. In reality, then, the information captured in a document such as a Gantt chart is a model or abstraction of the planning knowledge which is in turn distributed amongst the individuals who use and maintain it. Such knowledge is often subjective, partly tacit and thus difficult to elicit. In summary, the planning process exhibits substantial complexity in its own right; complexity which builds on that of the design process it aims to describe and control. To be useful in industry, supporting techniques should provide an appropriate abstraction of the design process in order to provide the overview

required for effective planning, while remaining simple and flexible enough for application in a broad range of situations.

Improving planning practice

A number of approaches have been proposed to improve design planning, most of which have had limited industrial impact. Several of these approaches use simulation techniques to derive plans from models which capture possible routes through the design process; such plans may then be optimized with respect to variables captured within the underlying process model. These results can be difficult to verify as they depend, among other factors, on the simulation achieving an accurate reflection of the possible population of processes.

An alternative approach, taken by this work, is to model plans directly - in other words, to provide a representation which allows engineers to describe their existing plans in a more coherent and accessible form than is currently possible. A variation on the 'Signposting' framework [2] has been developed, using a simple graphical format to capture plans which are valid only within a certain, well-defined range of scenarios. Plans are described in terms of tasks and their input/output data, together with resource requirements and uncertainty information. The method makes use of a novel approach to modelling iteration and uses simulation to resolve the resulting networks into the familiar, linear Gantt chart format. In the example case study, this planning methodology allowed schedules of previously unattainable detail to be developed through an iterative process of critique and refinement.

The framework recognizes the conflict between utility and robustness of plans in cases such as engineering design, where processes can be seen to exhibit a high degree of unpredictability. To illustrate this concept, it is useful to consider two types of framework which may be used for modelling plans. Firstly, some models attempt to capture emergent properties through understanding the behaviour of and interactions between individual tasks; an approach taken by O'Donovan [3]. A key benefit of this 'bottom up' approach is the ability to construct models from disjoint fragments of process knowledge; however, the main challenges lie in effectively characterizing task behaviour and in verifying the emergent properties revealed through simulation. Secondly, widely used network representations such as Gantt charts or Design Structure Matrices may be used to model plans or processes as static networks of tasks. These 'top down' models have proven pragmatically valuable in representing the structural overview required for planning; however, they are less robust to the types of uncertainty found in the design process. Models developed using such representations can require frequent modification to remain useful in the dynamic context of the design process.

The new method described in this paper allows hierarchical plans to be constructed which contain both 'top down' and 'bottom up' descriptions. At any level in the hierarchy, the user may choose the description which is most suitable to that part of the process. This choice must be informed by an appreciation of the situated nature of uncertainty in the design process.

Causes and effects of design iteration

Iteration is commonly seen as a major driver of uncertainty in design planning. Unfortunately, despite the concept of iteration forming a core theme across much design literature, the available insights often seem unsuitable for application to planning.

Both in literature and in industry it was found that iteration may stem from a number of causes (figure 1):

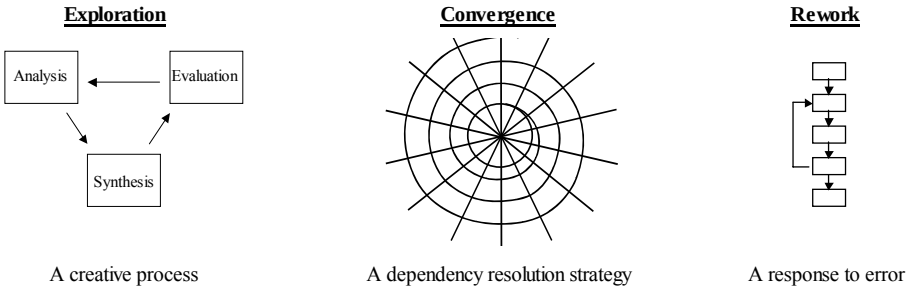


Figure 1: Iteration may stem from a number of causes

1. **Exploration**: Concurrent, iterative exploration of problem and solution spaces is fundamental to the creative process;
2. **Convergence**: Many design problems may be viewed as the selection of product parameters to satisfy complex dependency relationships. Where the problem is complex and/or requires integration of many personnel from disparate disciplines, an iterative process is often used to converge upon a satisfactory solution; and
3. **Rework**: Repetition of tasks may form a response to errors or unexpected events during the design process.

The effects of iteration on process behaviour were observed to vary with its cause and other domain-specific factors. Furthermore, the strategies used to plan design iteration were dependent upon its perceived behaviour and importance; observed strategies ranged from the explicit scheduling of a number of iterations through to the use of many coarsely grained, poorly integrated plans which allowed room for manoeuvre. For example, in a company where developing a complex product required considerable analysis effort, the need for cross-discipline interaction and concurrent engineering caused `convergence' iteration to form a strong, well-acknowledged influence on process behaviour.

A key complicating factor in modelling process plans was found to be the subjective nature of design tasks and iteration. While it was relatively easy to elicit small segments of process network, the individual segments often appeared to overlap and it was difficult to fit them together to form a coherent process. It eventually became clear that this difficulty was exacerbated by each engineer's holding a unique perspective on the process and on the location of possible iterations.

To illustrate, figure 2 depicts three alternative viewpoints of the concurrent design and manufacture of a component. From the programme manager's perspective, a convergent dialogue occurs between the design and manufacturing groups. However, the team developing the component perceives a sequence of many different tasks. A researcher conducting a protocol study might look closer still: from this perspective an iterative process emerges again, composed of many repetitions of a generic activity set. This subjectivity accounts for much difficulty in the planning, management and modelling of iterative design processes.

In the example case study, the largest plan contained less than 120 individual tasks; coherency was maintained by the researcher's understanding of the complete model. However, the complexity of the design process would render this approach infeasible for modelling a full project plan; future research will thus address the challenge of synthesizing a single picture of the complex design process by eliciting distributed, subjective knowledge.

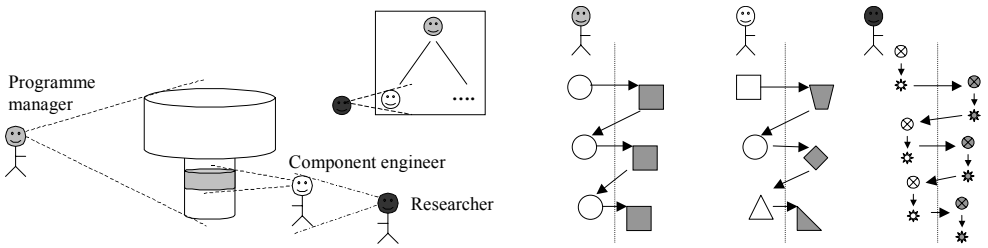


Figure 2: Design tasks and iteration are subjective concepts

References

[1] C M Eckert and P J Clarkson (2003) The Reality of Design Process Planning, International Conference on Engineering Design, ICED 03, Stockholm, 2003.

[2] Clarkson, P. J. and Hamilton, J. R. (2001) Signposting: a parameter-driven task-based model of the design process, Research in Engineering Design, 12(1): 18-38.

[3] O'Donovan, B., Eckert, C.M. and Clarkson, P.J. (2004) Simulating design processes to assist design process planning, Proceedings of ASME DETC04.

Design Change and Complexity

Chris Earl[1], Claudia Eckert[2], John Clarkson[2]

[1]Department of Design and Innovation, Open University, Milton Keynes, MK7 6AA,
[2]Engineering Design Centre, University of Cambridge, Trumpington Street, Cambridge, CB2 1PZ
http://design.open.ac.uk/earl.html
http://www-edc.eng.cam.ac.uk/people/cme26.html
http://www-edc.eng.cam.ac.uk/people/pjc10.html

Abstract: Design changes can be surprisingly complex. We examine the problems they cause and in what ways they are complex, mostly in the area of engineering design change. To assist this analysis we distinguish between (i) a static background of connectivities designs, processes, resources and requirements, (ii) descriptions of these elements and (iii) the dynamics of design tasks acting on descriptions. The background might consist of existing, similar designs, ways of describing them and established processes used to create them. We view design change, and design more generally, in terms of this model of background structure, associated descriptions and actions on descriptions. Sources of complexity in design change are examined and we indicate where these occur in different aspects of the model.

Keywords: change, structure, connectivities, description, action

Introduction

Many companies face the following situation: Customers request a new version of a design incorporating useful changes or marketing wants an updated product. Initially it might seem like a small change which can be implemented quickly. But during the process designers find it takes them longer then expected. The new requirements have affected parts which were expected to remain largely unchanged. Even experienced designers may not have predicted how changes would propagate across the design from one part to another. This has several implications: (i) Different parts are more expensive (ii) The original designers may not be available or be unable to explain their decisions or their design rationale. (iii) Designers of the new parts perceive that altering a complicated part involves high risk and try to avoid change, perhaps searching for work-arounds on simpler and perhaps more familiar parts. (iv) There may be several different records relating to the previous design but these may not be complete or it may not be clear which ones are relevant to the change. (v) The overall costs of change, in terms of time, resources and materials, can be large and unpredictable. (vi) The necessary time was not been planned into schedules and members of the project team need to move on to the next project. Customary practice may be abandoned and tasks compromised.

The modification or customisation of an existing design is not the only situation where change poses problems. A design process usually passes through several stages of signing-off parts and systems. Errors and mistakes in signed-off designs as well as new requirements from suppliers or clients can initiate change at any stage. If changes occur late in the process they can have serious effects, especially if the product has already proceeded to production. In this case the change takes place against the background of a nearly completed design rather than an existing one, but the problems are similar.

Responses to these problems include, managing the change processes (Fricke *et al.*, 2000, Terwiesch & Loch, 1999, Lindemann et al. 1998, Pikosz, & Malmqvist 1998) and devising effective representations (Cohen et al 2000). Recent research has comprehensively analysed types of engineering change (Eckert *et al.*, 2004), providing methods to represent linkages between parts in complex products (Jarrett et al 2004b) and to predict the risks associated with of propagation of changes through linkages among parts (Clarkson et al 2004). We will put these findings on managing change processes and analysing change propagation into a broader context by examining some general characteristics of change in design. First, change takes place against a rich background of knowledge and experience embodied in the current design which is the starting point for change. Second, the process of change is a fast moving, dynamic process, often highly creative in finding solutions. Third, change processes work on descriptions of different aspects of the

design such as function and geometry, the processes and resources available, and requirements of clients, customers, the company itself and its suppliers. These general characteristics help to reveal different sources of complexity in design change processes, particularly the complexity originating from a combination of order and uncertainty (Earl et al 2005). The ordered background of existing designs, processes and requirements is combined with an uncertain change process and unpredictable outcome.

Change

The two scenarios of change outlined above, namely modifying an existing design or recognising shortcomings in a nearly completed design, are part of a wider picture of design as an ongoing process of modification of previous designs. Cross (1989) identifies modification as a key aspect of design processes. Even innovative designs reuse parts, ideas and solution principles from existing designs. For example Dyson cyclone vacuum cleaners although innovative are in many ways similar to conventional ones in shape, brushes and basic function.

As with many areas of design research, investigations into change can be split into those that focus on the process of making an alteration (especially the management of the change) and those that examine the design itself. The majority of activity has concentrated on the former, for example the studies presented in Lindemann et al. (1998) or Pikosz & Malmqvist (1998). The close attention that has been paid to the management of change processes has in part been driven by the needs of companies to comply with Configuration Management and Quality Management standards (e.g. ISO10007 and ISO9000). Although ideally Configuration Management can be regarded as the general 'umbrella' process of managing change (Lyon, 2001) the focus is on document control and administration. Here we examine design change in terms of how descriptions change. This complements research on linkages among parts and analysis of the propagation of change along these connections (Eckert et al 2004, Clarkson et al 2004, Jarratt et al 2004).

Descriptions: Designers can interact with a physical object itself to make modifications, but mostly they rely on more abstract representations. The starting point of change can be represented by an existing design or abstract descriptions such as drawings, CAD files, indexed knowledge and in-service records. Whilst a design is being generated it exists as descriptions which may be partial and fragmented compared to the initial or finished design. Even physical prototypes may be partial descriptions. The process of designing is a transformation of descriptions. Appropriate and usable descriptions are critical. A description can refer to a specific object, perhaps an existing design, and represent certain features of this reference object. A description, once modified does not strictly describe its reference object, although it retains several features. A description may also exist independently of a reference object or refer to many potential objects.

Design descriptions concentrate on particular aspects of the design: the CAD models describe geometry, FEA models describe mechanical properties, the functional models describe functions etc. All but the simplest products have more detail than a designer can easily think about. Design features and elements are therefore grouped into higher level parts. For example a car engine is described hierarchically as engine block, pistons, sump etc. rather than a detailed list of all components. When thinking about those parts we again pick up on aggregate features, for example the sump consists of the sump, seals etc. Only when we focus on the sump itself, we might start looking at specific details which will determine the price and quality of a product. Descriptions at different levels in this hierarchy are used for different purposes during the design process.

Practically designers often talk and think about one design by reference to other objects. These objects may be competitors' designs or sources of inspiration. Just pointing to a familiar object can be a parsimonious representation from which designers can recreate details. Such object references do not necessarily pick out relevant features explicitly. Design descriptions through object references can exist on many levels of detail and be temporary and fleeting as designers focus on them (Kosslyn, 1980, 1994). A new design can inherit global properties and detailed features from an existing design which may never be explicitly questioned. Object references are an essentially different form of abstraction from the hierarchical descriptions which are based on a conscious selection of features. The object itself remains the primary mental cue for organising other descriptions derived from the object itself.

A change process involves more than just descriptions of objects and features. The ways that designers conceptualise the context in which they work and the process by which they generate a product are also descriptions. Further the descriptions are connected and influence each other. Indeed key drivers of the actions in change processes are mismatches between descriptions.

Mismatches and mistakes: Mismatches between how a design proposal behaves and its desired performance (or user requirements) become critical as a design progresses. They need to be rectified before the design can be brought to the market. However, changes may introduce new mismatches – mistakes are made - as well as remove others. We note that design proposals are essential prompts and tests of user requirements which may not be set firmly at the start of a design process.

The processes of change are not always smooth and well directed. Mistakes occur in many ways. Designs, or parts, may be inherited wrongly from previous designs or newly designed parts may contain mistakes. These cause disruption to a design process and need further changes to put them right. But mistakes, if based on shared assumptions about capabilities and competence across the design team or buried in the complexity of the project schedule, may not come to light until late in the whole process. By then many of the parts of the design are finished and tested in their details so fixing the mistakes can be costly. Although the majority of alterations made to parts of a design have little impact, a few can unexpectedly propagate to other parts, perhaps not even directly linked to the initially changed component. This knock-on effect has been referred to as an "avalanche" of change (Eckert *et al.*, 2004; Fricke *et al.*, 2000) or the "snowball effect" (Terwiesch and Loch, 1999). Such an event can have a major affect on the budgets and schedules of a particular project as well as more generally on the way a company and its projects are organised.

The exact point in time when an engineering change occurs during product development can have a dramatic impact upon the schedule and cost of the project (Lindemann & Reichwald,1998). Costs rise the later an alteration is implemented: changes that 'just' require alterations in the design phase are much cheaper than those that occur during production ramp-up. Once production has started the impacts spread further into many other business processes. Engineering changes lead to an increase in the amount of product data that must be handled, especially if one change propagates many further changes. Ensuring that only correct, current data is available can be a major problem (Wright, 1997). Further, changes affect the supply chain. Wänström et al. (2001) found that there was no consistent approach to handling the phase-out of old and phase-in of new parts.

Industrial studies on complex products: Since 1999 we have been carrying out empirical studies of change processes in complex engineering products including a helicopter manufacturing company (Eckert et al, 2004) and an ongoing study in a diesel engine company. Initially we concentrated on the overall process of change and identified the lack of understanding of dependencies between components as a major problem in managing changes and predicting their effects (Jarratt et al.2004a). In response a matrix-based change prediction method has been developed (Clarkson et al. 2004) as well as a method to capture the linkages between components (Jarratt et al., 2004b). The observed shortcoming of not recognising dependencies was confirmed in a parallel study with a jet engine company.

These industrial studies led to a distinction between two types of change (Eckert et al. 2004). First *initiated changes* are caused by outside factors, such as new customer requirement or new legislation. Second, what are called *emergent changes* arise from problems with a current design proposal in terms of mismatches with requirements and specification. These can be caused, by mistakes, supplier constraints and factors internal to the process such as resources, schedules and project priorities across the company.

Regardless of the type of the change, companies used the straightforward sequence - assess, generate possible solution, analyse implications and implement. Even if the process through which initiated and emergent changes are resolved is very similar, the attitude with which the change is handled is very different. If an emergent change arises from a mistake or a late modification from the supplier, designers often resent it as avoidable; while initiated changes are considered as normal business and designers regard their company's ability to accommodate customers' wishes as an asset.

Companies employ two strategies to manage engineering change (i) Changes by a core design team. Because a change often occurs when members have moved to another project, a change interrupts this project or is delayed until spare time becomes available. Changes generate additional connectivity between products. (ii) Changes are carried out by dedicated change teams, who have to invest considerable time and effort into learning about the original product, often through the original designers. Many companies employ a mixture of both strategies, using dedicated teams to handle routine changes and experienced designers to handle difficult changes.

These extensive studies on helicopters, diesel engines and turbo-jets (products with many parts, strong connections among parts and processes involving many different areas of expertise and capability) show that design change is complex and difficult to manage. We have established that classifying types of change, understanding the connectivities and linkages among parts, and providing tools to help this analysis, are valuable to the companies. We have also examined the some of the sources of complexity in change processes. For example, the structure of connectivities among parts and pathways for change propagation are sources of complexity. A type of chaotic behaviour can be identified – with small, apparently insignificant changes in one part causing unpredictable and potentially large changes to the design as a whole. A small change propagates in an 'avalanche' of changes, whose scope and magnitude are hard to predict. We now consider this and other types of complexity which arise during change.

Complexity

An analysis of complexity across the whole design process (Earl et al 2005) started from four main elements of design and product development. Figure 1 shows these elements: (i) Product - the design and its geometrical, physical and engineering characteristics, (ii) Process - the tasks used to create the design from initial specification to final product. These include the organisation, culture and resources of the company and its supply chain. (iii) Designer - the capabilities, knowledge and experience of designers and (iv) User – specifications, requirements and markets. The environment for this designing 'world' includes contexts, theories and influences as well as available methods and tools. Each element is a potential source of complexity, but perhaps more important is the recognition that complexities in design often arise from the relations between these four elements. Change complexities arise from these relations.

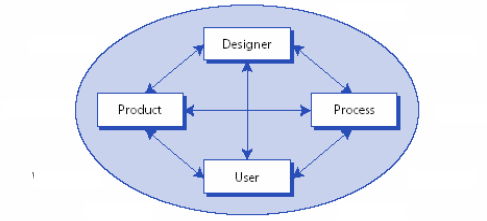


Figure 1 Elements of design

In this paper we take a complementary view of the sources of complexity. In creating a new design each of the four elements has a static and a dynamic component. For example in a change process the product has known and static parts as well as those parts which are subject to change. The process element may be dynamic but at a longer time frame than product. During each design the process will remain relatively static. Further, across different industries and types of product, the mix of static and dynamic components will vary. Mature products have extensive static elements in established product architectures, supply chains and well rehearsed processes with few large uncertainties. More innovative products have many dynamic components in each of the four elements. Intermediate types of design such as customised products, may have static product architectures but a dynamic and responsive process.

Characteristics of complexity- connectivities and dynamics: Complexity has enjoyed increasing attention as a research topic over the last decade. A science of complexity is taking shape, although complexity is still viewed in different ways according to the field of interest. However, there two key elements apparent. These are first the structural complexity of parts and connections, and second the dynamic complexity of behaviour. In the tradition of cybernetics (Wiener, 1948) complexity is distinguished from complicatedness. A system is complicated when its behaviour is predictable, even if it contains a large

number of parts. On the other hand a complex system cannot be predicted in detail, even though its overall behaviour may be bounded. Complex systems are dynamic, changing and evolving over time. The underlying connectivity representing how the different parts are related determines the constraints and potential for behaviour. Simon (1969) considers the complex engineered or 'artificial' systems as almost decomposable, that is they are hierarchical to some extent, but not fully decomposed into separate, independent parts. Connectivities of a complex design form a lattice structure rather than a tree structure although the latter is often an adequate approximation for almost decomposable systems. A familiar example of a complex system with underlying connectivities and associated dynamics are road networks. The network of roads itself or more usefully the sets of routes are a connected 'backcloth' (Johnson 1983a). These routes overlap and interact with each other. These interactions transmit dynamic effects between different parts of the road system changing the flows of road traffic over the connected set of routes.

Connectivity and dynamics can also be viewed in terms of information complexity. This expression of information content or entropy (Jaynes, 1957, Frizelle & Suhov 2001) takes into account both the underlying order described by connectivities in structure and the overall uncertainties of dynamic events on that structure. Axiomatic design (Suh 2001) aims to minimise complexity through reducing the connectivity between parts. This in turn is expected to reduce the uncertainties of dynamic events such as change propagation and unexpected behaviours. Modelling connectivities can improve product development processes as shown in the application of design structure matrix (DSM) based methods to represent connectivity and identify where dependencies can be reduced (Eppinger et al, 1994). Related models represent the connectivities of process tasks in product development directly (Clarkson and Hamilton, 2000; O'Donovan et al, 2004).

Complexity is also about uncertainties in dynamically changing systems. Chaotic systems (e.g. Alligood et al. 2001) are examples of bounded (ie limits to behaviour) unpredictability. An adaptive system changes its connectivities and dynamic behaviour in response to its environment whilst coevolving systems develop mutual changes of structure and behaviour (e.g. Kauffman and Macready, 1995). Unlike chaotic behaviour these dynamics are unbounded in the sense that as changes to structure are allowed, new structures and radically new behaviours can occur. These distinctions are summarised in Figure 2.

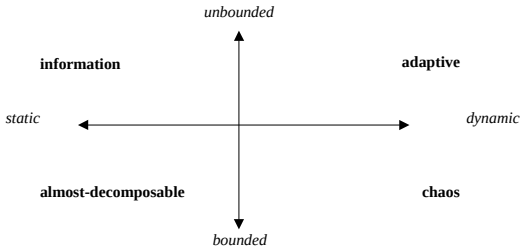


Figure 2 Types of complexity

Timescales: In drawing a distinction between static connectivities and dynamic behaviour we note that this is relative. For example the connectivities in a product architecture or organisational structure develop more slowly than individual products or the rapid changes during product development. Over an extended timescale, individual product developments and the change processes within them will affect underlying connectivities in product architectures as well as the organisational structures of the company. These changes to underlying connectivities - the background structure for product developments - come about indirectly through management and strategic planning. On the other hand changes to the background structure directly affect product development.

Over a long period designs and processes both affect each other and mutually change. For example new people design different products and the new properties of these products require different people to develop them further. At an even longer timescale one could argue that the processes that designers carry out to create a product remain relatively constant, while the products that they are creating change. In this sense the descriptions of the products change or 'move' over the background of the processes.

But complexity as seen by participants in design at all stages, levels and timescales is dependent on the descriptions which are employed to represent products, processes, users and designer's knowledge and expertise. Many descriptions, each partial, are used together. Hanks et al (2003) present an analysis of problems with using descriptions across domains especially the propagation of misunderstandings arising from inadequate descriptions of design requirements. They provide evidence that attention to domain semantics and avoidance of informal heuristics can clarify connections within and between descriptions. Static complexities come from these connections within and between descriptions. For example a geometric in CAD has a complex structure of parts and layers. This shape description is intimately linked to a material strength description; indeed there may be considerable overlap between them. During product development descriptions are modified as new parameters are calculated and properties analysed. New descriptions may be added or previously abstract and uncertain descriptions become more detailed. For example a new requirement from a customer which initiates change may involve a new description; a test result may reveal previously unexpected behaviour (although we remark that new behaviour is rarely completely unexpected) which necessitates a new description. Descriptions can also be found to be inconsistent, for example when mistakes reveal between proposed design and user requirements as inconsistent. In each case a change process involves tasks involving actions on descriptions.

Change processes take place against a highly structured background of existing products, newly designed parts and company processes as well as designers' expertise and knowledge. Change processes act on descriptions. In the next section we present a simple three level model of Background, Descriptions and Actions to help identify sources of complexity in change processes.

Background, descriptions and actions

The background might present the underlying connectivities of parts of a product type and general physical principles for the behaviour of that type of product. Descriptions of a specific design proposal are developed through iterative action of synthesis, analysis and test. In a sense the product 'flows' through the processes (Earl et al 2001). Complexity arises from interactions between 'flow' and background. A static background structure of connectivities is expressed through various mediating descriptions of product, process, designer and user (Figure 1). Some descriptions are changed though actions. This general picture of design is summarised in Figure 3. Complexity arises at each level in this model, and in the interactions between levels. The background represents the underlying order expressed through structure and connectivity whilst the actions represent dynamics and uncertainties. Actions take place on descriptions or directly on the background for innovative and radical changes where appropriate descriptions may not be available.

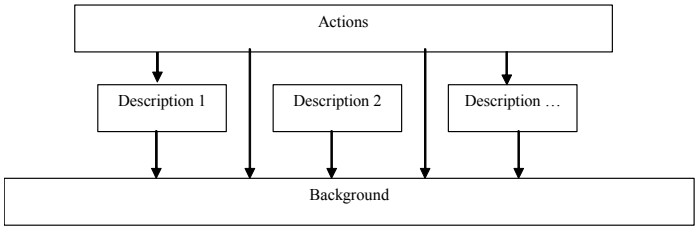


Figure 3 Three layer view of design

The background can evolve slowly over time and is essentially static. Examples of elements in the background are (a) The starting point of a change process, perhaps a competitor's product, (b) manufacturing capabilities and the technical properties of materials (which form the background for manufactured shapes) and (c) the physical principles for devices of a certain type. The structure of the background arising from connectivities can be analysed through multidimensional relations with methods such as Q-analysis (Johnson, 1983a,b, Johnson 1995), which models both connectivities and dynamics within a common hierarchical framework. The background is accessible through descriptions which have properties and structures of their own. The types of complexity discussed in the previous section have their focal points at different parts on the three level model (Figure 4). Adaptive (and co-evolving) properties are mainly on the actions level. Chaos is mainly concerned with how the structure of the background determines the predictability (or otherwise). For example, how change propagates depends on established

linkages and connections among parts. Information complexity as information is about possible behaviours within the background connectivities. Almost decomposable systems characterise the descriptions of engineered or artificial systems (Simon 1967). Eppinger et al. (1994) and Suh (2001) both consider complexity reduction by understanding connectivities in the descriptions used.

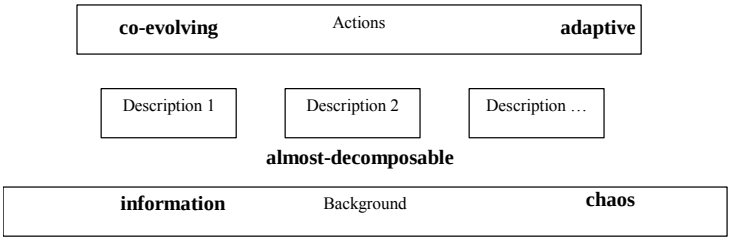


Figure 4 Types of complexity most appropriate at each level

Problems in design change can arise from the misalignment between background, descriptions and actions. For example descriptions may not be consistent with the actual background or may distort its properties. Further, descriptions may have insufficient scope to cover all aspects of the background.

In the background element of the model there will be many properties of the product which are beyond the control of an individual designer, perhaps inherited from past products or through product platforms adopted by the company. Some properties are side effects of other highly desired properties. For example if a material is chosen for its weight properties, the thermal or conductive properties are side effects. Manufacturing processes enforce properties on products. The background also includes the physics of how the product works. For example the functioning of a jet engine depends on the physics of airflow through compressors. General characteristics of performance are part of the background such as the potentially chaotic behaviour that can occur near conditions of optimal performance of a jet compressor. The company organisation, supply chain, markets, the skills levels or the personalities of the designers and a whole host of other properties can be seen as a background against which the designers operate on a particular project.

The idea of connectivities in a background can be applied more widely to design processes across an industry sector. An example is the development of fashion in clothing (Eckert & Stacey, 2001). When a season's fashion appears, it seems fairly coherent with similar trends, colours and materials. However designers have not directly collaborated. Perhaps they looked at the same sources of inspiration and followed the same trend predictions. They are connected through suppliers and customers who provide feedback on the developing design and constraints on materials and tools they make available. As the new fashion appears in the shops, designers look at it and use it as a way to refine their own ideas.

A description is an abstraction and a selection of features. For example a CAD model covers shape but not surface micro geometry. However tiny variations in the surface from manufacturing processes can have a large effect, for example where fatigue will occur over the life time of the product. During the design process direct physical interaction with the background is limited. Physical prototypes are built to test some properties, but otherwise designers create and operate on descriptions in what is referred to by Bucciarelli (1994) as an 'object world'. A design process involves actions on a range of descriptions. These may start with physical parts of the background (eg an existing product), through more abstract representations, and returns towards a direct interaction through a prototype and test. Delaying this direct interaction through using increasingly accurate product simulations is a current trend. Where designers ordered a test 10 years ago to see how a product or a part behaves, now there is only time for one test and little iteration beyond it.

Change processes are strongly constrained by background structure and connectivities. Researchers advocate setting these up explicitly so as to make future changes easier. Martin and Ishii (1997) propose a method to analyse which margins will be critical for likely changes and design those into the product in the first place. Axiomatic design (Suh, 2001) advocates a structured approach to design with a clear assignment of functions to components or parts. Connectivities within the product itself are reduced and designers are more aware of the linkages and margins that do exist. The design process becomes less prone to mistakes

and the design more robust in performance. A side effect might be that a design is more resistant to change in the future. These methods in setting up background structure to accommodate change will necessitate tradeoffs between current and future products as well as between product and process complexity.

Changes are often difficult to carry out, because they require considerable effort to capture the background - understanding the current design and the reasons why it is the way it is. Design rationale is rarely captured and documentation does not identify potential changeability of parts. Although these and similar problems in change seem to come from of the background process they actually arise from the description layer. This is recognised in a major new UK research 'grand challenge' that is looking at providing 'immortal' design information, ie background, description and action records for existing designs.

An example of design change

As we indicated above several studies have been conducted on change. Eckert et al 2004 report change processes in Westland Helicopters in some detail. Without going into extensive details these are products which integrate many complex subsystems, from airframe to controls, avionics, power systems and transmissions, which are all customized and thus the targets of change processes. The background covers strong connectivities among its many, wide ranging, elements from existing product range and types, assessments of product performance in service, technical knowledge and expertise through to established processes for subsystem design and integration. The background is deeply embedded in company practices and capabilities. Descriptions used by designers have an extensive range across the company including for example, customer specifications, CAD, engineering analysis and simulations, test results and plans for process including schedules.

The customisation of a helicopter, such as the current fleet of presidential helicopters, involves considerable design effort. Westland does not have a base product, but uses various existing designs as a starting point for each new version. Therefore the company has incompatibility problems between the various designs used as the starting point as well as changes that come in later. This background is not a nicely structured representation of the problem; it is a medley of elements whose connectivities include incompatibilities. Other elements of the background are more structured including technical constraints on product architecture, company processes in tendering, design and manufacture, and supply chain relations. Our studies suggest that recognition of the extent of the internal background - context, starting points and constraints - on which the new design is based is as important as the external imperatives of customer need. The background extends further to the connections and linkages between parts of the helicopter. Mapping this aspect of the background (Eckert et al, 2004, Jarratt et al 2004b) has helped the company to appreciate sources of complexity. The map of connectivities is a first step in understanding the 'amount of uncertainty' or information complexity at the start of the design process. Even with a map of connectivities changes can propagate unpredictably with a chaotic-like complexity.

In a helicopter most components are affected to some extent by overall product parameters, such as balance or rotational frequency which can lead to a wide range of change propagation. Changing just one component can alter these overall parameters which are then brought back on track by changing several other components and so on. Often changes go on in parallel, which although unproblematic on their own can cause large problems if they happen at the same time. For example a new version of a military helicopter (in the EH101 series Figure 5) a few years ago required a troop seat to be fitted to the inside of the helicopter and a large sensor on the outside of the fuselage. The fuselage could have carried the additional weight of one of the changes, but not both, so that the fuselage needed to be reinforced, taking up more space on the inside of the craft. However the fuselage cannot be reinforced without upsetting the balance of the entire helicopter. Therefore other parts needed to be rearranged in the craft. Every time a component is moved, geometry needs to be re-evaluated and possibly changed with the cables or pipes leading to it rearranged. The knock-on effects were very costly, but as the company had contractual obligations to carry out both changes they had no choice. Another example of design difficulties caused by change is the addition of a large and heavy radar to the front of the craft which required changes to the tail of the craft for balance and manoeuvrability. In these examples, overall product parameters are cutting across descriptions of the product as decomposed into functional or technology subsystems.

This change caused the company many problems and several designers independently commented on it as an example of how Westland struggles with changes. The complexity model proposed in this paper helps to explain this. The change was difficult on all the layers proposed: the background of the fuselage they started with did not have the redundancy to accommodate the change. A decision was taken early in the design of the EH101 series on the extent of margins for parts and their behaviour, including overall margins for the product. These margins were designed in and allowed for uncertainties in product performance and operational conditions. Margins were eroded from version to version over the process of many modifications in the evolutionary development of the EH101. These margins can cause cliff edge effects, where a tiny change in a design parameter near a margin can have a huge effect, perhaps catastrophic, on the behaviour of the part and the whole design. Similarly, a small change in behaviour of a part, within allowed margins, can have large knock-on effects across the product. While theoretically the behaviour near each margin is predictable, the overall effect, as a design moves closer to several margins in different parts, is unpredictable and chaotic behaviour. In this case the changes are originally evaluated separately with no single one pushing the product over the margin.

Figure 5 Westland EH101

The design of the helicopter is highly interconnected, where parts like the fuselage connect many aspects of the product together, effectively transmitting information between the parts. For example it would be theoretically possible to mount troop seats on the floor, thus distributing the weight over a larger area. The present helicopter design of the EH101 series is neither modular nor does it follow principles of form and function division, largely because of concerns of weight penalties. Margins are not noted in CAD models or 2D schemas, therefore companies depend on designers remembering and communicating changes to margins among themselves. In the example above, adding sensors and troop seats fall under the responsibility of different teams, who are only linked through a common interest in the properties of the fuselage and overall product parameters. This organization and associated project division has evolved to meet the core challenges of helicopter design. Problems arise when designers try to act on unconnected parts of the background, using descriptions from their own expertise area. The further the change propagates across the product, the less well the organisation is equipped to deal with it, especially if there is a lack of overview. Such an overview is important in dealing with changes such as adding the heavy radar at the front with its associated changes to the tail of the craft.

Conclusion

In this paper we have reviewed recent work on change processes in design. A model of Background, Descriptions and Actions distinguishes the static background for design development from the actions on descriptions to effect design change. The background layer describes the inherent and persistent structural properties of the product and processes. Complexities can include underlying chaotic behaviour of both products and change processes. The descriptions layer reflects that designers interact primarily with descriptions rather than directly on the background. Fragmented descriptions or those misaligned to the structure of the background may miss critical properties only revealed at later test. The actions layer describes change processes and reflects the complexity of the process of adaptation (and sometimes coevolution) of the design to requirements.

References

Alligood K T, Sauer T, & Yorke J 2001 *Chaos: an introduction to dynamical systems*, Springer.
Bucciarelli, L.L. (1994): *Designing Engineers* MIT Press.

Clarkson P J, Hamilton J, 2000,Signposting: a parameter-driven task based model of the design process *Research in Engineering Design*, 12 (1), 18-38
Clarkson, P.J., Simons, C.S. and Eckert, C.M. (2004) 'Predicting change propagation in complex design' in *ASME Journal of Mechanical Design*, 126 (5), 765-797
Cohen, Navthe S, Fulton R 2000, 'C-FAR Change favorable representation' Comp Aided Des 32, 321-38
Cross, N. (1989*), Engineering Design Methods*, Chichester: John Wiley & Sons
Earl C., Johnson J. and Eckert C 2005 'Complexity', Ch 7 in *Design Process Improvement - a review of current practice*, Springer
Earl C, Eckert C, Johnson J (2001) Complexity of planning in design, *ASME DETC'01*, Pittsburgh,
Eckert C, Clarkson P J and Zanker W, 2004 'Change and customisation in complex engineering domains' in *Research in Engineering Design*, 15 (1), 1-21
Eckert C and Stacey M, 2001 'Designing in the context of fashion - Designing the fashion context' in *Designing the Context Symposium*, Delft University of Technology, The Netherlands, 113-129.
Frizelle G. and Suhov Y. M, 2001 "An entropic measurement of queueing behaviour in a class of manufacturing operations", *Proceedings of Royal Society Series A*, 457, 1579-1601,
Fricke E et al, 2000 Coping with Changes: causes, findings and strategies, *Systems Engineering*, 3, 169-79.
Hanks K, Kimberly S, Knight J, 2003 'Improving Communication of Critical Domain Knowledge in High-Consequence Software Development: an Empirical Study', *ISSC'03*, Ottawa, Canada
Jaynes E. 1957, Information Theory and statistical mechanics, Physical; Review 106 620-630
Jarratt, T.A.W. (2004) 'A model-based approach to support the management of engineering change', PhD-thesis, Cambridge University Engineering Department
Jarratt T, Eckert C, Clarkson P.J & Stacey M 2004a 'Providing an overview during the design of complex products: the development of a product linkage modelling method' in *DCC'04, MIT*, 239-258
Jarratt T., Eckert, C. and Clarkson, P.J. (2004) b 'Development of a product model to support engineering change management' in *TMCE 2004*, Lausanne, Switzerland, 1, 331-342
Johnson J H (1983a). Hierarchical Set Definition by Q-analysis, Part I. The Hierarchical Backcloth. *International Journal of Man-Machine Studies* 18(4): 337-359.
Johnson, J 1983b. Hierarchical Set Definition by Q-analysis, Part II. Traffic on the Hierarchical Backcloth. *International Journal of Man-Machine Studies* 18: 467-487
Johnson J 1995 The multidimensional networks of complex systems, in: Networks in Action. Springer
Kauffman S &Macready W 1995 Technological Evolution and Adaptive Organizations *Complexity,*1 26-43
Kosslyn, S M 1980 *Image and Mind* Harvard University Press, Cambridge MA (1980)
Kosslyn, S M 1994 *Image and Brain* MIT Press, Cambridge MA (1994)
Terwiesch, C. and C. Loch, H, 1999, Managing the Process of Engineering Change Orders: The Case of the Climate Control System in Automobile Development. *J Product Innovation Management,* 16, 160-72
Lindemann, U. and R. Reichwald, (1998*) Integriertes Änderungsmanagement*, Berlin: Springer.
Lindemann, U., R. Kleedorfer, and M. Gerst, (1998) The Development Department and Engineering Change Management, in *Designers: The Key to Successful Product Development*, E. Frankenberger, P. Badke-Schaub, and H. Birkhofer, Editors, Springer, London. p. 169-182.
Lyon, D.D.(2001*) Practical CM - Best Configuration Management Practices*, Butterworth, Oxford
Martin, M.V. and Ishii, K. (1997) "Design for Variety: Development of Complexity Indices and Design Charts," DETC97/DFM-4359, *ASME DTC/CIE Proceedings* CD, ISBN 0-7918-1243-X.
O'Donovan, B. Eckert, C.M. & Clarkson, P.J. (2004) Simulating Design Processes to assist in design process planning , *ASME DTM* Salt Lake City, Utah, USA, September 2004.
Pikosz, P. and J. Malmqvist. (1998) A Comparative Study of Engineering Change Management in Three Swedish Engineering Companies, *ASME DTM*. 1998. Atlanta, GA, USA: ASME.
Simon, H. (1969). *Sciences of the Artificial*. MIT Press, Cambridge, MA.
Suh N P (2001) *Axiomatic Design - Advances and Applications*, Oxford University Press, New York
Wänström, C., P. Medbo, and M.I. Johansson. (2001) Engineering Change from a Logistics Perspective. in *NOFOMA Conference* - Nordics Logistics Research Network. Reykjavik, Iceland.
Wiener, N. 1948, *Cybernetics or control and communication in the animal and machine*, MIT Press,
Wright, I.C.(1997) A Review of Research into Engineering Change Management: Implications for Product Design*. Design Studies*, 18: 33-42.

Creativity in the Design of Complex Systems

Neil Maiden & Sara Jones

Centre for HCI Design, City University, London EC1V 0HB
http://hcid.soi.city.ac.uk/people/Neilmaiden.html or Sarajones.html

Abstract: This position paper describes applied research to support more creativity in the specification and design of complex socio-technical systems and in particular air traffic management systems. It summarizes a series of creativity workshops, the air traffic management projects to which they were applied, and results from these workshops for the projects. The paper ends with a short discussion and directions for future research into creativity in the requirements and design phases of complex systems development.

Keywords: Requirements, Creativity, Innovation.

Introduction

There has been little work on introducing creativity into the development of socio-technical systems, though there are some exceptions, such as the work by Couger et al at the University of Colorado. The closest most projects will typically come to introducing creativity into the development process is through the use of brainstorming or RAD/JAD techniques where some constraints on idea generation are removed for a short period. The situation is worse still in the development of complex, and particularly safety-critical systems, where methods introduced to deal with complexity (such as the RUP (Jacobsen et al. 2000)) and to deliver safe systems (see, for example, Leveson et al. 2001) can sometimes limit innovation and creativity. Many methods adopt a reductionist approach that limits opportunities for combinatorial creativity, whilst the need to capture full rationale means that new ideas resulting from innovative thinking need extensive and time-consuming verification before their inclusion in a design.

Integrating creative techniques into the structured processes that are needed to handle complexity often complicates innovative design processes. In particular we need new techniques to integrate creative thinking with assessments of risk of tight coupling that are used to handle complexity and safety. The use of creativity in requirements processes can bring enormous benefits. Requirements are a key abstraction that can encapsulate the results of creative thinking about the vision of an innovative product. This vision can then inform the rest of the development process. Ours is a view shared by Couger (Couger 1989), who suggests that the use of creativity techniques nesr the conclusion of requirements definition can be particularly beneficial. However, this is a view that requirements engineering researchers and practitioners, with their current focus on elicitation, analysis and management, have yet to take on board.

In RESCUE, our integrated process for requirements engineering, we combine the systematic approach offered by structured methods with opportunities for real creativity. Processes and methods needed to model, analyze, specify and sign-off stakeholder requirements are integrated with creative thinking techniques as described below. We have applied this process in a number of UK and European large projects including CORA-2 (a socio-technical system for resolving conflicts between aircraft), DMAN (a system for managing departures from major airports), MSP (a multi-sector planning system for gate-to-gate) and EASM (a socio-technical system for enhanced airspace management). All of the systems were large and complex. The requirements process for each lasted a minimum of 10 months. The two completed projects – CORA-2 and DMAN – specified 22 and 15 use cases and 400 and 700 requirements respectively. The MSP and EASM systems are of a similar size. In each case, it has been deemed worthwhile to make a significant investment in the requirements process, and in particular, in the use of techniques that encourage creative thinking about future systems.

Managers in the domain of air traffic management are understandably cautious about big new ideas. Change is potentially risky, and very expensive, not least because of the need for extensive retraining of air traffic controllers. However, there is an inescapable need to move on from the technologies of the 1960's, including paper flights strips often still annotated by hand, in order to cope with increasing demand for air travel.

The RESCUE Process

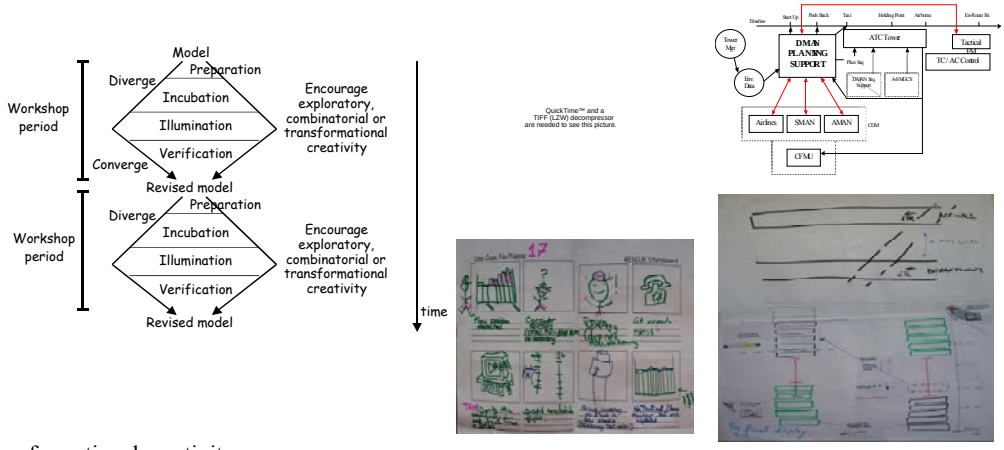
RESCUE was developed by multi-disciplinary researchers at City University, working together with experts in the domain of air traffic management from Eurocontrol (Maiden et al. 2003a, 2003b), to help address this need. It supports a concurrent engineering process in which different modeling and analysis processes take place in parallel. The concurrent processes are structured into 4 streams. Each stream has a unique and specific purpose in the specification of a socio-technical system:

1. Human activity modeling provides an understanding of how people work, in order to baseline possible changes to it;
2. System goal modeling enables the team to model the future system boundaries, actor dependencies and most important system goals;
3. Use case modeling and scenario-driven walkthroughs enable the team to communicate more effectively with stakeholders and acquire complete, precise and testable requirements from them;
4. Requirements management enables the team to handle the outcomes of the other 3 streams effectively as well as impose quality checks on all aspects of the requirements document.

Creativity workshops normally take place after the system boundaries are specified, to discover and surface requirements and design ideas that are essential for system modeling and use case specification Inputs to the workshops include the system context model from the system goal modeling stream and use case diagrams from the use case modeling stream.

Creativity Workshops in RESCUE

Creative workshop activities were designed based on 3 reported models of creativity from cognitive and social psychology. Firstly, we design each workshop to support the divergence and convergence of ideas described in the CPS model (Daupert, 2002). As such each workshop period, which typically lasts half a day, starts from an agreed current system model, diverges, then converges towards a revised agreed model that incorporates new ideas at the end of the session. Secondly, we design each workshop period to encourage one of 3 basic types of creativity identified by Boden (Boden, 1990) – exploratory, combinatorial and transformational creativity. Thirdly, we design each period to encourage 4 essential creative processes reported in (Poincare 1982): preparation, incubation, illumination and verification. The incubation and illumination activities are determined by the type of creativity that we seek to encourage. Figure 1 depicts the creativity workshop structure and demonstrates examples of outputs from the DMAN workshop, which are typical of what might be expected from a creativity workshop of this kind. Within this workshop framework, we implement a range of creativity techniques – from brainstorming and analogical reasoning to encourage exploratory to storyboarding for combinatorial creativity and constraint removal for



transformational creativity.

Figure 1. Clockwise from the top-left – the structure of a RESCUE creativity workshop, a typical workshop environment, an emerging system architecture resulting from combinatorial creativity, a

visualization of controller information displays resulting from transformational creativity, and a storyboard depicting behaviour specified in one use case specification.

Discussion and Future Work

The DMAN operational requirements document was delivered to the client in 2004 and provided us with the chance to analyze the outcomes from the creativity workshop on the final requirement document to determine their impact, as reported in Maiden & Robertson (2005). This analysis revealed important associations between results of the creativity workshop and elements of the use case specification, which in turn led to more detailed scenarios that were walked through to discover and document more requirements for the DMAN system. Although we do not claim that these requirements would not have been discovered and documented without the creativity workshops, we do report an association between the workshop outcomes and their documented discovery later in the requirements process.

We are also using workshop data to validate and extend a descriptive model of creative requirements engineering based on models that underpin the workshop design (Boden 1009, Daupert, 2002, and Poincare 1982). We are using protocol data to investigate life histories of creative ideas from conception to verification, and linking these histories to patterns of stakeholder communication and artifact use. We believe that these models have general applicability to the design of interactive systems of which air traffic management systems are an example.

Finally, we are also investigating how to integrate creative thinking techniques into other RESCUE sub-processes. One limitation is that the creativity workshops are expensive and time-consuming, so fostering and guiding creative thinking within other sub-processes involving fewer stakeholders is desirable. Therefore, we are currently exploring how to extend the ART-SCENE scenario walkthrough tool (Mavin and Maiden, 2003), designed to ensure requirements correctness and completeness, to support creative thinking.

Acknowledgements

The authors wish to thank all of the workshop participants, and acknowledge the support of Eurocontrol, NATS and Sofreavia in the development of the paper.

References

[1] Boden M.A., 1990, The Creative Mind, Abacus, London

[2] Couger J. D. 'Ensuring Creative Approaches in Information System Design', report no. 89-2, Centre for Research on Creativity and Innovation, College of Business at University of Colorado, Colorado Springs.

[3] Daupert, D. (2002) The Osborn-Parnes Creative Problem Solving manual. Available from www.ideastream.com/create.

[4] Jacobson I., Booch G. & Rumbaugh J., 2000, 'The Unified Software Development Process', Addison-Wesley-Longman.

[5] Leveson N, de Villepin M., Srinivasan J., Daouk M., Neogi N., Bachelder E, Bellingham J., Pilon N. & Flynn G., 2001, 'A Safety and Human-Centred Approach to Developing New Air Traffic Management Tools', Proceedings Fourth USA/Europe Air Traffic Management R&D Seminar.

[6] Maiden N.A.M., Jones S. Flynn M., 2003a, 'Innovative Requirements Engineering Applied to ATM', Proceedings Joint Eurocontrol/FAA ATM'2003 Conference, June 2003, Budapest.

[7] Maiden N.A.M., Jones S.V. & Flynn M., 2003b, 'Integrating RE Methods to Support Use Case Based Requirements Specification', Poster paper, Proceedings 11th International Conference on Requirements Engineering, IEEE Computer Society Press, 369-370.

[8] Maiden N.A.M., Manning S., Robertson S. & Greenwood J., 2004, 'Integrating Creativity Workshops into Structured Requirements Processes', Proceedings DIS'2004, Cambridge Mass, ACM Press, 113-122.

[9] Maiden N.A.M. & Robertson S., 2005, 'Developing Use Cases and Scenarios in the Requirements Process', to appear in Proceedings 26th International Conference on Software Engineering, ACM Press.

[10] Maiden N.A.M. & Rugg G., 1996, 'ACRE: Selecting Methods For Requirements Acquisition, Software Engineering Journal 11(3), 183-192.

[11] Mavin A. & Maiden N.A.M., 2003, 'Determining Socio-Technical Systems Requirements: Experiences with Generating and Walking Through Scenarios', Proceedings 11th International Conference on Requirements Engineering, IEEE Computer Society Press, 213-222.
[12] Pennell L. & Maiden N.A.M., 2003, 'Creating Requirements – Techniques and Experiences in the Policing Domain', Proceedings REFSQ'2003 Workshop, June 2003, Velden Austria.
[13] Poincare H., 1982, The Foundations of Science: Science and Hypothesis, The Value of Science, Science and Method, Univ. Press of America, Washington 1982

Diversity as a Determinant of System Complexity

Brian Sherwood Jones, Paul Anderson

Digital Design Studio, Glasgow School of Art, House for an Art Lover, 10 Dumbreck Rd, Glasgow G41 5BW.
http://www.gsa.ac.uk/dds/

Abstract: Diversity of viewpoint is discussed as a major determinant of system complexity – in particular as affecting project complexity during design, development, manufacture and roll-out. The viewpoints of particular interest are those arising from diversity of implementation technology, and from multiple specialist engineering disciplines. The paper discusses the historical treatment of complexity in design offices, the challenge of diversity to a project, and its impact on the formal and informal organisation of the project, and on individual cognition. Conclusions are drawn for system acquisition and for design and development approaches. Metrics for the challenge posed by the diversity of a project are needed, so that organisational capability can be aligned with project need – for project processes in particular. Complexity-as-diversity makes demands on the organisational climate and the treatment of multiple (potentially conficting) sources of bad news. Interactive visualisation is considered to reduce cognitive demands, support individual and team decision making, and to have an effect equivalent to reducing complexity.

Keywords: systems engineering, diversity, visualisation, cognition, culture.

Introduction

Thesis: If we take the primary task of systems engineering to be :"To identify, realize and maintain the requisite emergent properties of a system to meet customers' and end users' needs" (Hitchins, 2001), then in crude terms, the more emergent properties we need to manage, the more complex the task. This paper examines the complexity of the task facing the project as a whole, the task facing teams within a project, and the task facing the individual designer or specialist. Diversity relates closely to the number of emergent properties. A system can be considered as a way of looking at the world, and needs a viewpoint and an observer.

The overarching (or, less imperiously, underpinning) systems engineering standard, ISO/IEC 15288:2002 'Systems engineering - system lifecycle processes' represents a major international achievement in codifying, normalizing and harmonizing the actions of contributors throughout the system life cycle. On this basis, systems engineering can be considered to have four sets of processes operating through the life cycle. These are shown below in Table 1.

Agreement processes						
Acquisition					Supply	
Enterprise processes						
Enterprise environment management		Investment management		System life cycle processes management	Resource management	Quality management
Project processes						
Project planning	Risk manage - ment	Project assesment	Configuration management	Project control	Information management	Decision making
Technical Processes						
Stakeholder requirements definition		Requirements analysis		Architectural design	Implementation	Integration
Verification	Transition		Validation	Operation	Maintenance	Disposal
Special processes						
Tailoring						

Table 1 - Processes in ISO/IEC 15288 Systems engineering - system lifecycle processes

The technical processes will be effected by diverse specialist viewpoints. The viewpoint of the project processes is directly concerned with integrating this technical diversity. The enterprise and agreement processes are less affected by technical diversity and so are not discussed further. Managing emergent properties or viewpoints requires 'requisite imagination' (Westrum, 1998). Increasing diversity increases the number of ways in which things can go wrong, and the demands on requisite imagination and its vigilance. The job of representing a viewpoint on a project is termed a role. The role-viewpoint mapping may or may not be one to one.

Engineering view of complexity: The traditional engineering view of complexity is that it is driven by parts count, perhaps factored by the number of interfaces (e.g. Meyer and Lehnerd, 1997). The software equivalent is Source Lines of Code (SLOC). Capers Jones (1996) has shown the non-linear impact of complexity measured this way (see Table 2).

Size, function points	Size, KLOC	Coding %	Paperwork %	Defect removal %	Management and support %
1	0.1	70	5	15	10
10	1	65	7	17	11
1,000	100	30	26	30	14
1,000	100	30	26	30	14
10,000	1,000	18	31	35	16

Table 2 - Changing Types of Effort as Programs Grow in Size, Capers Jones (1996)

However, we would propose that it is likely that SLOC is confounded with diversity – it is plausible that larger systems have more viewpoints – and that diversity has an influence that is at least as great as that of parts count or SLOC. Certainly it is possible to have built artefacts of comparable complexity-as-parts-count with very different complexity-as-diversity, and this has not been recognised in engineering circles where the emphasis is on ease of mass manufacture rather than ease of design.

Driver for diversity: "Today's systems mostly draw their high levels of functionality and performance by combining system elements that contribute widely different functionality, and frequently employ distinctly different implementation technologies. A key goal of systems engineering is, therefore, to achieve novel product or service characteristics that appear to have come from a single, homogeneous entity, yet have actually been achieved through a carefully-crafted bringing-together of intrinsically dissimilar system elements." (Arnold et al 2002).

Simple example: A simple example was provided by Williams (2001) on the rail-wheel interface in the railways. The rail is part of the civil engineering of the track. The wheel is part of the mechanical engineering of the propulsion and suspension systems. Enabling the two disciplines to understand each other's requirements and constraints has led to dramatic improvements in both cost and safety.

Structure of paper: Some practical and theoretical background is provided on the management of complexity. The basis for the rest of the paper is that the challenge of diversity can be addressed by three complementary approaches:

- The use of systems engineering processes to develop a common picture among diverse roles to enable trade-offs to be made at a project level with mutual understanding.
- Considering the organisational climate and its ability to address interpersonal communication among diverse roles.
- Enabling specialist engineering to reach further "into the design" itself. This approach needs to meet the decision making and cognitive constraints of the individuals concerned. The proposal is that interactive visualisation supports this.

Some conclusions are offered.

Background – the practical management of complexity

This section is a precis of the historical evolution of specialist viewpoints, taking the last 50 years or so as the time frame. Fifty years ago, the number of specialists on a project would be limited to a few key

disciplines such as aerodynamics, structures, naval architecture. Beyond that, there would be fairly general viewpoints of electrical and mechanical engineering and the like. At the level of a draughtsman and non-graduate engineer, considerable generality would be expected.

The viewpoints discussed here are considered to arise from two sources:

- Diversity of implementation technology, e.g. hardware, software, people.
- Diversity of specialist engineering – the "–ities" such as usability, safety, reliability, Electro-Magnetic Compatibility (EMC), supportability, security. These viewpoing are also known as coherence requirements.

As something of a caricature, there was a time long long ago when technology and customer demand were sufficiently stable that the implementation of the specific technologies and meeting the specialist engineering needs of project stakeholders could be achieved by 'good design practice'. A designer could be expected to design for safety, maintainability, cost etc. with an adequate understanding of the materials, their fabrication etc.

This breadth of vision became impossible for complex systems (for reasons worthy of investigation), and so specialist viewpoints gained representatives on projects. Designers became split up by increasing diversity of implementation technology; electrical, mechanical, software etc. Integrating these implementations to meet the needs of specialist engineering became problematic. The specialist engineering disciplines are now well behind the curve; all they can do is react to a design once it has reached a level of maturity. The project manager sees a large part of his design budget going to people who do nothing but complain when it is too late. The truly cynical have pointed out that it is rare for coherence requirement owners to behave in a coherent manner and co-ordinate their complaints, or indeed to realise that such complaining is an important part of their role.

The response to this situation has been to 'divide and conquer' – to set up Integrated Project Teams (IPTs) that each deal with a part of the system. If the interfaces between the parts are clean, and the system-level emergent properties are addressed, this can work. However, there are some issues that are still to be resolved:

- The implementation-specific technologies still need an integrated view of the IPT scope of supply. This doesn't happen by itself.
- The specialist engineers are still behind the curve.

There is the real potential to be much closer to the design, but there are some obstacles to be overcome:

- The specialist resource is now divided among independent competing demands (perhaps several IPTs per individual), so you still have only half of them turning up.
- The specialist engineering community is now in the game of building big computer models and databases. These provide a firm basis for the discipline, but there is a direct conflict between building the model and being party to critical early design decisions.

Theoretical orientation

This paper is essentially practical in orientation, but does draw on a number of theoretical bases. This section provides a brief outline of the theoretical underpinning for the paper. Theory has been used only insofar as it affects the management of diversity and complexity.

Role of complexity theory: Addressing complexity is recognised as important to the future of engineering, and many authors have been drawn to complexity theory as an avenue of research (e.g. Bullock and Cliff, 2004). This paper considers that analogies with natural systems need to be treated with caution – there is nothing very natural about systems engineering in practice. Firstly, there is the option of predetermination '*reculer pour mieux sauter*' cf. Elster (1979). Secondly, it could be argued that project-based systems engineering is an explicit, and indeed unwelcome, alternative to the 'natural' evolution of an enterprise. If a system can be acquired or delivered as normal business, without a major project-based structure, then probably so much the better (so long as the necessary processes happen).

Role of systems theory: The relationships between systems theory, systems engineering and systematic engineering can be somewhat fraught. The view taken here is that systems enginering as set out in

ISO/IEC 15288 captures much of the necessary systems thinking, which can then take only an indirect role in considering the treatment of complexity.
Systems engineering and business excellence: The tenet behind the systems engineering presented here is that business excellence is the result of professional competence and process capability, as shown in Figure 1 (based on Arnold et al. 2002).

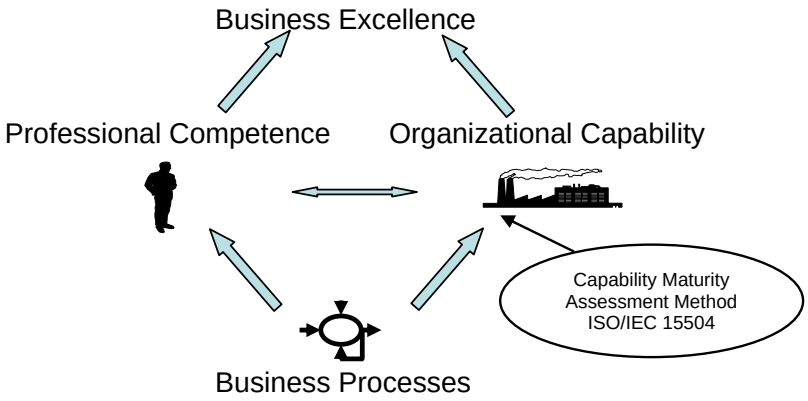


Figure 1 – Achieving business excellence

Requirements and design: It has become established in many systems engineering communities that it is much more effective to discuss evolving requirements than to discuss the emerging design. Gause and Weinberg (1990) have shown the benefits of doing this. The authors' experience is that project discussions on requirements are very different in nature from discussions of the design. However, there are grounds for saying that better use of the design is essential to individuals wrestling with system complexity. Firstly, the design is a large part of what the customer is paying for and getting it right is of some importance. Secondly, reviewing the design is where trade-offs between viewpoints find their final resolution. The resolution of requirements trade-offs is a separate issue to the resolution of design trade-offs.
Process modelling and assessment: One of the distinctions between process models and methodologies is the ability to make assessments of an organisation. The five parts of ISO 15504 include resources to support Process Improvement (PI) and Process Capability Determination (PCD) (also known as Capability Evaluation) (the most relevant parts are cited). PCD is taken as being a valid measure of an organisation, and, used appropriately, provides an indicator of the organisation's ability to meet the demands of complexity, including diversity.

ISO/IEC 15288 is concerned with the life cycle of a system down to the level of element detail where individual implementation technology practices can be applied – technologies such as software, electronics, biology, hydraulics, mechanics and a host of other science and technology disciplines, including human sciences. The following aspects of system engineering are required if the diverse roles are to communicate effectively:

- A common view of the life cycle (though each implementation or speciality will have its own variations).
- An ability to speak at a project level; this appears to require a process view. Specialist tools and methods are needed by the specialist, but cannot be understood at a project level. Overlays to ISO 15288 (such as ISO PAS 18152, SSE-CMM, ISO 12207) are required.
- A working set of 'project processes' from ISO/IEC 15288.

Formal management of diversity by Systems Engineering
Genius is not the solution: The role of individual competence and professionalism is not to be under-estimated. At the level of software teams," "Good tools typically increase productivity by 25%-35% and good processes can increase productivity by 50%-100%; but the best people are typically 10-20 times more productive than the average, and as much as 100-200 times better than the worst." (Yourdon, 1995).

However, it is contended that at the level of project processes, solutions that rely on very high levels of system architect professionalism (e.g. RAE/BCS, 2004) will not work for systems of even moderate complexity. "System design in its simplest embodiment is dependent on a convergence of specialized information from multiple technology domains which contribute integrally to either the function and embodiment of the system or to the processes needed to manufacture it. It is unlikely that a single individual today can be a master integrater of many technologies, let alone competently keep abreast of the developments in a single field." (Boff, 1987). The role of the 'maestro' in setting culture and standards (Westrum, 1998) is recognised. However, if complex engineering is to continue to grow, then it cannot depend on the ready availability of such people.

Babel is not an option: "While precise communication among specialists within a given technology can be a tedious process, it pales by comparison with the difficulties involved in attempting precise communication among specialists from across multiple disciplines. Specialized technical domains typically evolve specialized languages in order to communicate ideas and data unambiguously within a given frame of reference among specialists. Hence, in order for specialists to communicate effectively across domains, they must find a common frame of reference. The larger the number of individuals involved or the greater the differences in domains of expertise, the more difficult it will be to develop a common frame of reference and the more likely the potential for misunderstanding in communications." (Boff, 1987). Figure 2 illustrates the difference between specialised tools and methods, which can be understood properly only by the relevant role, and the overlay process models that mediate between these and project level system engineering processes.

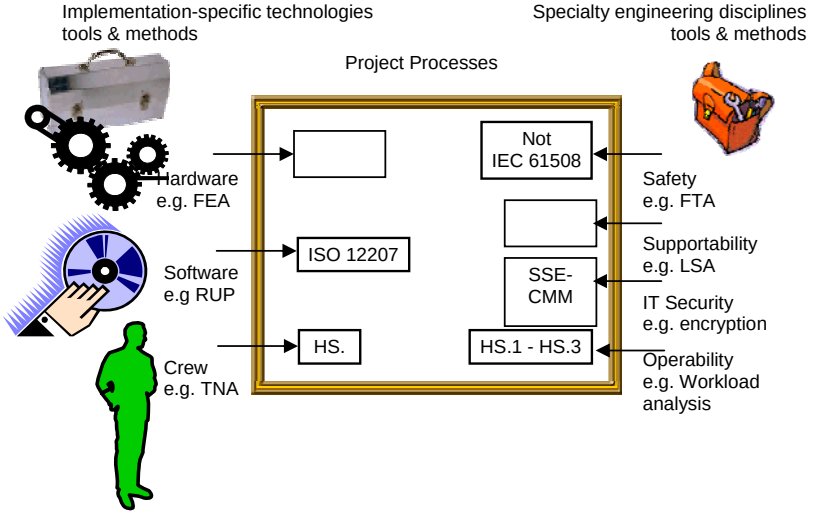


Figure 2 - Overlay process models mediate between specialists and the project to provide a common picture. (The HS.1 – HS.4 refer to processes in ISO PAS 18152)

From the viewpoint of operability (or Quality In Use[2]), the author's experience is that the specialist community (variously terms Human Factors, HFI, HSI, HCI, usability) has split in response to the proposal for overlay models and the prospect of Capability Evaluation. There is a group of practitioners that recognises the potential very enthusiastically, and there are other practitioners and the bulk of researchers who do not see the relevance. Splits of this sort are likely in most such communities.

Project processes: The Project Processes in ISO/IEC 15288:2002 are concerned with managing the resources and assets allocated by enterprise management and with applying them to fulfil the agreements that the organization enters into. They relate to the management of projects, in particular to planning in

[2] Defined in ISO 9126:2000 as "The the capability of a system to enable specified users to achieve specified goals with effectiveness, productivity, safety and satisfaction in specified contexts of use."

terms of cost, timescales and achievements, to the checking of actions to ensure that they comply with plans and performance criteria, and to the identification and selection of corrective actions that recover progress and achievement. The number of technical viewpoints to be managed has a direct impact on the ease of achieving project process outcomes. The view in the systems engineering community is that the way to meet the demands of greater diversity is to increase process capability levels for these processes.

Matching challenge and capability: There are two points to make about the extent to which project processes need to be performed in relation to system complexity. Firstly, there has been some work (Benediktsson et al) relating process maturity to Safety Integrity Level (SIL) requirements i.e. there is a link between the performance demands of the system of interest and the process capability requirements. Secondly, there ought to be a more general link between the complexity-as-diversity demands of a project and the process capability of the enterprise. Developing metrics to measure the diversity demands would not be particularly difficult and could be aligned to the PCD of the organisation.

Informal management of diversity; culture and teambuilding

The use of process models provides a way of addressing the formal organisational aspects of diversity. The informal organisation needs addressing as well.

Social practicalities: The Australian monthly barbeque, or a night out for a curry in the UK, seems to be a powerful way of getting people to talk to each other and build a common picture of the project. The demise of stratified canteens in UK engineering removed lunch as a means of achieving this. Social engineering can be cheap and effective. The more complex the project, the larger the restaurant bill. The number of interactions between viewpoints (a good indicator of the complexity of the social system and the demands of diversity) rises factorially rather than linearly with the bill. The author was part of a Prime Contract Office that was assembled and chastised by the project manager "I want to see more people sitting on desks and chatting".

Culture and dealing with bad news: Or, "How do you tell a mother that her baby is ugly?" The various viewpoints inevitably have to be the bearers of bad news. The treatment of bad news is a feature of Westrum's (1998) climates for information flow. The *pathological* organisation tends to suppress or encapsulate anomalies. *Bureaucratic* organisations can ignore or make light of problems and not seek problems out. The *generative* organisation encourages communication with a culture of conscious inquiry. The differences in their ability to manage diversity are apparent.

Team decision making and shared situation awareness: It is argued that a shared interactive virtual model of the system-of-interest is a major support to the development of a shared mental model, necessary for teamworking (Zsambok and Klien, 1997). So far as is known, this approach to the value of visualisation has still to be proven or properly researched.

Demonstration and argument: Given the problems of jargon referred to above, the explanation of specialist bad news can be a challenge to the role responsible, and can lead to the reversal from project scrutiny to specialist scrutiny e.g. as occurred with the Challenger Shuttle 'O' rings. The ability to use an interactive visualisation to demonstrate an issue enables non-technical communication to be made 'on the fly' rather than have technical communication (probably subsequent to the review or meeting) suppressed or ignored.

Organisational impact of visualisation: Experience at the Digital Design Studio (DDS) with the application of advanced visualisation to car design has been that the effect has been equivalent to reducing the complexity of the artefact. It would appear that the reduction in individual cognitive demands (discussed below) enables the project to be run as though it were simpler.

Complexity and individual cognition

Complex systems pose difficulties to those individuals representing one or more viewpoints. The mountain of documentation goes beyond the point where it can be considered readable (the author has worked on a project with well over 1000 databases). A specialist saying "it's in our database" is quite inadequate. Communicating specialist information is a major undertaking. There is thus a problem in identifying areas with potentially conflicting (or synergistic) requirements and constraints. With current project arrangements, potential conflicts between viewpoints are found early by social networking or by lucky searches through databases. By the time a conflict has been found in the model of the built artefact, potential re-work and conflict resolution has already built up (e.g Whyte 2002, p 63, 'Identifying errors and clashes'). It is proposed that better visualisation will allow group processes to operate in a way that gives more lead times on such conflicts.

The experienced designer: The expertise of the experienced designer needs to be recognised. Boff (1987) has characterised the expertise as shown in Figure 3. There have been many attempts in the expert systems era to capture this expertise within say a CAD system. These have been generally unsuccessful and a more promising approach would be to support this expertise and encourage its development.

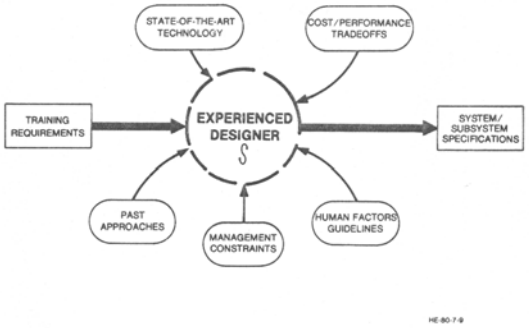


Figure 3 - Design Decision Process. Design requirements and specifications are determined by the subjective integration of a range of variables.

Naturalistic Decision Making: The proven way to capture expertise from experienced individuals and teams is the use of theory and practice from Naturalistic Decision Making (e.g. Zsambok and Klein, 1997). Resources have been developed to assist learning, training and decision support. Although extensively applied to the users of complex systems, there has been virtually no application to the designers of such systems. A promising area of application is the expertise associated with experienced designers and a review of the built artefact (at whatever stage of design/manufacture). Experienced designers with an established viewpoint (or possibly more than one viewpoint) have well-developed sets of cues that can be used to identify successful or problematic designs. One of the authors had the rewarding experience of developing synergy between the viewpoints of ease-of-use, ease-of-maintenance and ease-of-manufacture on a complex system. We argue that good interactive visualisation provides more and better cues to support such expertise. In particular it supports the development of synergistic what-if exploration.
"It's generally accepted that visualization is key to insight and understanding of complex data and models because it leverages the highest-bandwidth channel to the brain. What's less generally accepted, because there has been so much less experience with it, is that IVR [Immersive Virtual Reality] can significantly improve our visualization abilities over what can be done with ordinary desktop computing. IVR isn't "just better" 3D graphics, any more than 3D is just better 2D graphics. Rather, IVR can let us "see" (that is, form a conception of and understand) things we could not see with desktop 3D graphics." (van Dam et al 2000).

Current understanding of the value of Virtual Reality (VR) or visualisation: The short answer is that we know it works, but have not proved it yet. Further, the technical difficulties in achieving fluid multi-sensory interaction with large datasets are still being overcome, and so criteria for 'good enough' are not developed. van Dam et al (2002) have identified that fundamental research is still needed in this key area: "A new and highly interdisciplinary design discipline for creating compelling immersive environments, including visualizations. We cannot expect interdisciplinary design teams to evolve spontaneously or to invent their techniques from scratch each time. And environment design certainlycannot be left to traditionally-educated computer scientists, most of whom have no formal background in the creative arts and design or in human perceptual, cognitive and social systems."

Conclusions
The successful management of diversity offers the potential for huge gains in system outturn. This requires a combination of system engineering processes, a benign organisational climate and resources to support experienced designers. Achieving a common view by means of system engineering processes is not simple; Process Improvement is a hard road. Overlay models are still needed, and their widespread acceptance and adoption is some way off. The ability to perform Process Capability Determination offers a real incentive to improve and the potential to change the marketplace. Metrics for assessing the challenge

posed by the diversity of a project would be of considerable assistance to project management, planning and contract award.
Interactive visualisation offers enormous potential. Whilst there are formidable technical challenges and still some key research issues, it is entirely complementary to Process Improvement. For a number of organisations, it offers the possibility of a quick win during the difficult stages of improving project processes.

Acknowledgements
The authors would like to thank Stuart Arnold (Qinetiq), Jonathan Earthy (Lloyd's Register), Tim Brady (CoPS) and Jennifer Whyte (Imperial College) for their discussions and ideas, and two anonymous reviewers for their constructive comments.

References

Arnold, S., Sherwood Jones, B.M, Earthy J.V. (2002) 'Addressing the People Problem - ISO/IEC 15288 and the Human-System Life Cycle' 12th Annual International INCOSE Symposium, Las Vegas
Benediktsson O, Hunter R B, McGettrick A D. (2001) Processes for Software in Safety Critical Systems in Software Process: Improvement and Practice, 6 (1): 47-62, John Wiley and Sons Ltd.
Boff, K.R. (1987) The Tower of Babel Revisited: On Cross-disciplinary Chokepoints in System Design. in Rouse, W.B. & Boff, K.R. (Eds.) Systems design: Behavioral perspectives on designers, tools and organizations. New York: Elsevier.
Bullock, S., Cliff, D. (2004) 'Complexity and emergent behaviour in ICT Systems' DTI Foresight Report
Capers Jones, T. (1996) 'Applied Software Measurements: Assuring Productivity and Quality'
Elster, J. (1979) 'Ulysses and the Sirens' CUP
Fry, J. (2001) 'Software Estimation' talk given to the Scottish Software Process Improvement Network, Glasgow.
Gause, D.C., Weinberg, G.M. (1990) 'Exploring Requirements: Quality Before Design' Dorset House Publishing Co
Hitchins, D.K (2001) 'Putting Systems to Work'
ISO/IEC 15288:2002 Systems engineering -- System life cycle processes
ISO/IEC 15504-2:2003 Information technology -- Process assessment -- Part 2: Performing an assessment
ISO/IEC 15504-3:2004 Information technology -- Process assessment -- Part 3: Guidance on performing an assessment
ISO/IEC 15504-4:2004 Information technology -- Process assessment -- Part 4: Guidance on use for process improvement and process capability determination
ISO 9126:2000 Software product quality - quality model
ISO/PAS 18152:2003 Ergonomics of human-system interaction -- Specification for the process assessment of human-system issues
Meyer, M.H., Lehnerd, A.P. (1997) The Power of Product Platforms: Building Value And Cost Leadership, Free Press.
Royal Academy of Engineering, British Computer Society (RAE/BCS) (2004), "The Challenges of Complex IT Projects" ISBN 1-903496-15-2
Westrum, R.(1998) 'Organizational Factors Associated with Safety and Mission Success in AviationEnvironments' in the 'Handbook of Aviation Human Factors', edited by Garland, D.J., Wise, J.A. Hopkin, D.V., Lawrence Erlbaum Associates Inc.
van Dam, A., Forsberg, A.S., Laidlaw, D.H., LaViola Jr., J.J., Simpson, R.M. (2000), Immersive VR for Scientific Visualization: A Progress Report IEEE Computer Graphics and Applications No/Dec 2000, pp 26 et seq.
van Dam, A., Laidlaw, D.H. Simpson, R.M. (2002) Experiments in Immersive Virtual Reality for Scientific Visualization Computers & Graphics 26 535–555
Whyte, J. (2002) Virtual Reality and the Built Environment. Oxford, Architectural Press.
Williams, J. (2001) First Scottish Systems Engineering Convention, INCOSE UK
Yourdon, E. (1995) "The Yourdon Analysis" August 1995
Zsambok, C. E., & Klein, G. (Eds.). (1997). Naturalistic decision making. Mahwah, NJ: Lawrence Erlbaum Associates.

Design of the ICT infrastructure of an educational system

Pedro Baquero*, Rosa María Aguilar, Alejandro Ayala

Dept. of Fundamental and Experimental Physics, Electronic and Systems. University of La Laguna (Spain)
*E-mail address: pbaquero@ull.es; http://webpages.ull.es/users/pbaquero/

Abstract: In this paper we expose the design process in engineering an information and communication (ICT) infrastructure. We have based on the Viable System Model for decomposing the ICT infrastructure. A different framework for managing complexity has been applied to each subsystem and a special treatment for the design of the ICT Center has been carried out. ICT Center has the responsibility of guaranteeing the quality of service of the whole system.

Keywords: ICT infrastructure, Viable System Model, Quality of Service, Microsimulation.

Introduction

The information and communication technologies (ICT) infrastructure is a key element in many organizations. This infrastructure is composed by a set of hardware, software, services, procedures, processes and persons. Our vision considers ICT Infrastructure as an organization, with a great number of elements, with persons that interact with these elements and with other persons, with complex processes, with a great number of procedures, etc. This infrastructure should interact with its environment, should adapt to it and should evolve. With this vision we have consider the ICT infrastructure as a complex system. In our work we have developed a methodological framework to model and to design this ICT infrastructure concept.

This work has been applied to a concrete case: the ICT Infrastructure of the Canary educational system. The result has been a basic technological architecture with three evolutionary projects: the Individualize Networks of Schools, the Integrated Broadband Network and the Management System. From a common strategy, each one of these projects has their own evolutionary strategy. This basic technological architectural has been designed considering that at the same time we are planning, designing, building, using and operating. This study has been framed inside MEDUSA and MEDUSA PRIMARIA projects. These projects are an initiative of Government of Canary Islands (Spain) to extend the ICT use in the educational system. The complexity of this ICT Infrastructure is increased since only a little number of persons is the responsible of managing the design, the planning, the acquisition, the installation and the administration. In this sense a framework to reduce its complexity has been design.

Our framework consists of the decomposition this ICT infrastructure in different subsystems. This decomposition has been based on the Viable System Model (VSM). VSM permits to manage the decomposed parts in an integrated way. Each subsystem can be considered as a system with less complexity than the whole system. For each subsystem has been designed an appropriate framework that permits to manage its particular complexity. These frameworks use specific methodological techniques, also simulation techniques based on microsimulation has been applied to a specific system. The framework of this work can be generalized to the design of other ICT infrastructures.

As any system, ICT Infrastructure can fail or can not operate correctly. Thus, a specific treatment has been realized to design this system with a controlled quality of service (QoS). For example, when an element fails it must be repaired, or when a teacher needs a specific service, it should be provided. If the repair or provision time is high, QoS is low. More and more the organizations demand bigger QoS for their ICT Infrastructure. Traditionally, the assessment of fulfilling a certain QoS is carried out based on the designer's experience. We have designed a framework to predict QoS of an ICT Infrastructure. In our model there is a subsystem that is responsible of guaranteeing QoS. This subsystem is the most complex of the whole system. For this system a framework based on microsimulation techniques has been used.

Firstly, summary of our case study is made. Secondly, we develop the concept of ICT Infrastructure and how its complexity is managed. Thirdly, we expose the design of the ICT Center. ICT Center is the

responsible system of guaranteeing QoS. In that section a framework based on microsimulation has been described. Finally conclusions are exposed.

Case study

Canary Islands is a region of Spain formed by seven islands. It is located in the Atlantic Ocean, 1,250 km from Europe and 210 km from the African coast. It has a population of 1,694,477 inhabitants, being 40% of them concentrated on the metropolitan zones of the two main islands. The territory is strongly fragmented due to its insularity and orography. The non-university educational system is formed by 850 administrative workers, 301,622 pupils, 19,660 teachers and 1,264 schools. At the beginning of year 2001 the penetration was 23.6 students per computer with an important number of obsolete computers. On the other hand, the available computer material in the schools was used to cover the derived necessities of the ICT conception as a curricular subject, not being approached the ICT use as a didactic instrument in the different areas and curricular subjects. Neither the schools had an infrastructure of data cabling that facilitates the installation of ICT equipment. In year 2000 the number of schools with Internet access was about 24%, and only 2.2% had a WEB site. On the other hand, while there were some teachers with positive ICT attitudes, there were other teachers who had resistances of different intensity toward the ICT use.

The Canary Islands educational system is framed inside Spanish educational system characterized by a decentralized model of administration that distributes the responsibilities mainly among National Government, Regional Governments and the educational centers. National Government has reserved exclusively the exercise of the responsibilities that safeguard the homogeneity and the substantial unit of the educational system. Regional Government deals with, among other duties, the administrative ownership in their territory, administration of personal, orientation and attention to the pupil, helps and grants, etc. Regulations have been establishing the principle of autonomy of schools. They have capacity for the decision-making in curricular aspects. Educational centers or schools should elaborate three different documents where their pedagogic and curricular organization is reflected: the educational project, the curricular project and the didactic programming. It is therefore at the school where the initiative is focused on including ICT for support for the educational project. Regional Governments will be able to motivate and to support this type of initiatives.

At the present time, Government of the Canary Islands, through Department of Education, Culture and Sports, is developing a specific ICT project for non university education. It is basically conceived as an integral programme where all the educational elements are identified. This project is bounded to the educational administration and the public schools of Canary Islands. These are projects with very wide objectives. This project is developed in two Phases: Phase I (2001-2004) and Phase II (2004-2006). At the end of Phase II all schools will have Internet connection and WEB site and there will be a ratio of less than 12 pupils per computer. At the present time this project is at the beginning of Phase II and it is at a level which allows its evaluation.

The ultimate and general aim of these projects is to integrate ICT in educational non university environments in the Canaries in an effective way. This integration should lead us to qualified teachers and students in a short/middle-term period of time, so that they are used to logical and critical use of tools and technological resources, and that will permit new ways of teaching and learning, and that will also help to establish new ways of communication and contribution with other educational agents.

Projects of this dimension require a set of actions that become the basic pillars for a correct execution of it, and these actions are carried out in a coordinated and complementary way. The first action initiated is the creation of infrastructures and equipment in the whole educational system.

The connectivity between schools equipments to each other is possible thanks to the Local Network that is created in each school. Each Local Network contains all the equipment and network electronic elements that permit to share resources, applications, contents and forums of communication. At the same time, they facilitate the exchange with the outside in a safe way, from each endowed area. Each Local Network consists of connecting points distributed all over the school (management department, computer science

classrooms, “Classrooms Medusa”, "classroom corners", library, departments, “Special Education” classrooms and laboratories).
The local area networks as a whole make up the second level of network (Educational Intranet), with the same philosophy as the local network. Servers and specific tools permit the Network management, sharing resources, applications, etc., and they are also improved with new functionality such as the distribution of applications, backups, virus shields, among others. The Intranet configuration enables the maintenance of interconnected machines through the corporate network of the Government of the Canaries.
The second basic pillar consists of training teachers, students and other agents involved in the execution of the Project. Users training is conceived as functional, practical and adapted to the contexts, to the materials and environment in which the Project develops. Training contents and offers are flexible. They are collected in an annual Training Plan, provided with a modular structure to facilitate teachers the make-up of their training itinerary. This Plan is continually updated and improved.
The provision of contents is another strategic focal point of the Project. The shortage of educational and specific contents related to ITC, or borne by them, has not favored the approach to ICT in schools, as well as integration and use of ICT as instrumental support in the different subjects. The policy of contents provision is undertaken in different ways. In this sense, the promotion and support to innovation and educational research projects will be another source of provision, with the added value that these materials are already contextualized in specific classroom situations, so that the level of motivation is very high, because they will be suggested by teachers that work with them.
The strategic bases for the design of the ICT infrastructure have been that the basic necessities of the educational community and the deployment of ICT infrastructure should be synchronized with the objectives for transforming the educational system. Thereby, in our vision of the global design, the different parts of ICT infrastructure have been integrated together with the processes that eliminate the obstacles for the ICT integration in education. As reported in Pelgrum (2001) the top 10 obstacles consisted of a mixture of material and non material conditions. The material conditions were the insufficient number of computer equipments and of Internet accesses. As non material conditions were the lack of skill of teachers and the lack of supervisory and technical staff. While the material conditions are fulfilled with the deployment of equipment and Internet accesses, the non-material conditions require the “deployment” of human resources, services and management processes. The result has been an ICT educational, human and technological architecture that we have denominated ICT infrastructure.
This ICT infrastructure is composed (approximately) by 25.000 PCs, 1.200 servers, 3.000 peripherals, 4.000 switches, 3.000 access points (WI-FI), 1.200 routers, 50.000 network points, 1.000 cabling infrastructures, 1.100 ADSL lines, 100 satellite lines, central services (as software deployment, security copies, monitoring services, etc.), corporate applications, 1.200 cooperative environments in each school, an ICT coordinator in each school, etc. In this ICT infrastructure the human infrastructure play a very important role and is composed approximately by 20 technical operators, 20 software developers and a technical office with 8 persons. Also, we can consider that the 20.000 teachers are inside of this ICT infrastructure when they are considered as an element to fulfill the educational activities.
This basic technological architectural has been designed considering that at the same time we are planning, designing, building, using, operating and redesigning, and only five persons are responsible to manage this ICT infrastructure.
The requirements of this ICT infrastructure are basically: it should have a good quality of service (QoS), should be economically efficient and should be changing and evolutionary. QoS should guarantee a good service to the users: when the ICT infrastructure fails the service should be repaired or when a user needs a service it should be provided.
The process of optimal design for guaranteeing a specific QoS in a changing environment is not direct from a deterministic viewpoint. QoS depends of the quality of ICT equipment, a good installation and a correct assessment of the human infrastructure.

Concept of ICT infrastructure

Our concept of the ICT infrastructure is not only a set of equipment or elements. The ICT infrastructure enables to share the ICT capabilities which provide services for other systems of the organization (Broadbenta et al, 1999). For Broadbenta et al these capabilities require the complex combination of the technical infrastructure (cabling infrastructure, hardware platform, base software platform), ICT shared services (as communications services), ICT applications (as WEB services), the human operators and the managerial expertise to guarantee reliable services (see figure 1). All these resources are designed,

developed and managed over time. In our system ICT infrastructure does not include the specific computer applications, but the teachers or other users should experience and innovate using specific computer applications on the ICT infrastructure.

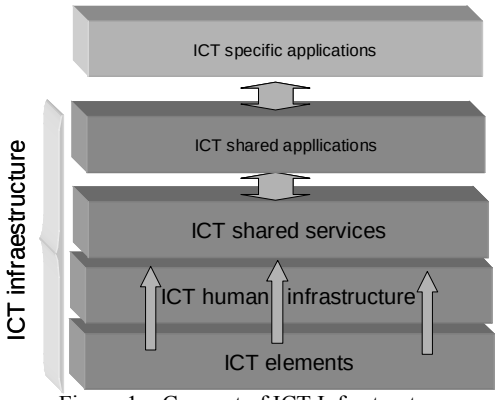


Figure 1 – Concept of ICT Infrastructure

The ICT infrastructure must be flexible to support the operation among different applications and to facilitate the communication of the information inside and outside of the enterprise. Thus, it must be flexible and integrated:

- Building a flexibility infrastructure implies cost and complexity because that supposes to add a characteristic that may be exercised in the future, and must consider the variety of user necessities that an organization can handle without modifying the infrastructure significantly. An organization can take different approaches to invest in the ICT infrastructure investments. ICT infrastructure needs to respond more rapidly to changes in the environment and between the different functional units.
- The integration increases the importance of relations among services and information. This integration implies the capability of exploiting resources shared across services, locations and departments. In this way, an ICT infrastructure must be unique and shared rather than separate ICT platforms.

Other aspects that permit to have a flexible and integrated infrastructure include the knowledge, the skills and the experience embodied in the human infrastructure.

This conception of ICT infrastructure in a large organization can be considered a complex dynamic system (variable environment, organizational system, a great number of elements, etc) in which deterministic and mathematical rules representing all the details of the models can not be easily formulated.

Methodologically, the first step in our analysis of this complex system has been to use the method of decomposing the whole system in subsystems with a smaller complexity degree. In our framework each subsystem can be treated independently, although in each subsystem the whole integration and the synergy with the other subsystems have been considered and one subsystem is the responsible of the integration of all the parts. The decomposition of this system has been based on the Viable System Model (VSM) (Beer, 1984). The Viable System Model considers an organization interacting with its environment. Beer pointed out that a system is only viable if it has a specific management structure. According to the proposed VSM a set of management tasks is distributed to five systems ensure the viability of any social system. The five systems are Operation, Coordination, Integration, Intelligence and Policy. These five systems can be summarized as follows: Operation realizes the primary activities; Coordination regulates and coordinates the different subsystems of Operation; Integration is the controlling unit of the operational level (Operation, Coordination and Integration). It has to ensure the operation of all the system and to optimize the allocation of resources; Intelligence is the link between the primary activities and its environment. On this level the future developments according the systems capabilities and changing of the environment (customer demands) are planned; and Policy is the main decision level of the whole system. The main roles of Policy are to provide clarity about the overall direction, values and purpose of the organizational unit.

Figure 2 shows the VSM model where an organization is composed by two elements: Operation which does all the basic work and Metasystem which provide services to Operation by ensuring the whole organization works in an integrated and viable way. Operation contains the productive units (in our case,

schools). Metasystem procures cohesion, optimization, synergy, cohesion, stability and future planning to ensure adaptation to a changing environment. Both Operation and Metasystem must be in contact with, and interacting with, their environment. In our case the set of students (and their parents) is the environment.

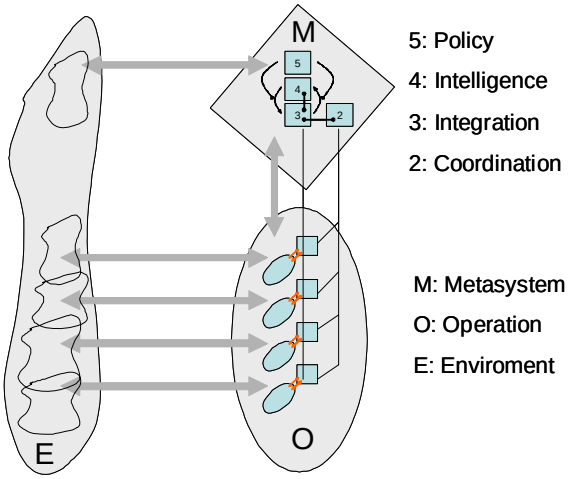


Figure 2 – Viable System Model

We have considered our ICT infrastructure as an organization. Thus, VSM model has been applied to the ICT Infrastructure and, moreover, Operation, Coordination, Integration, Intelligence and Policy have been identified. ICT infrastructure as a viable system is shown in figure 3. In our model each subsystem has a specific framework.

We have decomposed Operation in other VSM systems: the corporate applications and all the schools (ICT infrastructure of schools are considered as a VSM system). The number and the situation of all ICT elements can be seen in figure 3. To reduce the complexity of Operation a uniform solution has been designed for all the schools. Thus, important scale economies will take place with the centralized acquisition of ICT equipment. The uniformity of equipment has important economical profits, also, facilities and simplifies the operation, administration and centralized maintenance. Also, the selection of a uniform technological solution is the only viable way of executing a project that embraces all the great quantity of centers from a unique technical project office. In Operation teacher play two roles: one role as human infrastructure and the other one as environment. The complexity due to the great number of teachers has been managed identifying three types of teachers in function of their ICT skills. Each school ICT Infrastructure has synergies with the whole system, Intelligence subsystem is supported by Intelligence of the whole system (for example, backup copies are realized centrally). Thus, technical staff and technical knowledge are not necessary in schools. There exists an ICT coordinator in each school that is a teacher (without high technical knowledge) dedicated to promote ICT use.

On the other hand, the complexity of Metasystem is completely different to Operation. While in Operation ICT elements are the basic elements, in Metasystem the human infrastructure plays an important role. The number of persons in each subsystem is shown in figure 3. This system should fulfill the functions of the Metasystem: integration, monitoring, control, optimize and resolve conflicts inside the whole organization and realizes future planning to ensure adaptation to a changing environment. Thus, we have decomposed the Metasystem in four systems (based on VSM): Coordination (including the communication network), Integration (software developing, infrastructure deployment, educational support and the ICT center) and Intelligent and Policy (as management system):

- Coordination guarantees a correct interaction of all the parts of the system and with other systems (i.e, Canary Government Network). Its complexity is managed with the help of outsourcing and choosing few types of broadband accesses. Also Coordination guarantees the security of the system, avoiding that an element can damage all or a part of the system. This function is realized with security policies and specific security systems. This subsystem avoids that an element, teacher or student can suppose a risk for the whole system.

- Integration permits the adaptation of all the system and guarantees the continuity of service. Also it guarantees QoS (quality of service) of the whole system. Its complexity can be reduced decomposing it in four subsystems. The integration and the synergy of these subsystems are mainly realized by Intelligence. The four subsystems are Infrastructure Deployment Office, Software Development Office, Educational Support Office and ICT Center. Infrastructure Deployment Office is the responsible of the infrastructure endowment of the whole system. This office designs specifics solutions and manages and controls to the suppliers. Software Development Office is the responsible of the continuous software development of the corporate and cooperative environments. Educational Support Office is the responsible of training teachers and promoting the use of the ICT infrastructure. And ICT Center is the responsible of guaranteeing the correct operation of the ICT Infrastructure.
- The management system (Intelligence and Policy) guarantees a unique vision of the project, following the VSM model, the management system carries out a centralized control. Also it is the responsible of assessing the whole system.

With the exception of ICT Center, all the subsystems can be designed using deterministic approaches. For example, the human endowment of Infrastructure Deployment Office can be easily estimated knowing the average time to visit a school, to project an ICT Infrastructure of a school, to check it, etc.

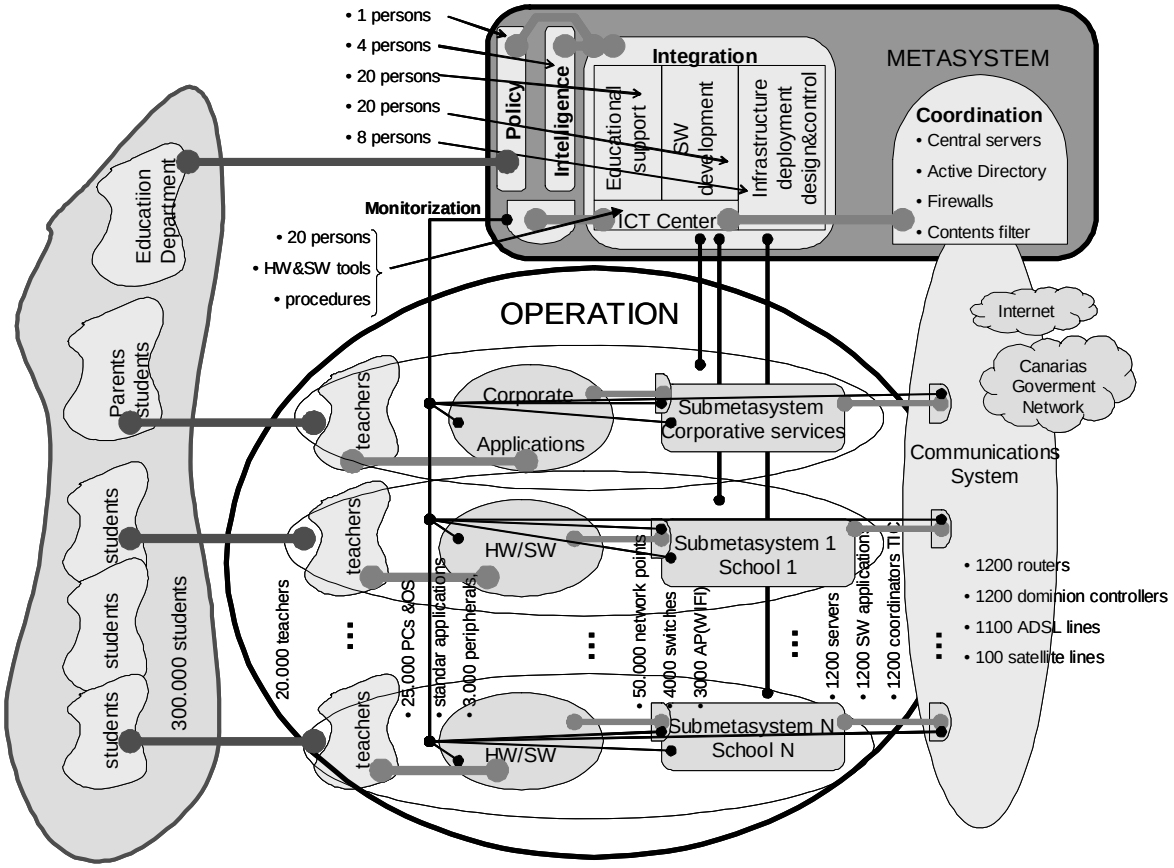


Figure 3 – ICT Infrastructure as Viable System Model

Design of the ICT Center

In this work we have focused our analysis on a specific subsystem: the ICT center. This subsystem is the responsible of guaranteeing QoS of the whole system and plays an important role to permit the integration of all the parts. ICT Centre administrates the whole ICT infrastructure, as well as the attention and service

to all users. While the other subsystems can be easily designed using project programming techniques, the design of ICT Center should use other types of approaches since it can be considered as a complex system where deterministic techniques cannot be used. ICT Centre should be endowed with an organizational structure and technological tools to carry out the centralized administration of ICT resources. All kind of inquires, problems, petitions or mishaps are managed until their complete resolution. And it is constituted as a unique point of direct attention. ICT Centre is organized in a similar outline as recommended by the methodology ITIL. ITIL (IT Infrastructure Library) is a broadly grateful methodology and adopted in the sector of the IT and that it supports and it reinforces standard of quality like the PD0005 (British Standards Institution's Code of Practice for IT Service Management) and the ISO9000.
Figure 4 shows an outline of the ICT Center. ICT Center is composed by tools that permit to monitor the elements, the network, the computer systems, to make an inventory of all the hardware and software, to provide services (SW deployment, remote control of computers and servers, realizing backups, etc), and to administrate the network, the active directory, the routers, the switches, etc. Also a human infrastructure that interacts with these tools is necessary. The human infrastructure is composed by Level 1 and Level 2. Level 1 is composed by phone operators with little ICT knowledge and their cost is low. Level 2 is composed by engineers and their cost is very high. Level 2 can be divided in specialized teams. Also, there is a Level 3 that is composed by the maintenance service of the suppliers.

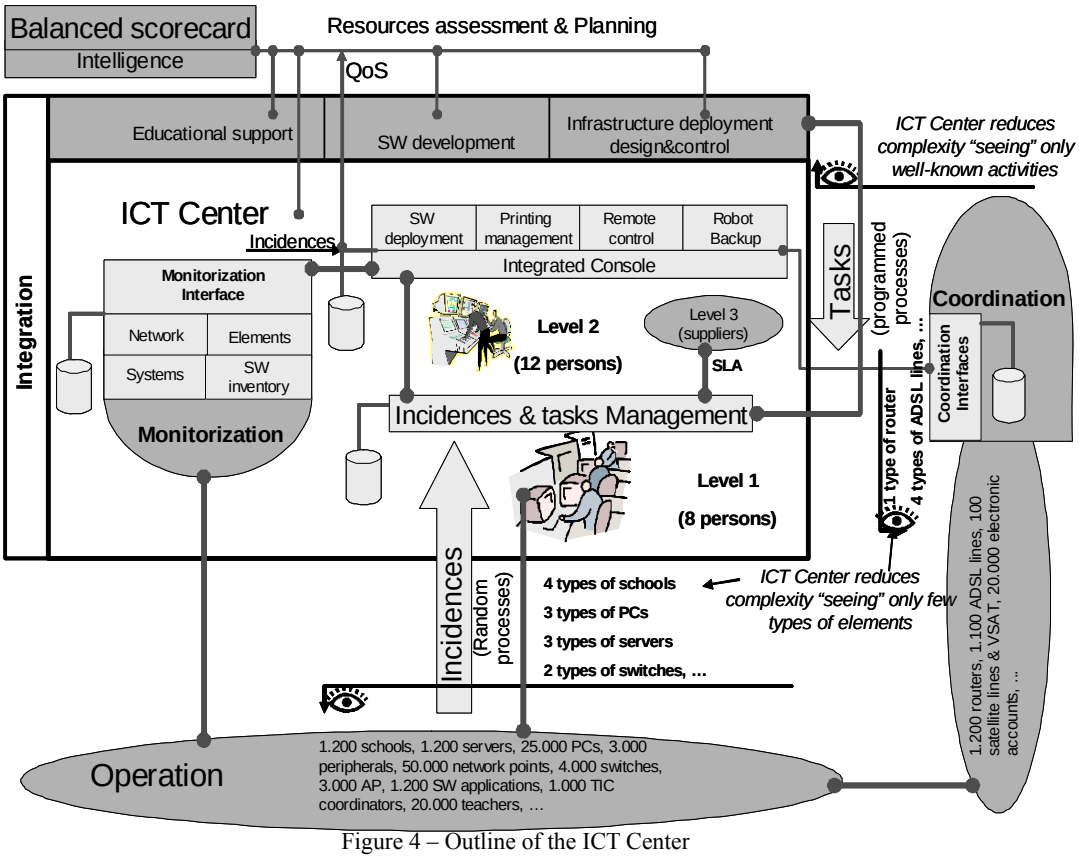


Figure 4 – Outline of the ICT Center

When an element or system fails or does not function, ICT Center is the responsible of its repair or reconfiguration. When a teacher needs a specific service (for example, a new electronic account), ICT Centre is the responsible of providing it. These fails, problems, petitions or mishaps are denominated *incidences*. The incidences can be reported by teachers or by the monitoring system. Incidences are mainly generated in Operation and Coordination System where there are a great number of elements. As shown in

figure 4, ICT Center only "sees" few types of elements, thus ICT Center can reduce the complexity of its operation. The incidence rate is a random process.
Also ICT Center is the responsible of the operation of the systems, for example, realizing backups, configuring elements and systems, etc. Also, ICT Center should carry out different activities requested by other subsystems (Infrastructure Deployment Office, SW Development Office, etc). These activities are denominated *tasks*. A task can be accepted if it has been previously defined: ICT Center only "sees" well-known activities. This way also contributes to reduce the complexity of the operation of the ICT Center. The task rate can be considered as programmed processes.
The requirements of the design of this system must guarantee a good quality of service (QoS). QoS depends of the assessment of human resources. This assessment must consider both the number of persons in each level (Level 1 and Level 2) and their knowledge. The assessment of these resources can not be realized using deterministic approaches due to: ICT Center can be considered as a human organization, the duration of each activity realized by the human infrastructure is random, incidences rate is random, each incidence has a different treatment, there is an important number of type of incidences and activities, etc. Thus, ICT Center can be considered a complex system.
With the purpose of simulation and design, ICT Center has been decomposed in other subsystems. In figure 5 shows the ICT center decomposed in four subsystems: subsystem 2 (where its Level 2 operates the technological tools), subsystem 3 (composed by the technical staff), subsystem 1 (that manages incidences) and subsystem 4 (Intelligence) that controls all this system:

- Subsystem 3 does not process incidences or tasks. It is a static system where the number of elements can change and the knowledge of each element varies with the time. Subsystem 3 feeds with human resources to subsystems 2 and 1. These human resources are shared by these two subsystems.
- Subsystem 2 processes programmed tasks, thus the occupation of the human resources (Level 2) can be easily predicted. The output of subsystem 2 is the operation state of the systems.
- Subsystem 1 processes random incidences. The occupation of its resources cannot be easily predicted. The output of subsystem 1 is QoS of the whole. QoS is mainly measured by the resolution times of the incidences.
- Subsystem 4 indicates the assessment of human resources that subsystem 3 should have.

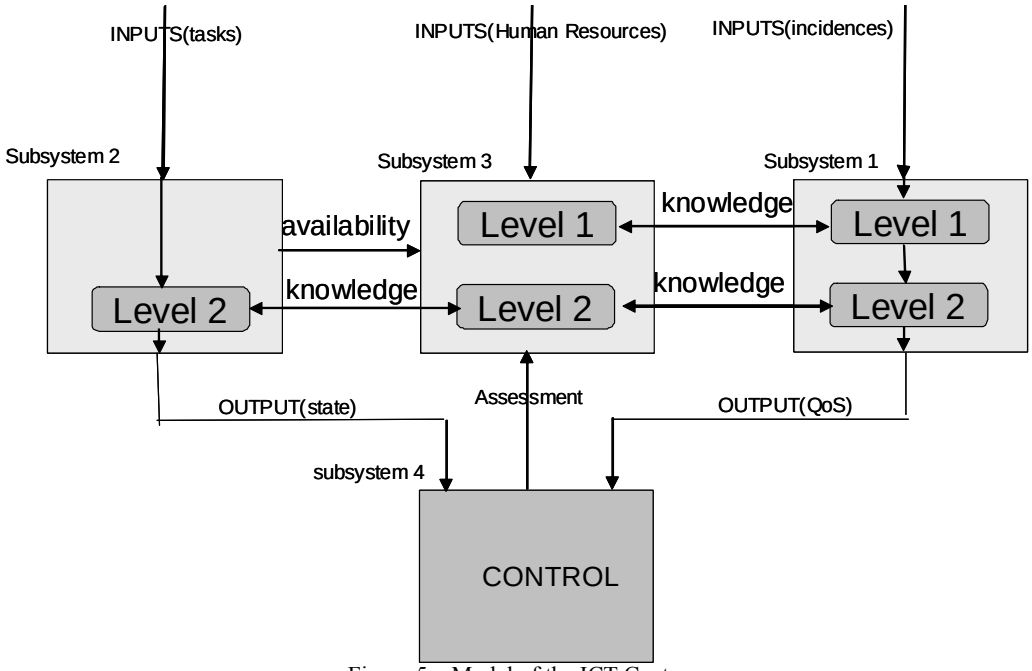


Figure 5 – Model of the ICT Center

Despite the fact that subsystem 1 can be analyzed as a discrete events system that change their condition each time a new event happens (new users, new necessities, new skills, new services,…), complex organizations as this ICT center are modeled more efficiently if we consider the system components as entities that flow throw them (process oriented simulation). A process is an ordered time sequence of interrelated events. This sequence describes the pass of an item throw the system. Other typical feature of this kind of models is the appearing of the transaction concept. An actor of the system requires a particular resource and its action is determined by this fact. Transactions in process oriented simulations have to be defined as a sequence of steps. Studying transactions in complex organizations involves the concept of microsimulation: all the actors involved in relevant transactions are included in the model and they are simulated. These actors behave according to established rules that can be deterministic, stochastic or a mixture of both. In the framework of process oriented simulation each actor is a process and in this ICT Center we can have hundreds of processes competing for resources. This is the typical situation of incidences waiting for being resolved by this ICT center.

With this approach the resolution time of the each incidence can be predicted over the time, and thus can be assessed with the human resources to fulfill a certain QoS. In figure 6 shows as incidences are resolved. The resolution time has been simulated using an available tool that uses the previous concepts. Initially this tool was developed for the process simulation in a hospital. A discussion and justification of this tool can be found in Moreno et al (1999) and Moreno et al (2001). Other frameworks can be found in Unger & Cleary (1993) and Gilbert & Doran (1993). Adaptation of this tool for simulating ICT Center has been direct.

In this system, the concrete resource (human operator) that resolves incidences also fixes the quality of service, due to their experience the resolution is carried out in more or less time. An analytic approach of the problem would not allow us to discriminate the specific performance of each resource and each incidence, while we can carry out this analysis centered in the resource with a modeling oriented to the process. This is due to each element (resource and incidence) that flows for the system is simulated in an individualized way: we carry out a microsimulation.

Figure 6 shows the average resolution time of all the incidences. Other results could be represented easily (for example, average time of each incidence type). Also, with the simulations realized we could predict the occupation of the resources and thus their efficiency can be measured. One advantage to choose this approach is that with a process oriented modeling we focus on the local problems that are not only important in their own, but also because sometimes they are in the origin of emergent behaviors that affect the complex system as a whole (Harding, 1990). Also, this methodology allows us to know the influence of the peaks of incidences on the QoS of the whole system: a massive arrival of incidences can be simulated (for example, a widespread propagation of a virus can imply personalized actuation in each PC) and it can be analyzed the influence of the resolution time on the habitual incidences.

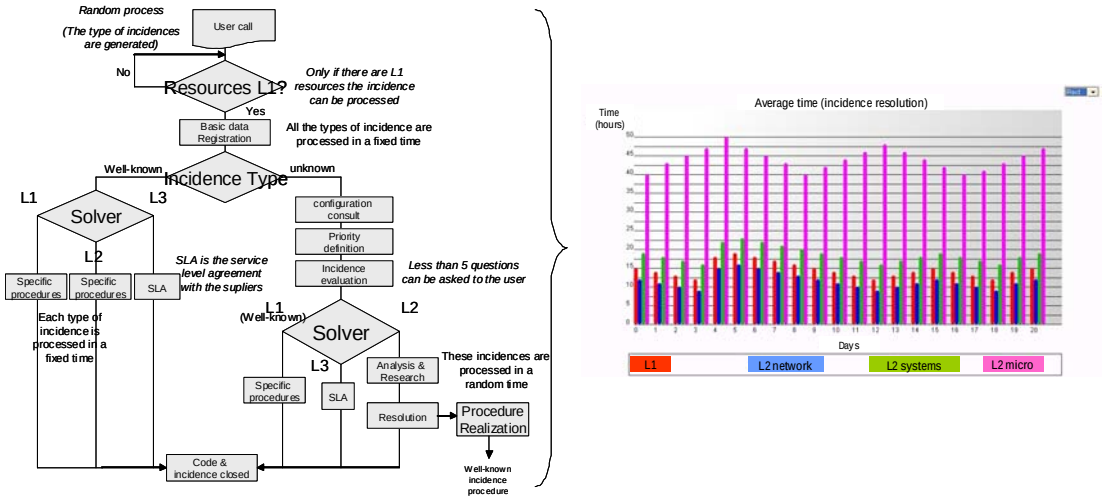


Figure 6 – Simulation of the resolution process of incidences

This ICT Center has been operating since two years ago (during Phase I), thus we have been able to use the previous experience to establish parameters in our simulation, for example, resolution time of well-known and unknown incidences and types of incidences. Due to that other parameters have not been measured, in our simulation we have realized some simplifications (human knowledge has been considered as constant, i.e., it does not change with the experience and we have considered that the well-known incidences and processes are processed in a fixed time).
In a real ICT Center there are other many QoS indicators (for example, waiting time of the user to be assisted). With the microsimulation it is possible to predict these types of indicators, although it is not possible to predict other indicators as the perceived quality for the user. These indicators are measured using surveys to the users.

Conclusions

We have seen as an ICT Infrastructure can be considered a complex system and it should work with a certain QoS. Also we have seen as applying a framework based on VSM facilitates the analysis and the design of a complex system as this ICT infrastructure, and that inside this system ICT Center is the responsible of ensuring that the whole system works correctly.
Unfortunately we have not found works that have approached neither the treatment of an ICT infrastructure like a complex system, nor the QoS prediction in an ICT Center that allow us to be carried out a comparative discussion with our framework.
Our focus based on VSM has allowed us to design an ICT Infrastructure solution that can be managed by few people. QoS is a key aspect of this system and a good design should consider it. Using microsimulation adapts quite well to model the QoS performance in an ICT Center since it is possible to predict easily different indicators that measure the QoS of the ICT Infrastructure. As inconveniences we have that it is not possible the prediction of other important indicators as the perceived quality of the service for the user. Another inconvenience is the difficulty to extrapolate the experience of a specific ICT Center to other one, since the QoS performance depends on many factors, like a correct installation of the administration tools, the knowledge of the users, the grade of stability of the ICT Infrastructure, etc. This implies that to carry out a better assessment it is necessary to have real data of the history of an ICT Center. For it, it becomes necessary a previous period of time operating before carrying out a final design of the necessary resources in an ICT Center. Once there is a previous experience it is possible to carry out predictions of the QoS performance simulating different situations and resources.
Finally, we indicate that the framework developed in this work can be applied to other ICT Infrastructures since all these have similar structures.

Acknowledgement

This work has been funded by B2Canarias Consulting Group SL.

References

Beer, S (1984). The viable system model: Its provenance, development, methodology and pathology. Journal of Operational Research Society, 35, 7-25.
Broadbenta, M, Weillb, P, Neoc, B.S. (1999). Strategic context and patterns of IT infrastructure capability. Journal of Strategic Information Systems 8, 157–187.
Gilbert G., Doran J., (1993). Simulating societies: the computer simulation of social processes, UCL Press.
Harding A., (1990). Dynamic microsimulation models: problems and prospects, London School of Economics,.
Moreno L., Aguilar R.M., Martín C.A., Piñeiro J.D., Sigut J.F., Estévez J.I., Sánchez J.L. (1999). Patient Centered Simulation to Aid Decision-Making in Hospital, Simulation, 7, 373-393.
Moreno L., Aguilar R.M., Piñeiro J.D., Estévez J.I., Sigut J.F., González C., (2001). Using KADS methodology in a simulation assisted knowledge based system: application to hospital management, Expert System with Applications, 20, 235-249.
Pelgrum, W. J. (2001). Obstacles to the integration of ICT in education: results from a worldwide educational assessment. Computers & Education, 37 (2), 163-178.
Unger, B, Cleary, C (1993) Practical Parallel Discrete Event Simulation, ORSA Journal on Computing, 3(5), 242-244.

Complexity of Design in Safety Critical Interactive Systems:

Gathering, Refining, Formalizing Multi-Type and Multi-Source Information while Ensuring Consistency, Reliability, Efficiency and Error-Tolerance

Sandra Basnyat, David Navarre, Philippe Palanque

(Basnyat, Navarre, Palanque)@irit.fr
LIIHS – IRIT, University Paul Sabatier, Toulouse, 31062, France
http://liihs.irit.fr

Abstract: The design of a usable, reliable and error-tolerant interactive safety-critical system is based on a mass of data of multiple natures from multiple domains. In this paper we discuss the complexity and dangers surrounding the gathering and refinement of this mass of data. This complex and currently mostly informal process can be supported using models that allow handling data at a high level of abstraction. However, not all relevant information can be embedded in a single model. Thus, the various models ought to be consistent and coherent with one another. This paper discusses methodological issues. We present a set of issues raised by the gathering and the modeling of data and some issues raised by their consistency. These issues are addressed in a preliminary unifying framework describing the various models, the data embedded in each model and the interconnections of models.

Keywords: Design, Verification, Safety Critical Interactive Systems, Consistency, Reliability, Error-Tolerance

Introduction

Human-Computer Interaction and related disciplines have argued, since the early days, that interactive systems design requires the embedding of knowledge, practices and experience from various sources. For instance, user centered design (Norman, 1986) advocates the involvement of human factors specialists, computer scientists, psychologist, designers … in order to design useful and usable systems. While designing interactive software, the use of formal specification techniques is of great help as it provides non-ambiguous, complete and concise models. The advantages of using such formalisms are widened if they are provided by formal analysis techniques that allow checking properties about the design, thus giving an early verification to the designer before the application is actually implemented.

During design, one should try consider all stakeholders. That is, "persons or groups that have, or claim, ownership, rights, or interests in a corporation and its activities, past, present, or future. Such claimed rights or interests are the result of transactions with, or actions taken by, the corporation, and may be legal or moral, individual or collective" (Clarkson, 1995). The consideration for all stakeholders leads systems designers and analysts to look at the same system (the one to be designed) from multiple perspectives. Such perspectives come from, but are not limited to domains such as human factors, produce development, training, product management, marketing, the customers, design support, system engineers and interface designers. A number of these domains will be discussed more in detail hereafter and more precisely describing the roles they have in supporting interactive safety-critical systems design.

Due to the large number of domains involved, it is highly unlikely that the data gathered, analyzed and documented will be represented in the same way. For example, it is unlikely that the system engineers will take into account all information provided by human factors analysts (for instance about work practice and users). This is not only because of time constraints and the amount of data involved, but also and mainly, because the kind of notation they are used to employ cannot record that information efficiently. This can have serious effects on the reliability, efficiency and error-tolerance of a system. For example, if a task is represented in a task model by a human factors expert and if that information is not represented (in one way or another) in the system model by a systems engineer there is no means to ensure and check that the system will support this task.

It is clear that there is a need for formalizing not only the process of gathering this mass of data, but also for refining and modeling it when necessary in order to provide valuable input to the system design.

The paper is structured as follows. The next section deals with the issues raised by information gathering per se. Section "Sharing and Embedding Information" discusses the feeding and embedding of information from one phase to another within the design process. Section "Formalizing Information**"** deals with the need for formalization of information and data. The following sections discuss multi-type and multi-source data respectively. This data has to be gathered throughout the development process in order to allow designers to reach the ultimate goals discussed in section "Ultimate Goals". The last section (section "Consistency") presents the consistency problem that has arisen from advocating the use of multiple models.

Gathering Information

The phase of gathering information for the design of a new system is crucial for the success of the end product. If performed incompletely, inaccurately or indeed ignored, gaps are left in understanding the scope, concept and function of the new system.

The process of experts gathering data from various domains for input into the system design has been studied as part of the Mefisto Method. 'The process cycle' (Palanque et al., 2000) describes a path that has to be followed to build both usable and reliable interactive systems. In the first phase of the process cycle, the observation phase, information such as work practice, existing artefacts, business and organizational constraints are gathered. Other approaches such as MUSE (Lim and Long, 1994) argue in the same way although the proposed process is different. In that paper, we claimed that in a real life safety critical system, such as in Air Traffic Control (ATC), it is unlikely that the whole domain will be analyzed in detail due to the quantity of data required. This problem will also result in gaps in understanding the scope, concept and function of the new system.

A rich source of information can be obtained from past experiences with similar systems. Since there is such a large amount of data to be gathered, experts can focus on case studies to understand more about the usability of a system and its safety. However, the process cycle (see Figure 1) does not detail how the information is gathered, who will gather it, or how the information will be recorded and reused.

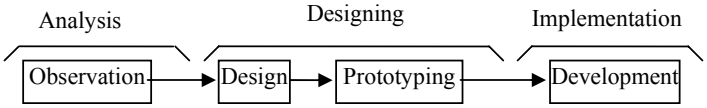


Figure 1 - Schematic view of the Process cycle

Sharing and Embedding Information

Gathering information is not a goal per se. The result of this activity should be used to feed other phases in the design process. This feeding cannot be left informal nor at the discretion of those responsible for these other phases. In addition, not all types of information are closely enough related to build useful bridges between them. On the other hand, some sources of information are so close that, not merging and cross validating them would certainly result in poorly designed and inconsistent systems.

For instance, scenarios and task models both convey information about user activities. It is thus possible to check that scenarios and task models (for the same activity) convey not only the same information but also the same sequencing of operations.

Similarly scenarios and system models both deal with the same operational system and thus ought to contain compatible and coherent information which should be checked at all stages of the development process.

These examples have not been chosen randomly. Indeed, scenarios are the perfect candidate as the corner stone of the consistency and coherence process.

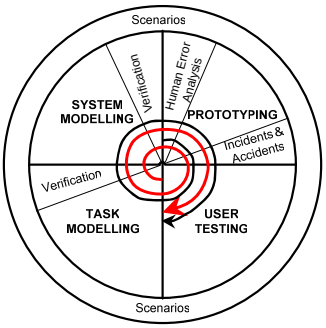


Figure 2 - Scenarios as a corner stone from (Palanque and Navarre, 2000)

Formalizing Information

There is a significant amount of literature on design process for interactive systems design the more referred to being the ones including prototyping activities and evaluations (Dix, 1998 and Hartson and Hix 1992). However little research exists on formalizing the process of 1) documenting the information such that experts of other domains can understand and reuse information for their analysis, 2) refining the information to share only what is necessary and 3) embedding data from one domain to another, all for input into the system design.

Modeling Principles: We promote the use of formal notations so that we can verify the properties of interactive safety-critical systems. Without such notations there are few means for designers to address reliability. However, formal notations may not be adequate for recording information that is idiosyncratically fuzzy and incomplete such as information gathered in the very early phases of the development process. Besides, it is important to note that in most cases, each model will be created by a different person with a different background within a different specialist domain which is likely to influence the kind notation they are able to master. Although it is most likely that one specialist will develop one or several models, they may also contribute to many more models. Thus the relationship between models and specialists can be considered as a many-to-many (M:N). That is, one specialist may contribute to one, zero or many models and one model can receive contributions from one, zero or many specialists. Even for a system that is not safety-critical, it is still necessary to ensure the system's efficiency and reliability but this kind of issue is more salient for this type of system.

Examples of Models: The following section provides an overview of the multiple models used in User Centered Design (UCD) approaches. A number of which can be supported using the UML (Rumbaugh et al., 1997). For example the domain model is supported by class and object diagrams, and the application model which includes the commands and data for the application providers, are the main focus of UML. Some models are only partially accounted for. Task models and scenarios can be described informally and incompletely using UML use cases. Other models are not at all considered in UML for example, user model, platform model and presentation model (Bastide & Palanque, 2003).

We hereafter present more precise information about some particularly relevant models for interactive systems design.

Requirements Model: The functional and non-functional requirements of a system are defined in the requirements model. Requirements describe in a declarative way what a system is supposed to do. The description of a requirement models using a precise and un-ambiguous (i.e. formal) notion allows analysing the model and identifying errors or inconsistencies. In addition, tools can generate tests from the requirement models useful for verifying that a system behaves as the original requirements prescribe (Palanque et al., 1997 and Campos and Harrison, 1997).

Task Model: A task model (Diaper and Stanton, 2004) is a representation of user tasks (in order to reach a certain goal) often involving some form of interaction with a system, influenced by its contextual

environment. Task models are used for planning and during various phases of user interface development for example. The models are usually developed by human factor's specialists following an extensive task analysis phase. For the design of interactive safety critical systems, task models can be advantageous for checking the properties of the future system.

User Model: A user model is a collection of information about a user and is a key component for providing flexibility and adaptation. They can incorporate generic information (valid over a wide range of potential users) such as (Card et al., 1983, Fitts 1954, Barnard and May 1994) and represent information about perception, cognition or interaction. Other user models are aimed at representing information for specific users such as (PUMA Blandford and Good, 1997 and OSM Blandford and Connell 2003). This information can be for instance, fed into a system model in the design phase in order to improve flexibility or in the evaluation phase in order to compute predictive performance evaluation (Palanque and Bastide,1997).

Environmental Model: An environmental or contextual model is developed by inspecting aspects of the environment of a current or future system. Information is gathered using techniques such as observation, documentation analysis or interviews. Examples of elements to be studied include location, temperature, artifacts, duration, social aspects and cultural ethics. The model can be used to identify causes of human behavior. Clearly, this can be beneficial for the development of an interactive safety critical system since contextual factors are a way of providing useful adaptation of the system to environmental changes.

Platform Model: A platform model includes a description of the platform and some platform specific characteristics. These models contain information regarding constraints placed on the UI by the platform such as the type of input and output devices available, computation capabilities… The model contains an element for each platform that is supported, and has attributes belonging to each element describing the features and constraints. Although this type of model is particularly useful for ensuring cross-platform compatibility of systems, they are critical when a given system is expected to be made available to several users working with different software and hardware environments.

System Model: System model is, by far, the one that has been studied the most as it is the main raw material of system construction. In the field of interactive systems, most contributions come from the field of software engineering and have been more or less successfully adapted to the specificities of this kind of systems. Since the mid 80s several formalisms have been proposed that were addressing system modeling either at a very high level of abstraction (Dix and Runciman, 1985, Harrison and Dix, 1990) (such as trying to capture the essence of interaction) or at a lower level in order to provide detailed modeling in order to support development activities (Paterno and Faconti, 1992, Palanque and Bastide, 1990). Specific issues raised by interactive systems modeling include, system state, system actions, concurrency, both quantitative and qualitative temporal evolution, input device management, rendering, interaction techniques …

Presentation Model: A presentation model details the static characteristics of a user interface, its visual appearance. The model contains a collection of hierarchically-ordered presentation elements such as sliders, windows and list boxes as far as WIMP user interfaces are concerned. For post-WIMP interfaces such graphical elements include icons, instruments … (Beaudouin-Lafon, 2000 and Van Dam 1997). Current state of the art in the field of safety critical interactive systems is also addressing these issues. For instance, ARINC 661 specification (ARINC 661, 2001) provides a detailed description of interactive components and their underlying presentation platform for new generation of interactive cockpits.

Architectural Model: An architectural model is a high level model of the application which describes the basic building blocks of the application. Examples of established architectural models are Seeheim model (Green, 1985) which makes explicit the user interface part of the application and the Arch model (Bass et al., 1991) which is an extension of the Seeheim model putting even more emphasis on the UI part. The Arch model divides all user interface software into the following functional categories, Functional Core, Functional Core Adapter, Dialogue, Logical Interaction and Presentation. From a modeling point of view, these components are usually dealt with individually. Various modeling techniques are applied to deal with these components and the following section address some of them i.e. domain model (related to functional core modeling) dialogue model and device model (a sub-part of the presentation component).

Domain Model: A domain model is an explicit representation of the common and the variable properties of the systems in a domain and the dependencies between the variable properties. (Czarnecki and Eisenecker, 2000). The model is created by data collection, analysis, classification and evaluation. The term domain covers a wide range of interpretations, for example, the problem domain, business domain and the system/product domain.

Theses models are necessary to understand the domain in which the future system will be built. In the field of safety critical systems the various domains involved (such as ATC, military systems …) have already received a lot of attention. Domain models are readily available and are meant to be exploited before dealing with any system within that domain.

Dialogue Model: A dialogue model is a collection of hierarchically-ordered user-initiated commands that define the procedural characteristics of the human-computer dialogue in an interface model. (Puerta, 2002). Dialogue modeling has been regarded as a particularly hard to tackle issue. A lot of work has been devoted to it and the notations used have evolved in conjunction with interaction techniques. For instance, early work focused on modal interaction techniques (Parnas 1969) and evolved to WIMP interaction styles (Bastide & Palanque 1990) to reach recent and more demanding interaction techniques as in (Dragicevic et 2004 DSVIS) for multimodal interaction.

Device Model: Input and output devices are a critical part of the interactive systems as they represent the bottleneck via which the interaction between users and system takes place. Their behavior is sometimes very complex even though it may be perceived as simple by the users. This complexity may lie in the device itself (as for haptic devices such as the Phantom (Massiem and Salisbury; 1994)) or in the transducers in charge of extending the behaviors of the devices (such as extending the behaviour of a mouse to cope with double or triple clicks that embed temporal constraints) (Buxton 1986, Accot et al.; 1996). Device models can also be viewed as a person's understanding of how a device works (Satchwell, 1997). In the field of safety critical systems describing the behavior of such devices is critical as it makes precise the interaction techniques.

Multi Type Data

The data obtained and analyzed by various domain experts can be considered as multi-type data. We have distinguished between two main types of data, pre-design data and post-design data. That is, data that is available before a system has been designed, and data that is available after a system is designed. This distinction and its impact on systems design are explained in more detail in the following sections.

Pre-design data: Data can be obtained throughout the design process before the system has been developed. Of course, much of this data can be made available and used for evaluation purposes, once a system has been designed. However; we have labeled it pre-design data because the techniques can be applied without the need of the current system.

Within this category of pre-design data, data can be further classified according to the properties of the data obtained. That is, formal or informal, complete or incomplete for example. Figure 3 illustrates on a three-dimensional cube, four examples of techniques that can be applied to obtain data before the system has been designed. By formal and informal we mean whether there only one interpretation of the models or not. Complete and incomplete refer to the fact that the model contains a sub set of the relevant information or deals exhaustively with it. Finally, high and low-level data refer to level of abstraction at which the information is dealt with.

To illustrate the complexities surrounding multi-type data, we have provided an example of seven techniques positioned in the Multi-Type Data Cube. Some of the examples presented in more detail later in this section, have been extracted from previous work on a mining accident case study (Basnyat et al. 2005). This type of presentation is used because of the overlapping properties of the techniques. For example, a Petri-net is considered (in this paper) as formal, complete and low level even though it is possible to use them to represent other type of data.

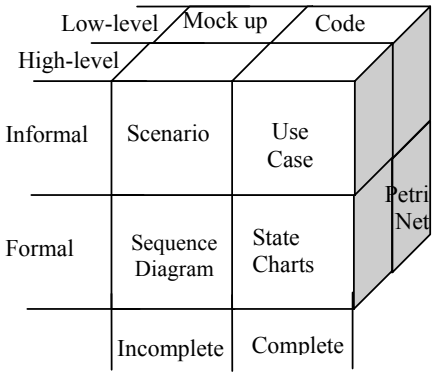


Figure 3 – Multi-Type Data Cube

To give a very brief overview, the case study is a fatal US mining accident (Mine Safety and Health Administration 2002). A Quarry and Plant system is designed to produce cement. However, the part we focus on is the delivery of waste fuel used to heat the plant kilns. The Waste Fuel Delivery System is comprised of two separate liquid fuel delivery systems, the north and the south. Each system delivers fuel to the three plant kilns independently and cannot operate at the same time.

Example of low level formal complete data: Figure 4 provides a simple Petri-net which models the ability to switch from the north waste fuel storage tank to the south waste fuel storage tank using a manual shut off valve.

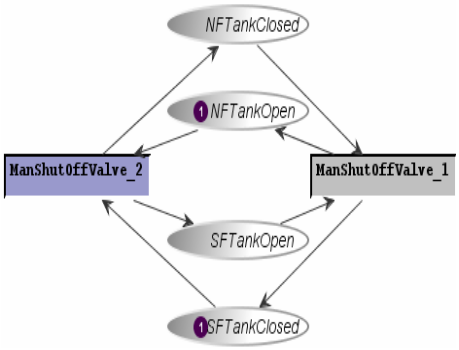


Figure 4 - Formal low level and complete data modeling using Petri-nets

Example of incomplete, informal and low level data: In safety-critical interactive systems design, scenarios can be used to elucidate the particular chain of events that lead to an accident but can also be used to identify alternate failure scenarios that might cause future adverse events. In this particular case study, it could be argued that as a result of the user's actions described in the following brief scenario, a 'hammer effect' occurred causing a fatal explosion. "Mr X closed the valves (after bleeding them) as quickly as possible because of the threat of fuel spreading."

One of the problems associated with ensuring consistency, reliability, efficiency and error-tolerance in the design of an interactive safety-critical system, lies in the probable limited use of fruitful information. Scenarios can be used in line with many techniques, such as task modeling, a priori and a posterior i.e. for design or evaluation activities. A careful identification of meaningful scenarios allows designers to obtain a description of most of the activities that should be considered in the task model. (Paterno & Mancini,

1999). Example of incomplete, formal and high level data: Figure 5 illustrates the event-based sequence diagram that can be used to map out what happened in the lead-up to an adverse event.

Post-design data: The second distinction of data we have made is post-design data. By this, we mean data that can only be obtained once the system in mind has been designed. Examples of such are usability analysis, incident and accident reports or the use of metrics for risk analysis (Fenton and Neil, 1999).

The design of a safety-critical interactive system must be grounded on concrete data, of which may be of multiple source and of multiple type. However, an additional way to compliment and enhance a system's safety is to take into account as much information from previous real life cases. One such type of data is an incident or accident report. To date, input to a safety-critical interactive system design from an incident or accident report has not been considered in a systematic way. We believe these reports can be extremely fruitful to the design of safer safety critical systems. In most cases, these reports are used by assigned experts to analyse why an incident or accident occurred and what could be changed to prevent future similar scenarios from occurring. In contrast, we suggest using the reports to improve future design. To be more concrete, we have implemented this approach on the same mining accident case study previously mentioned.

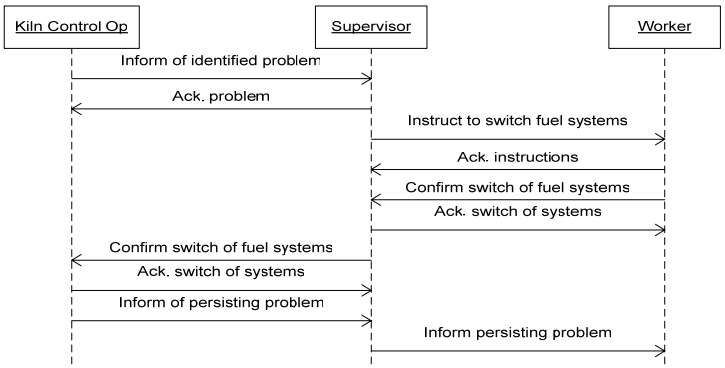


Figure 5 - High-level data, communication sequence diagram

The reports allowed us to achieve two things, 1) obtain and 2) deduce important information that could be embedded into future waste fuel delivery systems of mining plants. Such information obtained includes:

- Add additional fire sensors in the waste fuel containment area to detect heat from fire and activate the fire suppression system more rapidly. Ensure the Programmable Logic Controller (PLC) connectors are properly installed.
- Implement procedures requiring all equipment operators and their supervisors to review manufacturers' instructions and recommendations to ensure machinery and equipment is operated according to manufacturer's guidelines.
- Install audible and/or visual alarm systems in the waste fuel containment area.
- Ensure equipment is installed according to the manufacturer's requirements. Develop procedures and schedules and monitor them to ensure that the required maintenance is performed

Information deduced after implementing and analyzing the results of various safety analysis techniques resulted in the following findings. The system should be designed such that:

- A waste fuel delivery system cannot be started without being primed first.
- Motors cannot be turned on without fuel available in the pipes.
- Air is bled from the pipes before a fuel delivery system is turned on.
- Air cannot be bled while a waste fuel delivery system is on.
- An emergency shutdown button should available to operators.

Multi-Source Data

The data gathered and analyzed for input into a safety-critical interactive system design is collected by multiple specialists of a wide-array of domains. This is due to the nature of safety-critical systems that range from cockpits to surgical equipment to mining instruments to name just a few but also to the variety of information that has to be gathered and the fact that this information stems from multiple domains of expertise. This combination of diverse specialists and diverse domains adds to the complexity of design of a safety-critical system. The following sections describe several such specialists and domains and the input they have on the design.

Human Factors: Human factors is a domain which aims to put human needs and capabilities at the focus of designing technological systems to ensure that humans and technology work in complete harmony, with the equipment and tasks aligned to human characteristics (Ergonomics Society).

Examples of human factors specialists are production engineers, health and safety- practitioners and interface designers. These are just a number of experts in the human factors field who all bring advantages to the design of the system. However, the complexity increases when considering the background of these experts and the ways in which their analyses will vary according to their backgrounds.

Health and Safety Practitioners: Occupational Health and Safety (H&S) practitioners are trained in the recognition, evaluation and control of hazards which place people's safety and health at risk in both occupational and community environments.

Techniques employed by H&S practitioners include risk assessments, postural analysis, legal and organizational factors, work equipment. As with most occupations, health and safety practitioners also have wide ranging educational backgrounds. Such as psychology, anthropometry or physiology. This results in multiple perspectives and methods of working on the same system.

Interface Designers: An Interface Designer is responsible for the presentation of the interface part of an application. Although the term is often associated to computing, the interactive part of a system can include controls and displays in many domains such as military aircraft, vehicles, audio equipment and so on. The educational background of an interface designer can be varied, computer science, graphics design or again psychology. It is probable that a psychologist and a computer scientist will base their interface designs on different principles. Stereotypically, for example, a psychologist may wish to ensure correct colors are used, whereas a computer scientist will want to employ the latest programming techniques with a flashy interface. Both perspectives can be advantageous to the overall design.

Engineering: Systems engineering is an interdisciplinary process referring to the definition, analysis and modeling of complex interactions among many components that comprise a natural system (such as an ecosystem and human settlement) or artificial system (such as a spacecraft or intelligent robot), and the design and implementation of the system with proper and effective use of available resources. (University of Waterloo). In the mining case study, mechanical and automation engineers were involved. However, other types of engineers include hardware, software and systems engineers. The combination of these engineers assists in the system development process.

Hardware Engineer: In reference to the case study, we can assume that the hardware engineers would have been responsible for the design and development of plant components such as the motors, grinders and fuel tank.

Software Engineers: The software engineers in the mining case study would have been responsible for the design and development of applications running on the hardware. Programs include the PLC software and the ‘F’ system software.

Mechanical Engineers: A mechanical engineer can have a variety of responsibilities such as, the design and improvement of machines and mechanisms, organization and maintenance of computer controls for production processes or even selection and installation of equipment for indoor environment control.

Automation Engineer: Automation engineers design, build and test various pieces of automated machinery. This can include electrical wiring, tooling, software debugging etc. One of the main fields of an automation engineer is to design automation systems from a collection of single components of different distributors.

Engineering and the Case Study: A combination of the work performed by the above mentioned engineers can be considered as partial cause for the fatal accident in the case study. One of the events leading to the accident was the failure of the PLC to automatically de-energize the fuel in the pipes when it received signals that the pressure was too high. This automated procedure operated as follows. A monitoring 'F' system received signals from temperature and pressure sensors located on fuel lines. The 'F' system transmits data to the PLC which raises audible and visible alarms in the control room. However, during the accident, the PLC was not connected and therefore did not automatically de-energize the pressure in the pipes.

Certification: Certification is a phase of the development process specific to safety critical systems. This activity involves independent organizations responsible for providing clearances prior to the actual deployment of the systems. This activity has a significant impact over the development process as its successful accounting is perceived by designers and developers as one of the main targets to achieve. Indeed, in case of certification failure, the whole development can be stopped and most of the time restarted with many negative economical and technological consequences. For this reason, certification authorities have developed specific development processes that 'guarantee' the quality of the product by means of structured and reliable processes. For instance DO 178 B (RCTA 1992) is a document describing such a design process widely used in the aeronautical domain.

Incident and Accident Analysts: Incident and accident analysts are interested in understanding system 'failures' and human 'error' often using accident analysis techniques and incident reporting techniques. (http://www.dcs.gla.ac.uk/research/gaag). Such analysts have varying educational backgrounds in computer science for example.

Since we are particularly interested in the domain of safety-critical systems, we have provided definitions of an incident and accident from the Federal Aviation Administration (FAA). An aircraft accident means an occurrence associated with the operation of an aircraft which takes place between the time any person boards the aircraft with the intention of flight and all such persons have disembarked, and in which any person suffers death or serious injury, or in which the aircraft receives substantial damage. (49 CFR 830.2). An aircraft incident is an occurrence other than an accident, associated with the operation of an aircraft, which affects or could affect the safety of operations. (49 CFR 830.2)

Ultimate Goals

The above mentioned issues increase complexity in the design of interactive safety critical systems due to the necessary ultimate goals of embedding reliability, usability, efficiency and error tolerance with the end product. Without such ultimate goals the development process would be far less cumbersome. This is a very important aspect of the work presented here as it points out the issues that are specific to the type of applications we are considering and thus less relevant to others more commonly considered.

Consistency

Consistency is a means to achieve reliability, efficiency, usability and error-tolerance of a system. This can be achieved by means of systematic storage of gathered information into models and the development of techniques for cross models consistency checking.

Model Coherence: One of the problems associated with interactive safety-critical design is the lack of coherence between multiple viewpoints and therefore multiple design models, of the same world. We believe there should be coherence between these design models to reduce the likelihood of incidents or accidents in the safety-critical systems domain. Some work on model-based approaches has tried to address these issues but there is still a lot to do before design methods actually provide a framework to support this critical activity. Indeed, it is still not commonly agreed that there should be a framework for multiple models as some current research argues that models of one kind could be generated from models of the

other kind. For instance (Paternò et al., 1999) proposes to generate user interfaces from task models while (Lu et al. 1999) proposes to generate task models from system models.

A Generic Framework for Ensuring Coherence: Although highly beneficial, it is unlikely that all techniques from all domains of all types of experts will be applied to the design of any given system. This is an unfortunate reality and this is why we are trying to focus on providing a unifying framework to help ensure that data of multiple domains can be gathered, refined and embedded into the design of the system.

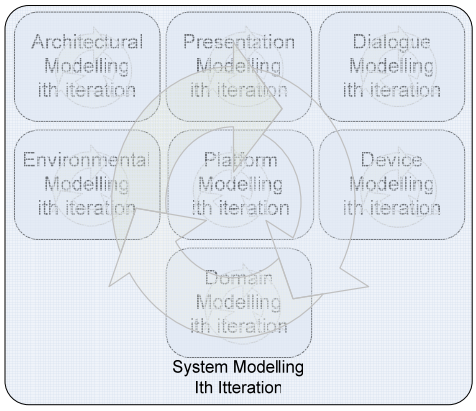


Figure 6 - Ingredients of the system model

As previously mentioned, formalizing this unified procedure is a way of ensuring that there are no ambiguities, that the description of the models and information is precise, that the framework allows reasoning about the system and to ensure consistency and coherence throughout the design and development process.

Figure 6 presents the various ingredients of the system part as described in the section detailing various types of models. This component is reproduced in Figure 7 where interactions with other models is emphasized. Figure 7 presents, as a summary and in a single diagram the set of information, data and processes.

Need For Systematic Tools Support: The complexity of design in the field of safety critical interactive systems clearly requires tool support for the creation, edition; formalisation; simulation, validation; verification of models and information, ability to check for inconsistencies; means for sharing and embedding data; cross-checking of hybrid models … To date, tools exist for the support of individual models, CTTe (Paterno et al., 2001) for supporting various activities around task modeling (edition, simulation, verification …), Petshop (Bastide et al., 1999) for supporting various activities around system modeling. Despite some preliminary work about interaction (Navarre et al. 2001) integration needs are still to be addressed.

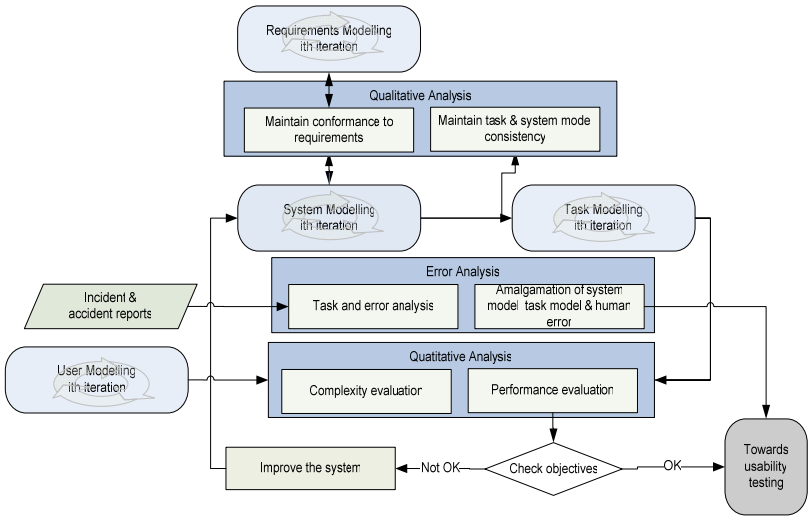


Figure 7 - Generic Modeling Framework

Conclusion

This paper discussing methodological issues, advocates the use of models for the design of interactive safety critical systems. It claims that the issues raised by the design of such systems require the use of systematic ways to support the gathering, refinement and storage of data. This data is, by nature, multi-disciplinary and thus requires a multi-notation approach to support individually each discipline.

However, this multi-notation approach calls for additional means in order to support additional activities such as verification of models consistency. Besides, in order to alleviate the burden for developers and designers, software tools supporting their activities are also at the core of the applicability of such an approach.

We are currently studying methods for integrating the necessary models for safety critical interactive systems design. To date, we have devised two approaches for integrating the task model and system model while taking into account human errors. One approach uses scenarios are bridge between the two (Navarre et al. 2001). The second approach uses task patterns as a means of cross-checking properties between the two models. This work is part of more ambitious work dealing with multiple models for safety critical interactive systems in several application domains including satellite command and control room, interactive cockpits for military and civilian aircrafts, command and control rooms for drones and air traffic control workstations.

References

Accot, J., Chatty, S., Palanque, P.(1996)A Formal Description of Low Level Interaction and its Application to Multimodal Interactive Systems, In Proceedings of the Third Eurographics workshop on Design, Specification and Verification of Interactive Systems, (DSV-IS 96) F. Bodard & J. Vanderdonckt Eds. Springer Verlag 1996.pp. 92-104

ARINC 661 Cockpit Display System Interfaces to User Systems. Arinc Specification 661. April 22, 2002. Prepared by Airlines Electronic Engineering Committee.

Barnard, P. and May, J. (1994) Interactions with Advanced Graphical Interfaces and the Deployment of Latent human Knowledge. Interactive Systems: Design, Specification and Verification. DSVIS 1994 pp15-49

Bass, L., Little, R., Pellegrino, R., Reed, S., Seacord, R., Sheppard, S., and Szezur, M. R. (1991). The Arch Model: Seeheim Revisited. User Interface Developpers' Workshop. Version 1.0 (1991)

Bastide, R., Palanque, P., Sy, O, Duc-Hoa Le, Navarre, D. (1999) PetShop a case tool for Petri net based specification and prototyping of Corba Systems. Tool demonstration with Application and Theory of Petri nets ATPN'99, Williamsburg (USA), LNCS Springer Verlag, 1999.

Bastide, R and Palanque, P. (2003) UML for Interactive Systems: What is Missing in Workshop on Software Engineering and HCI, INTERACT 2003, IFIP TC 13 conference on Human Computer Interaction.

Bastide, R., Navarre, D., Palanque, P. and Schyn, A. (2004) A Model-Based Approach for Real-Time Embedded Multimodal Systems in Militart Aircafts. Sixth International Conference on Multimodal Interfaces. ICMI'04 October 14-15, 2004 Pennsylvania State University, USA.

Basnyat, S., Chozos, N., Johnson, C., and Palanque, P. (2005) Multidisciplinary perspective on accident investigation. Submitted to the Special issue of Ergonomics on Command and Control.

Beaudouin-Lafon, M. (2000). Instrumental interaction: an interaction model for designing post-WIMP user interfaces. CHI 2000: 446-453

Blandford, A. & Connell, I. (2003) Ontological Sketch Modelling (OSM): Concept-based Usability Analysis Proc. Interact 2003. 1021-1022. IOS Press.

Blandford, A. and Good, J. (1997) Programmable user models - exploring knowledge transfer between devices. PUMA working paper WP5.

Booch, G., Rumbaugh, J., Jacobson, I. (1999) The Unified Modelling Language User Guide. Addison-Wesley

Buxton, W. & Myers, B. (1986). A study in two-handed input. Proceedings of CHI '86, 321-326
Campos, J. C. and Harrison, M. D. (1997) Formally verifying interactive systems: A review. In M. D. Harrison e J. C. Torres, editors, Design, Specification and Verification of Interactive Systems '97, Springer Computer Science, pp 109--124. Springer-Verlag/Wien, Junho 1997.

Card, S.K., Moran, T.P. & Newell, A. (1983). The Psychology of Human-Computer Interaction, Lawrence Erlbaum, New Jersey

Carroll, J. M. (1995). Introduction: the scenario perspective on system development. In J. M. Carroll (Ed.) Scenario-based design: envisioning work and technology in system development (pp. 1-18). New York: John Wiley & Sons, Inc.

Clarkson, M. B. E. (1995). A stakeholder framework for analyzing and evaluating corporate social performance. Academy of Management Review, 20: 39-48

Czarnecki, K and Eisenecker U. W. (2000). Generative Programming—Methods, Tools, and Applica-tions. Addison-Wesley, 2000. ISBN 0-201-30977-7

Diaper, D. and Stanton, N.A. (2004) The Handbook of Task Analysis for Human-Computer Interaction. Lawrence Erlbaum Associates.

Dix, A. and Runciman, C. (1985). Abstract models of interactive systems. People and Computers: Designing the Interface, Ed. P. J. &. S. Cook. Cambridge University Press. pp. 13-22.

Dix, A., Finlay, J., Abowd, G., & Beale, R. (1998). Human-computer Interaction. Prentice Hall, Second Edition, Prentice Hall Europe. ISBN: 0-13-239864-8.

Fenton N. E. and Neil M, (1999). Software Metrics and Risk, Proc 2nd European Software Measurement Conference (FESMA'99), TI-KVIV, Amsterdam, ISBN 90-76019-07-X, 39-55, 1999.

Fitts, P. M. (1954) "The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement." Journal of Experimental Psychology 47. pp. 381-91

Green, M. (1985). Report on Dialogue Specification Tools", in G. Pfaff (ed.). User Interface Management Systems. New York: Springer-Verlag, 1985, 9-20.

Harrison, M and Dix, A. (1990) State Model of Direct Manipulation in Interactive Systems. In Formal Methods in Human-Computer Interaction. Cambridge Series on HCI. Edited by M. Harrison and H. Thimbleby. P.129

Hix, D. and Hartson, H.R. (1993). Developing user interfaces: ensuring usability through product and process. John Wiley and Sons, New York. ISBN: 0-471-57813-4.

Lim, K.Y and Long, J.B (1994) The MUSE Method for Usability Engineering. Cambridge University Press, Cambridge.

Lu, S., Paris, C., Vander Linden, K. (1999) Toward the automatic construction of task models from object-oriented diagrams. Engineering for Human-Computer Interaction. Kluwer Academic Publishers. 169-180.

Massiem T. H. and Salisbury J. K.. (1994) The phantom haptic interface: A device for probing virtual objects. In Proc. of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, Chicago, IL, USA, November.

Navarre, D., Palanque, P., Bastide, R., Paternó, F., and Santoro, C. (2001). "A tool suite for integrating task and system models through scenarios." In 8th Eurographics workshop on Design, Specification and Verification of Interactive Systems, DSV-IS'2001; June 13-15. Glasgow, Scotland: Lecture notes in computer science, no. 2220. Springer

Norman, D. A. (1986). Cognitive Engineering. In D. A. Norman & S. Draper (Eds.). User centered system design: New perspectives in human-computer interaction. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc

NTSB Title 49 – Transportation, Subtitle B – Other Regulations Relating to Transportation, Chapter VIII – National Transportation Safety Board. Part 830 - Notification and Reporting Of Aircraft Accidents or Incidents and Overdue Aircraft, and Preservation of Aircraft Wreckage, Mail, Cargo, and Records. Section 830.2 Definitions.

Palanque, P and Bastide, R (1990) Petri nets with objects for specification, design and validation of user-driven interfaces. In proceedings of the third IFIP conference on Human-Computer Interaction, Interact'90. Cambridge 27-31 August 1990 (UK)

Palanque, P and Bastide, R. (1997). Synergistic modelling of tasks, system and users using formal specification techniques. Interacting With Computers, Academic Press, 9, 12.

Palanque, P., Bastide, R., Paternò (1997) Formal Specification as a Tool for Objective Assessment of Safety-Critical Interactive Systems. INTERACT 1997: 323-330

Palanque, P., Navarre, D. (2000) Gaspard-Boulinc, H. (2000) MEFISTO Method version 1. The Mefisto Project ESPIRIT Reactive LTR 24963 Project WP2-7. September 2000.

Parnas, D.L (1969). On the use of transition diagrams in the design of a user interface for an interactive computer system. In Proceedings 24th National ACM Conference, pp. 379-385.

Paternò F., Mancini, C. (1999) Developing Task Models from Informal Scenarios, Proceedings ACM CHI'99, Late Breaking Results, ACM Press, Pittsburgh, May 1999.

Paternò. F. and Faconti, G. (1992), in Monk, Diaper & Harrison eds. On the Use of LOTOS to Describe Graphical Interaction People and Computers VII: Proceedings of the HCI'92 Conference, Cambridge University Press, pp.155-173, September, 1992.

Paternò, F., Breedvelt-Schouten and N. de Koning. (1999). Deriving presentations from task models. In Engineering for Human-Computer Interaction. Kluwer Academic Pub. pp. 319-338.

Paternò, F., Mori, G. and Galimberti, R. (2001) CTTE: An Environment for Analysis and Development of Task Models of Cooperative Applications, In ACM Proceedings of (SIGCHI'2001), March 31-April 5, Seattle, WA. (Extended Abstracts). 21:22

Puerta, A.R. (2002) The MECANO Project: Comprehensive and Integrated Support for Model-Based Interface Development. In (CADUI'02), pp. 19-35.

RTCA/DO-178B. Software Considerations in Airborne Systems and Equipment Certification, December 1. http://www.rtca.org/ (1992)

Rumbaugh, J, Jacobson, I and Booch, G. (1997) Unified Modeling Language Reference Manual, ISBN: 0-201-30998-X, Addison Wesley, est. publication December 1997

Satchwell, R.E. (1997) Using Functional Flow Diagrams to Enhance Technical Systems Understanding. Journal of Industrial Teacher Education. Volume 34, Number 2. Winter 1997

Stirewalt, K., and Rugaber., S. (1998) Automating UI Generation by Model Composition. Automated Software Engineering 13th IEEE International Conference October 13-16, 1998

United States Department Of Labor Mine Safety And Health Administration Report Of Investigation Surface Area Of Underground Coal Mine Fatal Exploding Pressure Vessel Accident January 28, 2002 At Island Creek Coal Company Vp 8 (I.D. 44-03795) Mavisdale, Buchanan County, Virginia Accident Investigator Arnold D. Carico Mining Engineer Originating Office Mine Safety And Health Administration District 5 P.O. Box 560, Wise County Plaza, Norton, Virginia 24273 Ray Mckinney, District Manager Release Date: June 20, 2002

van Dam, A. (1997) Post-WIMP User Interfaces, Communications of the ACM 40, 2, 1997, 63-67

Websites

Enterprise Architect User Guide Version 4.5
http://www.sparxsystems.com.au/EAUserGuide/index.html?sequencediagram.htm

http://www.ergonomics.org.uk/ergonomics/definition.htm

University of Waterloo. What is systems design engineering?
http://sydewww.uwaterloo.ca/SystemsDepartment/WhatIsSystems/whatissystems.html.

http://www.dcs.gla.ac.uk/research/gaag

Uncovering the Information Needs in Complex Aerospace Systems

Iya Solodilova and Peter Johnson

Flightdeck Safety, Department of Computer Science, University of Bath
I.Solodilova@bath.ac.uk; P.Johnson@bath.ac.uk.

Abstract: A modern cockpit is the heart of a complex aerospace system. Representing complex information in a way that pilots can understand effectively remains a challenge. To understand how best to support pilots' information needs, we face the difficulty of having to study the complexity of the activity itself, the complexity of the environment, and the hidden nature of the information needs of the pilots. This paper shows how a "cue-recall debrief method" can be used as a powerful technique in investigating pilots' cognitive processes and activities. Moreover, it is claimed (Omodei, Wearing & McLennan, 1997) that this technique has little effect on the complexity of the activity, operating environment and the pilot's experience. Using this method has uncovered distinct information-evolution stages, references, and strategies that pilots use when flying an automated aircraft.

1. Introduction

Research shows that pilots have difficulties understanding automated aircraft systems (Sarter & Woods, 1994; 1995). Aviation Human Factors Experts involved in the design of advanced cockpits report that information content and the format of presentation on the interfaces gives little consideration to pilots' information needs (Newman, & Greeley, 2001). The information content is based on an ad-hoc approach, influenced by previous designs and availability of latest technological trends.

The modern glass cockpit is a complex environment and the tasks required of modern pilots are similarly demanding. A modern pilot must be constantly monitoring the condition of the aircraft. This involves repeatedly switching focus between different instruments and displays, while efficiently guiding the aircraft to its destination and planning for what the aircraft will need to do in the future.

Various forms of analysis have been used to analyse this environment and the demands placed on the pilot. However, such analyses generally divide pilots' work reducing them into 'chunks' in a 'vertical' fashion.

However, we argue, that for time critical, dynamic and evolving environments accurate temporal flows cannot be easily preserved using these approaches, as 'chunks' cannot adequately depict the information *flow* that exists the modern cockpit.

Additionally, retrospective interviews and structured questionnaires are common techniques that are used to inform such approaches. The interview, for example, cannot capture the temporal aspects of the aerospace environment and pilots' activities. The questionnaire is restricted by the predetermined content of the questions, which cannot adapt to the answers of already answered questions by the pilot. These approaches, in most cases, look for confirmation of information that is already known to the researcher. They are poor on discovering from real-time observations and pilots' own interpretation of information that pilots use, for example the presentation form, frequency, quantity and quality of information.

Methods are required that trace the evolution of information from the beginning to end and that can be used to inform the future design of interfaces of complex systems.

2. An Evolutionary Approach

To address the last problem we have devised a three-step approach that aims to uncover pilots information needs in the complex domain of aerospace and informs interface design. The *first step* involves capturing real-time data, where a pilot wears a head-mounted camera whilst flying an uninterrupted flight from beginning to end. The *second step*, the cued-recall-debrief interview, takes place immediately after the flight where the pilot reviews captured video footage with the researcher. Both of these steps are based on the 'cued-recall-debrief' method (Omodei, Wearing & McLennan, 1997), which we have tested and specifically modified for our approach during a preliminary study. The video footage captured from the pilot's point-of-view provides a powerful stimuli for "… evoking the recall of a wide range of cognitive and affective experiences with minimum distortion of the complexity and dynamics of these experiences" (Omodei, Wearing & McLennan, 1997). This cued-recall-debrief step reveals elements of pilots' thought processes and tracks pilots' needs for vital cues and information throughout the flight. The captured video

footage is interpreted by the pilot and serves as a guide to a researcher in later analysis, which is the *third step* of the approach.

An advantage of our three-step approach is that the empirical study and data analysis preserve the complexity of the environment and workflow, but do not influence it or interrupt it. In contrast to other observation studies where the researcher either interrupts the workflow, to ask questions about the thinking process of the operator, or asks the questions after the work has been completed, relying on the operator to recall the right moment and events that followed.

There are three main advantages to this approach (for more details, see Appendix). First, it acquires information without imposing a predetermined structure by a researcher. The structure and the content of information is guided by the events of the flight itself. Second, the probes for cueing pilot's comments and for identifying pilot's information requirements are provided through reliving the event by the pilot from his/her own-point-of-view. Third, the approach traces the evolution of information throughout the entire flight without interruption of any activities.

Steps One and Two of the Approach

Set up: Participants flew a full motion level five Hercules C130-J flight simulator on a regular flight from Sydney to Richmond. Each flight lasted between 15 to 20 minutes. Pilots flew one flight with full utilisation of automation and the second flight with minimum use of automation (see table 1).

Pilot		**Automated**		**Non-Automated**	
		Flight	**Debrief**	**Flight**	**Debrief**
Crew A	1	20 min	1 hour 30 min	20 min	1 hour 30 min
	2	20 min	1 hour 30 min	20 min	1 hour 30 min
Crew B	3	20 min	1 hour 30 min	20 min	1 hour 30 min
	4	20 min	1 hour 30 min	20 min	1 hour 30 min
TOTAL		**1 hour 20 min**	**6 hours**	**1 hour 20 min**	**6 hours**

Table 1 - Empirical Study set up

All flights for all participants were identical, and included three characteristics:

- *A runway change* aimed to increase pilots' workload. It allowed observation of how pilots dealt with new information and how they made changes to cockpit set up.
- *Simulation of other traffic*, on the radar display and on the radio to make the flight as realistic as possible.
- *A cloud base between 1500 to 25000 feet* to prevent pilots from seeing the ground at the top of a climb and to encourage pilots to rely on and use instruments throughout the flight. This also allowed pilots switching between instrument and visual operation during take-off and landing.

Rationale: The rationale behind observing *an automated and non-automated flight* was as follows:

1. Observation of an automated flight shows the problems pilots face and the workarounds pilots have invented to deal with recurrent problems with automation (Heymann, 2002). It also highlights where automation is most useful and is effortlessly used by pilots.
2. It well known that the Standard Operating Procedures that pilots use are devised, at least partly, to help pilots to overcome problems of poor automation design (also referred to as 'an indirect admittance of poor design', Demagalski, et al 2002).
3. Observing non-automated flight is less affected by automation-design-induced errors and shows pilots'-ways of dealing with information and 'pilot-like' operations, thus identifying activities that are based on deep-rooted 'flying experience'.
4. It is claimed that pilots use numerous strategies when collecting and using information in the automated and non-automated cockpit settings. Video footage captured both settings.
5. A non-automation fight operation focused on pilots using a more basic level of information available in the environment and the cockpit. In comparison, the fully automated flight focused on how pilots' obtained their necessary information with an 'abundance' of information available.

Rationale behind observation of the whole flight: We have observed the whole flight from 'power-up' to 'power down' to capture the following phenomena: (1) the aircraft environment is dynamic and time-critical, where current events are affected by past and present events and in turn affect subsequent events;

(2) the information is also dynamic, constantly changing and dependent on evolution of all events. It is wrong to separate this information flow. Pilots deal with rising situations that are full of information that is dependent on the progress of the flight. The study of isolated stages of flight does not show the extent of how pilots build on and construct information, how information evolves and how having or not having a specific piece/s of information affects subsequent flight stages.

Participants: The study involved observations of two crews (with two pilots in each crew) in a simulator. All participants were male military pilots. Pilots had on average 1600 (SD = 663) total flying hours and had extensive experience (on average 825 (SD = 415) flying hours) on the Electronic Flight Instrument System (i.e., an aircraft equipped with automation). All pilots had similar previous flying experience on both types of aircraft with and without use of automation (see table 2 below).

Pilot		**Flying hours**	
		Total	**Electronic Flight Instrument Systems**
Crew A	1	1700	400
	2	1100	800
Crew B	3	2500	1500
	4	1100	600
Mean of flying hours		**1600**	**825**

Table 2 - Flying experience

Step Three: Evolutionary Data Analysis

To avoid breaking down data as much as possible we have adapted an evolutionary data analysis technique that tracks links in data throughout the activity. The analysis can be thought of as a spiral, iterative progression through four stages (see Fig. 1: Evolution of the search): (1st Stage) search for answers to posed questions; (2nd Stage) search for commonalities and pattern; (3rd Stage) identify properties transpired; and (4th Stage) search for additional data with transpired properties. Each stage of analysis allows refining the main posed question, hence allowing the uncovering of more detail for each posed question (see Fig.re 1).
The main posed questions at the beginning of the analysis are aimed at the direction of interest, without limiting the field of search too early. Each main posed question requires several spiralling iterations to refine the question until the question has been either explored in sufficient detail or it cannot be broken down any further into data that would inform the interface design or additional analysis is not required at this time. The questions become more specific and are refined further with every cycle through the stages (see Fig. 1).
Three main questions posed were:
Q1 – What *information* do pilots use to identify aircraft state?

Q2 – Does the information have *structure* and if so, what is that structure?

Q3 – Do pilots have *strategies* in assembling and using information to identify aircraft state?

Figure 1 shows how the first question (Q1 - Information) is refined through several iterations (see numbered arrows from question to question in Fig. 1). The next two main posed questions (Q2 – Structure and Q3 - Strategy) would go through the same process as the first main posed question in figure 1 (see centre of the figure 'Q1 - Information'), with the only difference that at the centre of the figure there would be Q2 – Structure and Q3 – Strategy. Both of these questions would require their own iterations through four stages of analysis with surrounding questions aimed to answer the centre main poised question in required detail.

3. Results

The captured video footage (i.e., step one of the approach) and recorded cued-recall-debrief interview (i.e., step two of the approach) were transcribed to one document for each recorded flight for ease of analysis. This video data and the transcript were analysed using the evolutionary four-stage data analysis described above. We give several examples of data analysis for each main posed questions, followed by the summary of all results (see Fig. 2). The proceeding examples show a small proportion of the analysed data. It is timely at this stage to point out that all the analysis was done using real-time video footage and therefore the analysis is not removed from the original data.

Q1 - 'What *information* do pilots use to identify aircraft state?

The analysis began with identifying and collecting data to answer the first main posed question that explored the area of *information* (see centre in Fig. 1). During the first iteration through the data having the posed question in mind, we were looking for words that would identify 'information' that pilots use. The first words in the transcript that hinted to answer the question were '*referenced to*' or '*referred to*'. Around these words, we were likely to identify information that pilots used or 'referenced' in flight. Consider the following examples:

Pilot 04:42 M: …also just quickly *referencing* in for the **airspeed** for our rotate.

Pilot 06:42 M: …Just checking that generally set the right height, above the altitude tape there, checking **the cyan figure** …all of the *reference* number in a different colour, that they are the same, any *reference* figure is all in cyan. So if you see a blue *reference* number anywhere, that's it. That's a *reference* **number**.'

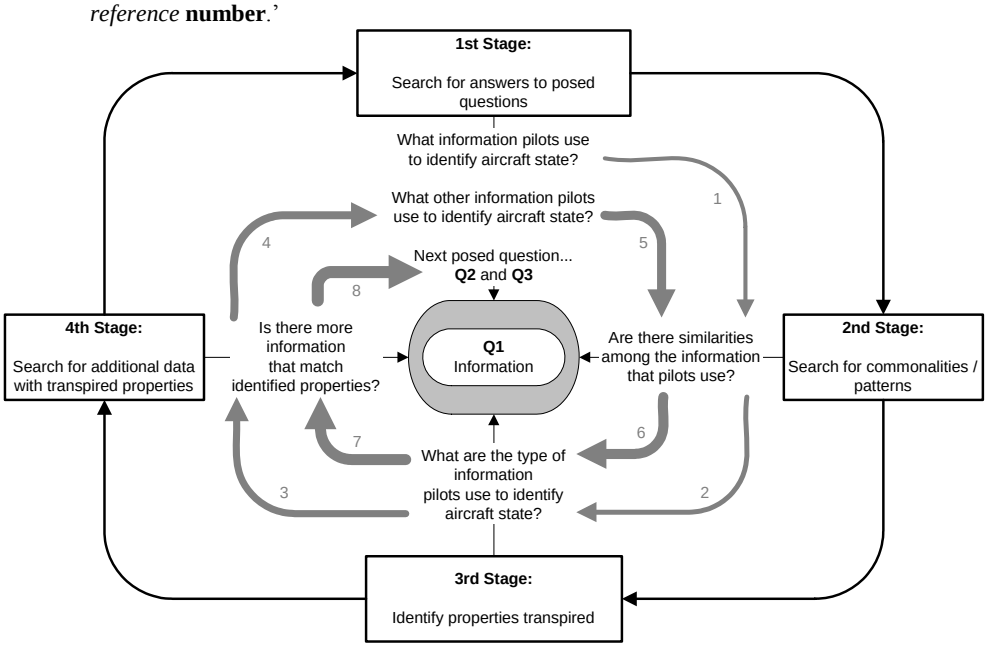


Figure 1 – Evolution of the search

The initial analysis of the above transcript suggests that the pilot is 'referencing' specific information, such as the '**airspeed**' to indicate to the pilot the next action **'**for our rotate'. From the transcript above we can see that the pilot is 'referencing' several instruments either to verify current *aircraft behaviour* or using the 'referenced information' to identify *the time to active next behaviour*, such as in the example 04:42.

This leads to the second stage of the analysis (see Fig. 1) in which patterns and commonalities in the data are identified. A pattern in timing begins to emerge, for example every two minutes throughout the flight the pilot is 'referencing' instruments to establish the aircraft's behaviour. The pilot identifies pieces of information on individual instruments as 'references' to determine the aircraft's behaviour. This is a commonality among the 'referenced information', which we found in the transcript around the words '*referenced to*' or '*referred to*'. This can be defined as a property of 'referenced information'.

The third stage of analysis involves identifying the properties of already established information. Thus far, the *transpired properties* of the 'referenced information' are: (1) the information is referenced throughout the flight at similar intervals of time. (2) The information is required to verify current aircraft behaviour. (3) The information is used as a *reference* (e.g. airspeed or height) to identify the moment of activation for the next behaviour.

The fourth stage involves searching for information that matches the described properties during stage three. Running through the data during the second iteration, keeping in mind the properties described above, the words '*constantly*', '*watching*' and '*monitor*' appear to point at information surrounding them

that possess listed properties. Hence, we are searching the transcript further to find words, such as the words '*constantly*', '*watching*' and '*monitor'*. Consider the following examples:

Pilot 06:28: …he is *constantly watching*, if I haven't busted a height (i.e., pilot jargon for – to break Air Traffic Control altitude restriction), airspeed or a heading or whatever…

Pilot 11:10: …I'm *watching* the speed caret come up and go above the wing, because we want to accelerate, but as to how much that goes before you get to 210 knots it's something that I had to *constantly monitor,* once I got to 210 knots, then I had to pull power back making sure the caret was on the wing. So it did not raise the workload a great deal, but it did a little bit. There is nothing that really tells you after 210 knots at this height that you need to set this power.

From the analysis of a complete flight it appeared that the timing of a pilot's comments containing words '*referenced to*' or '*referred to*, '*constantly*', '*watching*' and '*monitor*', fell into two-minute cycles. The property of all information surrounding these words are also relevant to aircraft behaviour, both verifying current aircraft behaviour (see transcript 06:28) and identifying the moment of activation for the next behaviour (see transcript 11:10). The data is then searched again using the 4th stage of analysis for the words that are similar in meaning to '*referenced to*', '*constantly*', '*watching and 'monitor'.* Doing this, a new fourth property is seen to transpire.

After examining what the data uncovered during the 4th stage, the new property of '*referenced information*' established itself. The 'referenced information' was *compared to some other features* to establish its correct or required position. This can be observed in the following two comments by the pilots, '*I'm watching the speed caret come up and go above the wing*' and '*then I had to pull power back make sure the caret was on the wing*'. The 'referenced information' here is the *speed caret symbol* and it is compared to a stationary relatively unchanging reference, the *wing symbol* on the display.

In both instances the 'referenced information' (i.e., the caret) would have no significance if it was not referenced against another feature (i.e., the wing) that was *constant* and *unchanging* relative to a monitored symbol. Thus, a new property of the 'referenced information' is established, i.e., the reference should be *constant, unchanging* and relative to another feature. Then again, to make sure it is not only specific to this piece of data, a next iteration through already collected data is required. All data has to be analysed again keeping in mind all four established properties.

Having analysed all captured video footage and transcripts from eight flights (i.e., four automated and four non-automated) is was discovered that the *information* (i.e., according to the first posed question – Q1) pilots used to identify the aircraft behaviour and to establish the point in time to activate the next event in both flights, automated and non-automated, was the same.

The properties of this type of 'referenced information' are:

- It is referenced throughout the flight at similar time intervals (e.g. two-minute cycles)
- It is required to verify current aircraft behavior
- It is required to maintain aircraft behavior
- It identifies specific conditions, limitation or boundaries of the system
- It is used to identify the moment of activation of next event/behavior/maneuver
- It is usually connected to other feature/or relative to them
- It is compared to other constant and unchanging features on the display
- When it crosses another feature it becomes a complete symbol (e.g. the wing and caret)
- Pilots have a picture in mind of how this 'referenced information' should align and wait for that moment of alignment to signify the next event

Q2 – Does the information have *structure* and if so, what is that structure?

The second question is now placed at the centre of Figure 1 entitled 'Q2 – Structure'. The same iteration through four stages has been undertaken. Initial iterations through data showed that pilots in fact had and used 'information structures' that helped them assemble and recall information. These 'information structures' became apparent when pilots used similar types of information in the same order. The scrupulous reading of the transcripts and the reviewing the captured video footage, produced the following information structures:

- Air Traffic Control call

- ATIS (i.e., Automatic Terminal Information Service) announcement
- Structure of a navigation plate (see table 3)
- Brief (e.g., Take-off, Landing) also has a structure
- Operating procedures
- Checklists

The 'structure of information' was found to be either imposed by something physical in the cockpit, such as a display layout, a navigation plate, or it was imposed by an operating procedure. The table 3 below shows two identified 'information structures' (i.e., Take-off brief and Navigation Brief). The real-time data column contains the original transcript from the flight and the 'cued-recall-debrief interview' column provides pilots comment on memorising information in a specific order provided on the plate. This 'information structure' is also reinforced by the operating procedure, which specifies the order in which the information is read from the navigation plate.

TIME LINE	SEQUIENCE OF EVENTS:	STEP TWO: REAL-TIME DATA	STEP ONE: CUED-RECALL-DEBRIEF INTERVIEW
03:08	TAKE-OFF briefing	'Glenfield 1 departure out of here runway 10; plate stated 4 October 2001, no amendments; gradient required 3.3%, which we can do; track 095 and 1TAC or 1000 feet, which ever is later, turn right, track 170 to intercept 144 for Richmond NDB, track to Glenfield then as cleared.' '*Copy*'	'All that is just interpreting what's on the plate there & by briefing it, it's actually putting into, right in to our minds, instead of always refer to it, some of it can be done from memory. And usually what I will do with departure, some of the departures would be quite long and complex. However, you really cannot keep all of that information in your head, so what you do is brief the First (i.e., First Officer – the co-pilot) or you just remember two to three instructions, so like maintain heading 095, 1000 feet or 1 TAC. Next what I'm going to do is turn, right turn on TACAN distance. TACAN is …a type of DME (i.e., Distance Measuring Equipment).

Table 3 – Transcript of two steps of the approach

Structuring information appears to be helpful to pilots in recalling and executing actions. Structuring of information happens during 'briefings', such as the brief before the flight or take-off. Structuring information helps pilots to remember 'information' at crucial point during the flight. Here is an example:

> Pilot 17:55: I am also, next thing I'm looking at validating the ILS by that outermarker check height again. And PNF (i.e. Pilot-Not-Flying) briefed a little bit before, as to what the height (i.e. 1295 feet), distance (i.e. 4.7 miles) we were looking for, so that's a next step.

This transcript shows the 'information structure' that is purposely placed along the timeline of the flight. The events of the flight are announced in the order that they were briefed earlier.

We analysed data from all flights and having iterated through all four stages of the analysis, we again established that the 'information structures' pilots used where not different between those used in automated and in non-automated flights.

Q3 – Do pilots have *strategies* in assembling and using information to identify aircraft state?

Q3 is the last main posed question that focuses on identifying 'information strategies' that pilots use to help them deal with vast amount of information. As from the previous example it can be seen that pilots utilise existing information structures to recall and assemble required information. It appears that pilots use this 'information strategy' throughout the flight to assemble 'referenced information' (i.e., the information identified in Q1 section).

Another obvious strategy pilots used was a 'scan'. This 'information strategy' was used to collect and update information they already knew about the state of the aircraft. We searched to this word 'scan' in the transcript to identify how often and for what type of information pilots use this strategy. The word 'scan' appears over ten times in just a single flight transcript. See one example below:

Pilot 04:53: All I'm looking for there on the PFD (i.e., Primary Flight Display), now my focus has come in inside once we are far away from the ground. All I am doing is getting my *attitude* and *heading* set on the PFD, so I'm concentrating on *putting the climb-dive marker where I want it*. Obviously we don't have any reference information there now, so I am just looking at the reference, *the pitch ladder*. So that's all. How many degrees I want & I was looking for about *7 degrees nose up* there. That's usually a good figure to remember. As *accelerating* at a nice *rate*, but not to quick, so you are not going to over speed the gear or anything like that. The other part of my **scan** is looking down at the compass card and quickly referencing and having a look at the level on there as to what heading I am flying.

Pilots also use 'a scan strategy' to maintain and to verify the aircraft's behaviour. Pilots identify, assemble and position 'referenced information' in their mind at similar time intervals along a flight timeline prior the flight or during a flight brief. The aligned 'referenced information' is then checked during the flight at assigned time intervals for accuracy against initial identified 'referenced information'.

Pilots also employ 'a brief strategy' to construct the information before the entire flight during a pre-flight briefing session and before each significant part of the flight throughout the flight. Pilots are constructing and aligning 'referenced information' in their mind, which can either be a visual cue in the environment (e.g. a view of the airport approaching from South) or on the display (e.g. altitude reading). Pilots would either image or draw the flight on the board before the flight, establishing important 'referenced information' (e.g. Navigation point on the display or Altitude). In the example 03:08 (table 3) the pilot uses "…*track 015 (i.e., heading) and 1TAC (i.e., navigation point) or 1000 feet (i.e., altitude)"* as major information 'references' to help establish significant points in flight that would indicate the time for an action to be executed "…*turn right, track 170"*. The pilot's comments state that, '…*by briefing it (i.e., take-off), it's actually putting into, right in to our minds…some of it can be done from memory'*.

These 'information strategies' were identified through iteration of four stages of analysis, refining the key words or structure of sentences that were repeated several types. For example, comment 03:08 shows how pilots briefed the departure and several minutes later the co-pilot executed briefed actions simultaneously recalling information briefed (e.g. '*heading 095, 1000 feet or 1 TAC*'). To identify similar strategies we searched the transcript for similar situations where information briefed was recalled.

We also found that pilots used an 'information strategy' to help them recall required information by *constructing future references using a timeline-sequence structure***,** which later in the flight triggered recollection of the required action to be executed.

These are only few examples of the strategies employed by the pilots in information utilisation identified in automated and non-automated flight.

4. Information Evolution Throughout the Flight

The four-stage analysis resembles tracing a 'spider-web' of information and no matter where the researcher starts until most of the routes that makes the picture complete are identified the analysis is not finished.

The analysis revealed that all information pilots use is connected or related to other pieces of information via an 'information structure' or an 'information strategy'. The information pilots used was constantly evolving. Pilots used strategies to update and generate new references (i.e., 'referenced information') to keep up with evolving information. Pilots applied 'information strategies' using 'referenced information' and 'information structures' to maintain flight path and the required aircraft behaviour.

As a result of the analysis a model emerged illustrating how pilots acquire and use information. A model consisting of eight phases is given that shows the spiral of how pilots acquire and use information, how information evolves and how pieces of information relate to other 'referenced information' (see Fig. 2). The model evolves into a spiral at each progression through eight phases, representing pilot's progression of acquiring information, gain knowledge and experience.

The **first phase** of information progression represents that the pilot has an existing knowledge based on experience, for example an existing 'information structure', such stages of the flight or steps in the operating procedure. For example, Air Traffic Control calls, and flight briefs, are strategies that the pilot has acquired through training and on-line operation. This first phase would also include a request for new flight information. At this point the pilots will build on the existing knowledge (i.e., previously used 'referenced information' and 'information structures') and experience (e.g. 'information strategies', such as 'brief'), adding new information to old 'information structures' for example.

The **second phase** represents the acquisition of all new information related to the flight. All information regarding this flight will be introduced during this phase, the brief, the planning of the route, route related

weather and restrictions. The second phase is also the beginning of the information acquisition phase and the processing of new information that continues through the next four phases.

The **third phase** of information acquisition involves identifying alternative airports and all the related information for alternative arrangements, such as additional route calculations, relevant weather and restrictions on route (e.g. height, speed, no-fly zone). New flight regulations and restrictions that are relevant or have been introduced will also be introduced in this phase.

The **fourth phase** is the choice of information acquisition and this is where new solutions, identification, division of work and problems are assigned between pilots. At this phase new information is generated out of all information previously acquired prior to the flight. The calculation of relevant-to-the-flight-information-references happens at this phase. If this phase is to be associated with a flight stage, it would be a 'brief before take-off', a 'brief' before a significant event or a 'brief' due to a change in original flight plan, containing more new information.

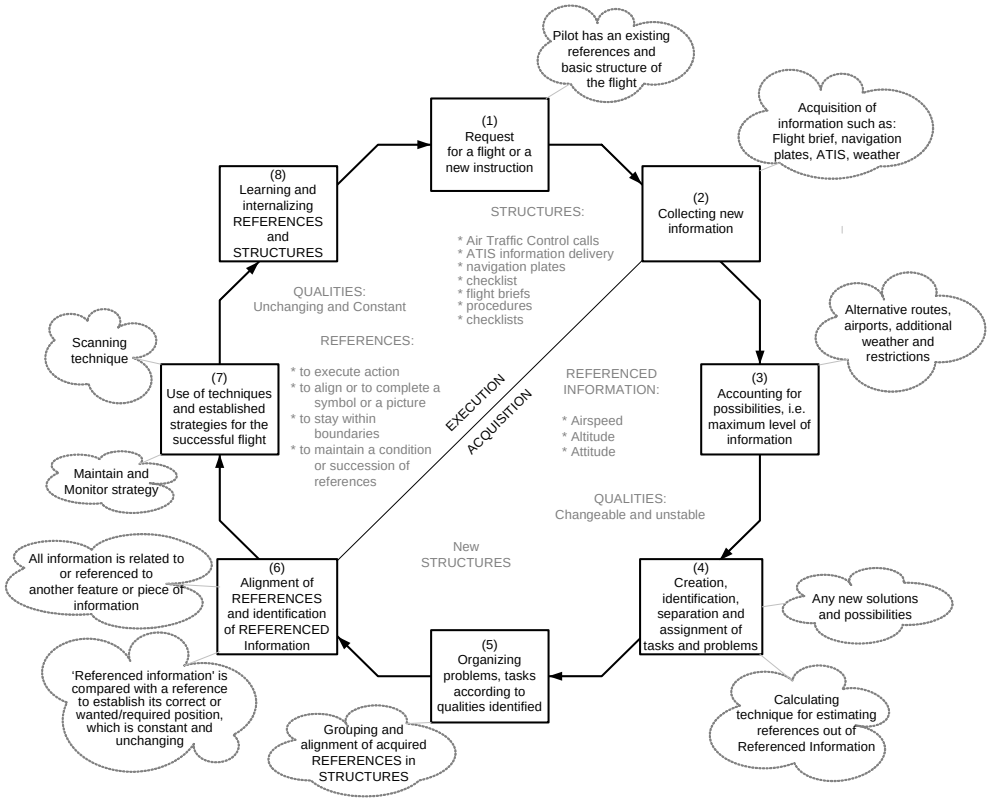


Figure 2. Evolution of Information Flow

The **fifth phase** involves organising information. This is where pilots group and align references that they later use in flight. At this phase the information gets sorted into structures to assist pilots in implementation of their strategies.

The **sixth phase** is the end of information acquisition and beginning of information use. All new acquired 'referenced information' and 'information structures' are compared with existing references and structures that pilots hold in their mind. All pieces of information fall into place; the blend of information, structures, new and old happens in this phase in the pilots mind. Pilots compose the references in their mind and position them on the display relative to other information or relative to already existing references. All information is connected and dependent on each other, and the links are established. This is the point of clarity. In this phase the information is not likely to change its position, unless a change in flight plan or situation occurs.

The **seventh phase** is the flight execution phase. All the 'information strategies', for example, 'maintain', 'monitor' and 'scan', pilots use to fly the aircraft are implemented here on the basis of newly acquired and organised information, and previous experience.
The **eighth phase** involves turning all newly attained information, such as 'referenced information' and 'information structures' into knowledge and experience. The iterations through the spiral bring the pilot to a new level, the phase one, with added knowledge and experience from the last flight. New iterations through the eight phases is triggered by a new flight or a change to the flight plan.

5. Conclusion

This paper demonstrated how a three-step approach without interruptions can elicit information that pilot require to fly the aircraft efficiently. The first two steps involved the use a headmounted video-camera's that provided a pilot with his own point-of-view to be a valuable cue to recollect the activities in flight and guide the researcher to information that is vital in a complex and demanding environment. Instead, of a researcher imposing their interpretation of the information structure, the relevance and meaningfulness of the information structure is derived from pilots' activities. That is, in our study throughout the flight the pilot's own-point-of-view, their activities and pilots' recollection of events were the source of all data acquired.
As a result of this study a model of Information Flow (Fig. 2) emerged, that shows how 'referenced information', 'information structures' and 'information strategies' evolve. This has been depicted diagrammatically (see Fig. 2) and shows how complex the evolution of information is during piloting of the aircraft. It shows how information is coming from many sources, is constantly changing, and being affected by events throughout the flight. Additionally, the model shows that pilot's have stored 'referenced information', 'information structures' and 'information strategies', which are regularly used and evolve. In related work (Solodilova, Lintern & Johnson, 2005) we show that these references, structures and strategies are poorly supported in current displays and consequently can be a source for pilots' confusion and misunderstanding of automation. In a previous paper (Solodilova & Johnson, 2004) we show that the use of references, structures and strategies to inform design can produce more efficient displays, where pilots perform twice as fast and with less error.
A further conclusion from this data analysis is that pilots already have existing information structures and pieces of information that are significant to them. We need to use the results of this study indicating how pilots use the information, structures and strategies to the advantage of pilots and to help design engineers in the design of information space in the glass cockpit.
The information layout of new glass cockpit interfaces should support 'referenced information', 'information structures', and 'information strategies' that evolved for over a century. The substantial amount of references and structures that pilots use, learned since they started to fly, should be the main source inspiration. Instead of inventing new ways of presenting information Pilot's own information use strategies should be supported and used.

Acknowledgments

This work is being funded by the EPSRC (grant number GR/R40739/01) and supported by QinetiQ and Westland Helicopters. We would like to thank the pilots, who participated in the flight observations study and in particular Squadron Leader Mark Thoresen of 85WG RAAF Base Richmond, NSW, Australia, for organizing and making this research possible.

References

Cantor, D., Andreassen, C. & Waters, H. (1985). Organization in visual episodic memory: Relationships between verbalised knowledge, strategy use, and performance, Journal of Experimental Psychology, 40:218-232.

Demagalski, J., Harris, D., Salmon, P., Stanton, N. A., Young, M. S., Marshall, A., Waldman, T., & Dekker, S. W. A. (2002). Design induced errors on the modern flight deck during approach and landing. In: Proceedings of HCI-Aero 2002, MIT, Cambridge, MA, pp.173-178.

Heymann, M. & A. Degani (2002). Constructing Human-Automation Interfaces: A Formal Approach. In: Proceedings of the International Conference on Human Computer Interaction in Aeronautics, Cambridge, Massachusetts, AAAI Press, California, USA, pp.119-124

McLennan, J., M. Omodei, M. & Rich, D. (1997). Using a head-mounting video camera system and two-stage debriefing procedure: A new approach to advances skills training in industry. Swinburne University of Technology Institute of Social Research, Melbourne, Australia.

McLennan, J., M. Omodei, M., & Wearing, A. (2000). Cognitive Processes of First-on-Scene Fire Officers in Command at Emergency Incidents as an Analogue of Small-Unit Command in Peace Support Operations. 2nd International Workshop on The Human in Command: Peace Support Operations, Breda, The Netherlands.

Omodei, M., Wearing, A., & McLennan, J. (1998). Integrating Field Observation and Computer Simulation to Analyse Command Behaviour. In: Proceedings of the NATO RTO Workshop on The Human in Command, June 1998, Kingston, Canada.

Neisser, U. (1976). Cognition and Reality: Principles and Implication of the Cognitive Psychology, Freeman, San Francisco, USA.

Newman, R. L. & Greeley, K. W. (2001). Cockpit Displays: Test and Evaluation. Ashgate, Aldershot.

Omodei, M., Wearing, A., & McLennan, J. (1997). Head-mounted video recording: A methodology for studying naturalistic decision making. In R. Flin, E. Salas, M. Strub and L. Martin (Eds.) Decision Making Under Stress: Emerging Themes and Applications. (pp. 137-146) Ashgate, Aldershot.

Sarter, N. B. & Woods, D. D. (1994). Pilot Interaction with cockpit Automation II: An Experimental Study of Pilots' Model and Awareness of the Flight Management System. The International Journal of Aviation Psychology, 4 (1):1-28.

Sarter, N. B. & D. D. Woods (1995). How in the world did we get into that mode? Mode error and awareness in supervisory control. Human Factors, 37 (1):5-19.

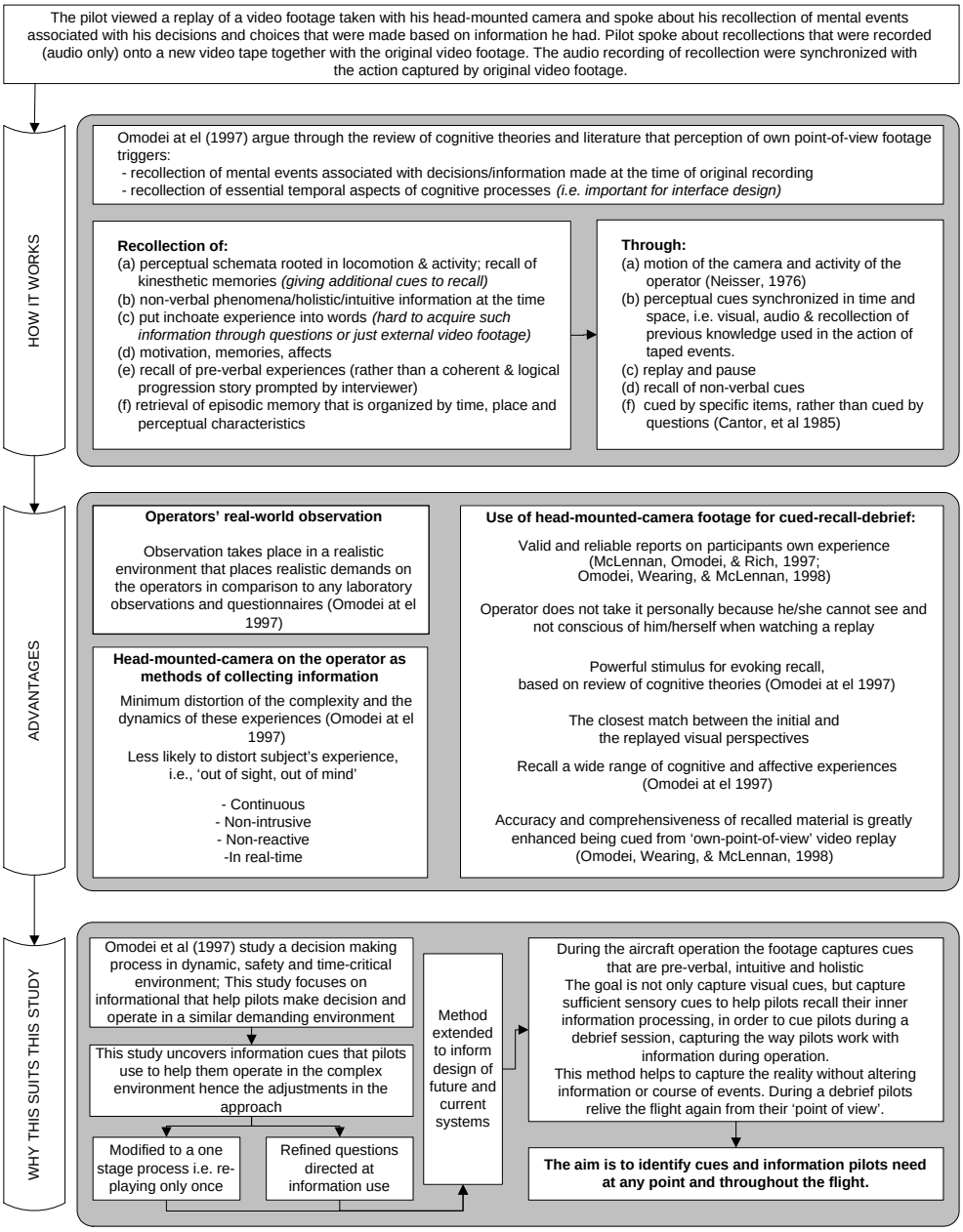
Solodilova, I., Lintern G., & Johnston, N. (2003) The Modern Commercial Cockpit As a Multi-Dimensional, Information-Action Workspace. In: Proceedings of the 12th International Symposium on Aviation Psychology, 14-17 April 2003, Dayton, Ohio, USA, pp.1096-1101.

Solodilova, I., & Johnson, P. (2004) 'Mind References' in the Glass Cockpit: An Experimental Study. In: Proceedings of the HCI Aero 2004, Toulouse, France, [CD-ROM].

Solodilova, I., Lintern G., & Johnson, N. (2005) A Mind-Reference Framework for design and evaluation of intuitive and natural interfaces. In: Proceedings of the 13th International Symposium on Aviation Psychology, 18-21 April 2005, Oklahoma City, Oklahoma, USA. In press

Apppendix

From original two-stage to a modified cued-recall-debrief procedure with the use of head-mounted camera on the operator



Validating a Process for Understanding Human Error Probabilities in Complex Human Computer Interfaces

Richard Maguire

SE Validation Ltd, 10 College Street, Salisbury, Wiltshire
rlm@sevalidation.com

Abstract: It does seem as though each new generation of equipment becomes increasing complex to operate, understand and interface with. A point agreed to by many of my colleagues who happen to occasionally hire the latest car and have to spend the first hour sat in it trying to tune in radio 2. In defence equipment the drive for better technical and operational capability places a new burden on the operators – as the complexity of the machine increases, there is greater potential for the human to make a significant contribution to that required capability, but also to unwanted irregularities, incidents and even accidents.

Two years ago I led a small team working on understanding human error rates for military aircrew in a new glass cockpit environment. During the research, a method for quantifying the human un-reliability was proposed and demonstrated. This paper now presents the results of a validation exercise undertaken on those derived figures.

Keywords: HEART, Human Error Assessment, Hierarchical Task Analysis, Fault-tree analysis, Human Computer Interfaces

Introduction:

Since error is endemic in our species [as described by Kirwan (1994)], there are really only two alternatives for modern society; either remove error prone systems completely, or try to understand them better and so minimise the error problems as far as reasonable practicable. Providing there remains a need for complex activities such as air travel and air defence, the first option will not be acceptable, so this limits and forces us to the latter alternative – understanding and mitigation.

In the field of military (all services) rotary wing aircraft, the UK accident rate for 1991 to 2000 averages at around 28 accidents per 100,000 flying hours (UK Inspectorate of flight safety). By comparison, the UK civilian rate for the year 2000 for all aircraft types was around just 6.3 accidents per 100,000 flying hours (Lawrence 2001). In the US Army (not all services) over the period 1987 to 1995 there were nearly 1000 rotary wing accidents costing some $96M and 200 fatalities (Braithwait 1998). These numbers indicate that understanding aircrew error, particularly in rotary wing aircraft, is of significant importance.

The rest of this paper is organised as follows. The concept of identification and quantification of human reliability is summarised; the original research is scoped and presented with the derived results; the validation task is then discussed and finally the validation results are recorded and commented upon.

Quantification of human reliability

The quantification of human reliability is based on having statistically relevant data of human tasks and the associated error rates. Any similar study could refer to the databases and call off the required values and have data that was reasonably fit for purpose. The basic problem with quantitative methods is a lack of data to form the foundation for the assignment of human probabilities to individual task elements. Given that underlying databases are incomplete, experts are asked to provide data that the databases cannot provide (Nagy 2002). This then, leads to a combination of subject matter expert opinion and quantitative analysis supplementing each other, which is open to criticism, argument and may not even be repeatable without careful recording of the expert's demographics. Conventional human reliability analyses are useful in the case of routine highly skilled activities, in the sense that humans may be said to behave very much like machines (Nagy 2002). There is not the need for deep thought, consideration and interpretation of the

operating environment. Simple human error analysis methods can certainly be adequate. Increasing complexity of the environment and the human task however, does need a more demanding assessment technique with subsequent validation. Kirwan (1994) suggests a three step method for understanding and decreasing human error, his steps are;

1.1.1 Identifying what errors might occur
1.1.2 Quantifying the likelihood of occurrence
1.1.3 Reducing the error likelihood

Classical risk analysis, as Kirwan records elsewhere, would also establish the severity of the occurrence, and also seek to reduce the impact. But as the method is specific to the subject of human error *occurrence*, it is perfectly acceptable.

Human error identification techniques are numerous, and there are many papers on each technique. As recorded by Wiegmann, Rich and Shappell (2000), Kirwan (1998) describes thirty-eight approaches for error identification. They are categorised by type of approach and are critiqued using a range of assessment criteria. Five broad classifications are developed; taxonomies, psychologically based, cognitive modelling, cognitive simulations and reliability oriented. Several analytical-method classifications are also derived; check-lists, flowcharts, group-based, psychological, representation, cognitive, task analysis, affordance-based, commission identification and crew interactions. The paper does not recommend any single technique, but rather suggests that it is a combinations of techniques and analytical methods that is required.

Similarly, there are multiple quantification techniques. Quantification has always been a thorny issue, and will likely remain so for many years to come. Some behavioural scientists have argued - at times very forcefully - that quantification in principle is impossible (Hollnagel, 2005). This may be true for a specific forecasted tasks with the obvious lack of a statistical-based referent. However, systematic tasks that are required to be regularly done by multiple persons, and which may be reasonably compared to statistically relevant historical data, will give usefully reasonable results, where none perhaps existed before. Consistently exact values of human error rates to three or four significant figures (as may be available for material failures), is currently just not possible, human behaviour is not that regular. Often however, that is not the principle requirement. This may be more on the lines of getting data that is useful for the purpose i.e. for comparative purposes, or simply to determine values to better than a one significant figure 'guestimate'.

There are occassions where quantification simply has to be done, i.e. when it has been deemed a formal requirement, and you've actually got to do it, for whatever reason. Several notable quantitative techniques are well documented in literature SHERPA, HEART and THERP, for reasons of brevity, this paper will only provide a summary of these.

SHERPA (Systematic Human Error Reduction & Prediction Approach) (Stanton & Wilson, 2000). SHERPA works rather like a human based HAZOP. Each task is classified into one of five basic types (checking, selection, action, communication and information retrieval) and then a taxonomy of error types are applied. For each error type an assessment of likelihood and criticality is made. The analysis can be summarised into a classic risk prioritised format, with a quantitative value being assigned to each task with human error. So whilst there are values with external utility, some quantification is done internally and it may be extended via expert opinion to an external temporal reference.

HEART (Human Error Assessment & Reduction Technique) (Maguire & Simpson, 2003) The HEART method involves a classification of identified tasks into proscribed groups from a look-up table, which leads to a nominal human error probability (HEP). Obvious error-producing conditions are applied to the task scenario under investigation in the form of multiplying value, and these values may be themselves factored according to the scenario. The combination of nominal HEP, error producing conditions and factoring ultimately lead to a final HEP value. Expert opinion is used to validate the selection of the task grouping and the error producing conditions.

THERP (Technique for Human Error Rate Prediction) (Nagy 2002): The THERP approach consists largely of a database of probabilities of different kinds of human error, together with performance shaping factors. The analysis starts with a task analysis, graphically represented as event trees. Event trees are structures with logical operators that are used to consider the different potential outcomes of some initiating fault or failure. Human activities are broken down into task elements, which when considered to fail, become the initiating faults. Performance shaping factors such as stress or time are used to modify the probabilities according to expert judgement. The modified result is an estimate of the likelihood of a particular task being carried out in error.

The initial research: Many aircraft have safety models with calls for human reliability to show an overall probability of a catastrophic event i.e. fatality or aircraft loss. The initial research was designed to assist in populating safety models with appropriate values. It was published two years ago (Maguire & Simpson, 2003) – a brief resume of the scope, methodology and results is probably required for this paper. A specific aircraft type was not specifically defined other than being a rotary wing machine – although, this in no way limits the use of the methodology. The humans under analysis were aircrew undergoing conversion to type training on the aircraft i.e. they are already pilots and are re-focussing on a new aircraft type. An arbitrary but typical mission profile was specified using subject matter experts from the Empire Test Pilot School at MoD Boscombe Down and the Army Training School at MoD Middle Wallop. The developed scenario was of a training pilot carrying out night flying, over undulating terrain, with tree hazards present and flying as one of a pair. As the flight was for a training purpose, the mission had a duration of two hours and was undertaken in good weather before midnight.

A brief overview of historical accident records (Greenhalgh 1999) indicated three flight phases were particularly prone to human error incidents – low-level transit flying, operating at the hover and at landing. These are also considered to be the flight phases where the constructed safety models could get most benefit. This paper will only consider the landing tasks in detail, serving as a demonstration of the original methodology and the validation task.

Following the guidance from Kirwan (1994), the first phase was to identify what errors might occur. This was done using Hierarchical Task Analysis (HTA) and constructing critical task lists. This was done using a combination of techniques – test-pilot interview, procedural analysis and goal analysis. Fortunately, there was a rich source of flight video data to review, and commentary from the corresponding pilot provided excellent information. A typical derived task hierarchy is shown in Figure 1.

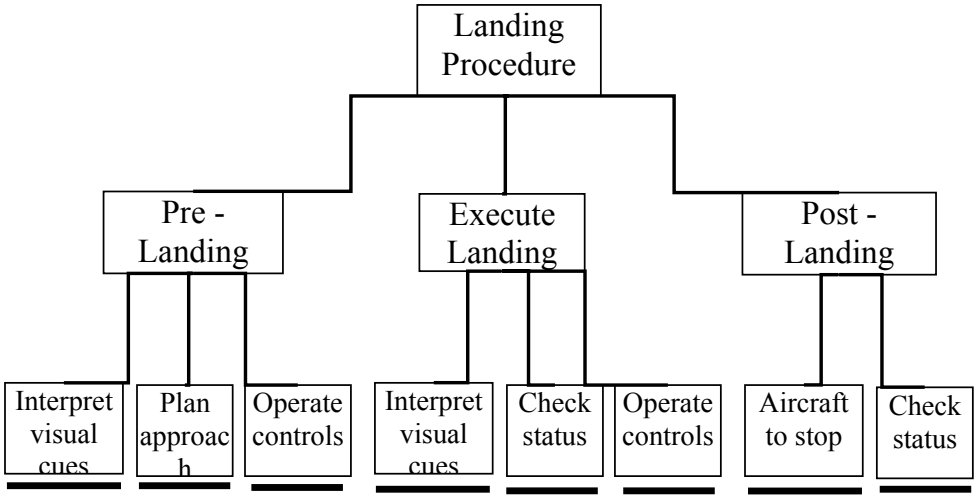


Figure 8Typical task hierarchy segment

The HEART method was utilised for the quantification process. The availability of rich scenario based information, and the need for a faster, cheaper and better response led to this selection. Even in retrospect, this decision has held up well. The HEART method gave a satisfactory set of results, the key stages in their development are shown in Tables 1 to 3.

The next part of the original process was to utilise the task hierarchy to develop a logically operable fault-tree structure for each task segment (Maguire & Simpson 2003). The attempts at these structures led to an increase in required detail being highlighted. For example, the operation of executing the landing had three human tasks initially. The inclusion of some identified crew-resource-management techniques not listed in the flight reference cards or procedures, meant that the extra routine, highly practised, rapid tasks of 'self-check' and 'crew-check' were allowed in the fault-trees. This collaboration between the crew members was shown to reduce the error potential by a full order, encouragement of developing such techniques and collaboration was made in the original research recommendations. The constructed fault-tree was then populated with the values from the HEART analysis. This is shown in this paper in figure 2. A summary of the results from that initial research are presented below, these serves as the object data for the validation task.

Landing phase completed with human errors	5.4e-2 per task
Transit flying phase completed with human errors	2.6e-2 per task
Actions from the hover completed with human errors	3.9e-2 per task

It should be noted that these values do not indicate the frequency of accidents and crashes, but rather the frequency of human errors during these flight phases. Of course the errors may propagate on to accidents, some may be incidents, probably the majority will be just irregularities, which may or may not be officially recorded.

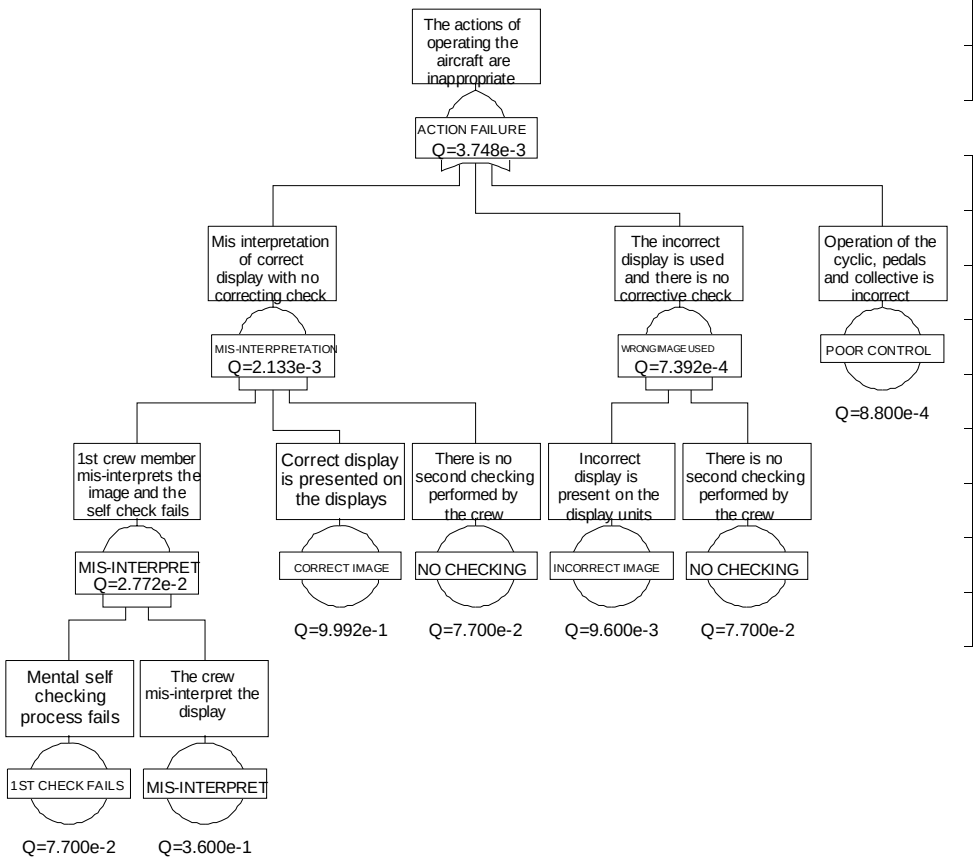
Task	***Description***	***HEART class***	***5th percentile nominal unreliability (per task call)***
Interpret visual cues	Complex task requiring high level of comprehension and skill	C	0.120
Plan approach	Routine, highly practised, rapid task not involving a high skill level	E	0.007
Operate controls	Completely familiar, well-designed highly practised routine task, performed several times per hour by highly motivated people who are totally aware of the action implications	G	0.00008
Interpret visual cues	Complex task requiring high level of comprehension and skill	C	0.120
Check status	Routine, highly practised, rapid task not involving a high skill level	E	0.007
Operate controls	Completely familiar, well-designed highly practised routine task, performed several times per hour by highly motivated people who are totally aware of the action implications	G	0.00008

Task	*Description*	*HEART class*	*5th percentile nominal unreliability (per task call)*
Aircraft to stop	Completely familiar, well-designed highly practised routine task, performed several times per hour by highly motivated people who are totally aware of the action implications	G	0.00008
Check status	Routine, highly practised, rapid task not involving a high skill level	E	0.007

Table 1 Landing task classification

Task	*Assigned Error producing conditions*	*Error multiplier effect*
Interpret visual cues	Operator inexperience = 3	3
Plan approach	Shortage of time = 11	11
Operate controls	Operator inexperience = 3	3
Interpret visual cues	Operator inexperience = 3	3

Figure 9 Example of developed fault-tree



The Validation Task

The information derived from the HEART analysis was for a customer, and that customer wanted to be sure that the information presented was valid and useful. A secondary research task was given to undertake a validation exercise to prove, or not, the accuracy of the original research. Comprehensive validation efforts have taken place on the HEART method along with a comparison of other human error quantification techniques . This validation exercise involved 30 UK based assessors using the three quantification techniques (10 each) HEART, THERP and JHEDI, to review 30 nuclear power plant tasks to determine the HEP (known to the authors). The results for all three techniques were positive in terms of significant correlations. It was found that 72% of all HEP estimates were within a factor of 10 of the true values, and these results lend support to the empirical validity of the techniques, and to human reliability assessment in general (Kirwan et al 1997).

A similar validation task for aircrew tasks has not been undertaken, so it is worthy from a scientific point of view (as well as a customer's) to carry out a dedicated validation exercise for the HEART results developed in the earlier research (Maguire and Simpson 2003).

Raw data for aircrew un-reliability in a typical glass cockpit-based rotary wing aircraft was available from the US Army Safety Center database as reported by Greenhalgh (1999). This data set gave 1370 recorded night flying events (to match the scenario description of the original research). A cut of the data was taken to give a smaller data set from which to derive a referent for the validation. This gave 235 records for the period October 1989 to October 1991. Analysis of the records gave the breakdown shown in Table 4.

Flight phase	***No. of incidents***	***Human error attributes***
Landing	29	15
Hover	37	9
Transit flying	49	6
Other phases (e.g. roll out)	120	5

Table 4 Breakdown of rotary wing recorded event data

The original study (Maguire & Simpson, 2003) derived values with units of '*per task*' and it is perfectly possible to establish similar units for the actual values based on the data in Table 2 in combination with information and expert opinion on the demographics of the flights that led to the accident data. UK and US experts have given the following estimated information, and it is not anticipated that these values are very wide of the real values.

The total number of night time sorties can be determined from the data in Table 2, by the equation;

$$(a \times b) / (d / 60) = 3000 \text{ sorties per year}$$

as the data set is over two years = 6000 total sorties

Information Items	***Value***
(a) Annual flight hours for the fleet	15,000 hours

Information Items	*Value*
(b) Proportion of flt hours as night flying	40%
(c) Night time sortie duration	120 minutes
(d) Number of task calls for landings per flight	3 landings
(e) Number of task calls for hovering	5 hovers
(f) Number of task calls for transit flying	5 transits

Table 5 Summary of flight data demographics

Combining the information in Tables 4 and 5 with the calculated number of sorties, gives a series of values for the nominal human error rates in the three flight phases. This is shown in Table 6.

	Landing	*Hover*	*Transit*
Number of sorties (derived as above)	6000 in total over two years		
Number of task calls per sortie	3	5	5
Number of task calls over two years	18,000	30,000	30,000
Number of recorded human errors	15	9	6
Nominal human error rate	8.3E-004	3.0E-004	2.0E-004

Table 6 Calculated human error rates

Whilst these data items appear to be fully appropriate for the validation exercise, they are limited in their completeness. These values represent the officially recorded data, the original research derived data were for the occurrence of human errors not aircraft accidents. This referent data does need to be supplemented to complete the range of human errors, not just those which are cited in accident reports.

There is an accepted relationship between accident rates, incident rates and irregularity rates. It is known by several terms – The Iceberg of Incidents (Davis 2002) and Accident Ratios (The Engineering Council 1993), and essentially it describes the relationship between major, minor and no-effect events. The ratio between these factors is quoted as 1 : 10 : 600.

The 30 human error attributed events from the data set can be arranged in the three categories to check the ratio, as recorded. This arrangement is presented in Table 7.

Flight phase	*Major*	*Minor*	*No effect*
Landing	3	2	10

Flight phase	*Major*	*Minor*	*No effect*
Hover	1	4	4
Transit	3	1	2
Iceberg ratio	1	10	600

Table 7 Comparison of accident severity ratios with Ice-Berg effect

The no-effect category is far too under populated, they appear to have been un-recorded by a factor of around 100 or so in each flight phase. Research cited in Davis (2002) and The Engineering Council (1993) indicate that these no-effect events do take place, but they are left unrecorded due to embarrassment, doctrine or an opinion that these events do not matter.

Supplementing the recorded data with the expected full data set related to the well recorded major events, gives new figures as the referents for the validation exercise.

Flight phase	*Proposed figures from HEART method*	*Figures from referent source*
Landing	5.4 E-02	8.3 E-02
Hover	3.9 E-02	3.0 E-02
Transit	2.6 E-02	2.0 E-02

Table 8 Comparison of HEART derived data and validation referent

By way of comparison with the Kirwan led validation exercises (Kirwan 1996; Kirwan et al 1997; Kirwan 1997), the proposed human error probabilities are likewise with-in a factor of 10 of the referent data. This does lend support to the empirical validity of the original research methodology of overarching task decomposition, fault-tree derivation and HEART quantification. I understand that the customer is satisfied with his human reliability data for his safety models.

Discussion

Although the method appears quite sound, a number of limitations need to be acknowledged before the data may be used. The experts who helped with the original research were UK based and so gave UK opinion on the task hierarchy breakdown. The referent information was from US sources so the differences in approach to flight safety, crew resource management and event recording is likely to be different. It remains unclear as to how much this has affected the results.

The availability of accurate flight demographics is a concern, although even if these values have error bands of +/- 50%, the end comparison is still within the same order of magnitude. A similar case has to be accepted for the quantity of no-effect events that are added back into the referent data set, which due to their size, obviously swamp the more severe putcome events.

However, a validation exercise has been carried out. The referent used for this exercise may be considered reasonable. The comparison between the proposed values and the referent has been shown to be satisfactory, and hence the method and data set derived may be considered fit for the prupose of better understanding human errors.

References

Braithwait Col. M 1998 'Spatial disorientation in US Army rotary wing operations' –. Aviation, space and environmental medicine Vol 69, No 11.
Davis R 2002 'An introduction to system safety management and assurance' - MoD ALTG-LSSO, MoD Abbey Wood, Bristol
Greenhalgh Lt Col J.G. 1999 'BLO aviation report 99/14 BAS(W) tasking matrix 96/1/092 TADS/PNVS FLIR quality – data from US Army Safety Center database Oct 89 to Mar 99' Fort Rucker, USA.
Hollnagel E 2005 'Human reliability analysis' – website based notes
Kirwan B 1994 'A guide to practical human reliability assessment' – Taylor and Francis.
Kirwan B 1996 'The validation of three human reliability quantification techniques – THERP, HEART and JHEDI – part I – Techniques and descriptions' Applied Ergonomics v27/6.
Kirwan B, Kennedy R, Taylor-Adams S & Lambert B 1997 'The validation of three human reliability quantification techniques – THERP, HEART and JHEDI – part II – Results of the validation exercise' v28/1
Kirwan B 1997'The validation of three human reliability quantification techniques – THERP, HEART and JHEDI – part III – Practical aspects of the usage of the techniques' Applied Ergonomics v28/1
Kirwan B 1998 'Human error identification techniques for risk assessment of high risk systems, part 1 : review and evaluation of techniques.' Applied Ergonomics v29/3
UK Inspectorate of Flight Safety figures 1991 to 2000.
Lawrence P 2001'Human factors in aerospace design for operations' –. University of the West of England.
Maguire R. L. & Simpson Dr A. 2003 'Quantification of military helicopter aircrew human error probabilities for safety models' MoD Equipment Safety Assurance Symposium 2003
Nagy G 2002 'Human reliability analysis : From action to context' – Carleton University.
Stanton Prof. N & Wilson J.A. 2000 'Human factors: step change improvement in effectiveness and safety' Drilling Contractor, Jan/Feb 2000.
Wiegmann D A, Rich A. M & Shappell S. A. 2000 'Human error and accident causation theories, frameworks and analytical techniques : an annotated bibliography' – Aviation Research Lab, University of Illinois

The Design of Complete Systems: Developing Human Factors Guidance for COTS Acquisition

Anne Bruseberg,

Systems Engineering and Assessment Ltd. Somerset, UK

Abstract: In this paper, we describe challenges and approaches to provide Human Factors support for the acquisition of COTS (commercial off-the-shelf) equipment as part of the MoD acquisition and development processes. Whilst the use of COTS generally prohibits influence on the design of the technical product since it is already completed, the design of the socio-technical system into which the COTS product needs to be integrated can be a significant source of complexity. This is especially so when dealing with Human Factors aspects, which are less tangible than engineering issues, and therefore more difficult to capture in requirements specifications and assessment metrics. In this paper, we describe challenges and solution approaches for Human Factors guidance on COTS selection, as part of the work on the Human Factors Integration for the Defence Technology Centres currently carried out at SEA.

Keywords: **MoD acquisition process, Human Factors Integration (HFI), COTS, guidance..**

Sources of Complexity in the Design of Healthcare Systems: Autonomy vs. Governance

A. Taleb-Bendiab, David England, Martin Randles, Phil Miseldine, Karen Murphy

School of Computing and Mathematical Sciences, Liverpool John Moores University, Byrom St, Liverpool, L3 3AF

Contact email: d.england@livjm.ac.uk
http://www.cms.livjm.ac.uk/2nrich

Abstract: In this paper we look at decision support for post-operative breast cancer care. Our main concerns are to support autonomy of decision making whilst maintaining the governance and reliability of the decision making process. We describe the context of our work in the wider medical setting. We then present a set of decision support tools based on the situation calculus as a means of maintaining the integrity of rule bases underlying the decision making system

Keywords: decision support, breast cancer care, autonomy, self-governing systems

Introduction

Our project on decision support for post-operative breast cancer care raises a number of interdisciplinary questions in a complex and emotive area. The project is collaboration between computer scientists, statisticians and clinicians which itself is a complex arrangement. The nature of the subject involves life-threatening decisions. Clinicians and patients are faced with the most difficult decisions. From an HCI viewpoint we are also faced with supporting and not supplanting clinician's judgments. There are also wider issues of the implications of our studies of historical clinical data and what that might mean for future approaches to prognosis. In an earlier paper we discussed our approach to process understanding in breast cancer care. We described how decisions on post-operative breast cancer treatment are currently governed by a set of medical guidelines including the National Institute for Clinical Evidence (NICE) guidelines (Nice 2005) in which the decision process is as follows: the clinician uses available data with a staging method to define the patient risk category. The risk category is then used to determine the type of treatment that the patient would be offered. The guidelines approach has been investigated by Woolf (Woolf 1999) who concluded that guidelines can cause benefit or harm and that the clinical guidelines need to be "*Rigorously developed evidence based guidelines [to] minimize the potential harms*" .

The aim of our project is three-fold.

1. To understand and support the clinician's decision making process within a set (or sets) of clinical guidelines;
2. To analyze historical data of breast cancer care to produce new rules for treatment choice.
3. To bring these two processes together so that treatment decisions can be guided by an evidence-based computer system.

The evidence-based approach to the delivery of medical care has gained wide recognition within the healthcare community, advocating that decision-making should use current knowledge and clinical evidence from systematic research (Rosenberg 1995). In breast cancer care, there are currently a few staging methods in widespread use by clinicians, namely the Nottingham and Manchester staging systems. However, there is no standard method to support oncologists' decision-making processes as to how and when to include new evidence, and how to validate emerging local patient data patterns or other models and practices from elsewhere. This means that there is no standard way to ensure that clinicians are following accepted guidelines or deviating from them. There may be valid clinical reasons why a standard decision path is not chosen (e.g. the age or infirmity of the patient) but these decisions are not currently recorded in a standard way.

In this paper we wish to address some of the sources of complexity in the design of healthcare systems. More specifically we are interested in safe and reliable decision support for post-operative breast cancer care and the wider lessons we can learn from our experiences. There are many contributory factors to the complexity of designing these systems, some of which we shall discuss below. These include:

- Environment and Context: The aims of national initiatives such as NPFit/PACIT in driving down costs and errors etc.
- The Autonomy of clinicians versus the governance requirements of clinical audit
- Resource limitations – staffing drugs, radiology etc
- The limitations of Medical Knowledge and how that knowledge evolves
- The co-evolution of knowledge and the needs for system validation and safety
- The Requirements for safe solutions and effective treatment
- The (moving) Requirements for knowledge elicitation
- The abilities of Computer Scientists to encode medical knowledge
- The limitations of data mining approaches as a means to supporting evidence-based, decision making.

We will concentrate on our approach to tool support of the decision complexities using the situation calculus (Reiter 1991).

National Programmes

In both the UK and US there are national initiatives to introduce greater use of IT in clinical settings. The broad aims of the NPFit (UK) and PACIT (USA) programmes are similar. They aim to streamline data processing to cut costs and reduce clinical errors. For example, it is proposed that electronic prescribing of medicines will cut costs in paperwork and reduce prescribing errors which account for a large number of patient deaths (44,000 to 98,000 deaths caused by medical errors in the USA). Both schemes aim to introduce electronic patient records, again to cut costs of paper records and reduce errors from paper-based systems. Both systems also look to more clinical governance and audit of medical processes so that medical staff are more accountable for their actions. The UK initiative is already displaying the signs of a large project out of control with the projected costs of £6Bn rising to between £18Bn and £31Bn. The lack of user centred design is evident by a recent (BBC) poll showing 75% of family doctors are not certain that NPFit will ever meets its goals. The first stage of the electronic appointment systems has largely failed to meets its use targets. However, a smaller scale introduction of region-wide IT in the Wirral was more widely accepted with 90% of family surgeries and the vast number of patients accepting the system. Thus IT systems can succeed. This is important for our work, for in order to succeed, it requires a working IT health infrastructure. Furthermore the twin goals of cost and error reduction may be mutually incompatible. As Reason points out (Reason 1997) organisations have processes for productivity and safety but circumstances will arise, either through unsafe acts or latent system weaknesses, which lead to organisational failure. Safety protocols may be violated in the name of efficiency or sets of latent weaknesses will line up to cause an accident. Many individual errors are the result of cognitive under-specification (Reason 1990) of the user's tasks. In our project we aim to over-specify and support clinical tasks by describing them in the situation calculus. This will provide a robust means of supporting decision making and ensuring that chances to decisions protocols remain valid.

Medical Knowledge and Computer Science

In a multidisciplinary project settings involving; clinicians, statisticians, computer scientists and public health specialists, our project has started by understanding the current decision-making practices as a prelude to systems' implementations. This will be evaluated using a set of small-scale controlled trials involving both patients and clinicians. The proposed method, unlike traditional decision-making techniques, including multi-criteria, will provide breast cancer clinicians and patients with a flexible decision framework adaptive to their decision practices. It will also allow for evolutions of decision models, decision resources (data) and users concerns. This novel approach will provide important insights into the development of an integrated decision support infrastructure for high assurance decision activities,

which will directly contribute to one of the NHS R&D high priority area of "Medical Devices Directives for cancer patient care".

To model, validate and enact clinical guidelines a new approach, using ideas originating from research in distributed artificial intelligence, has been developed. In this formalisation the treatment recommendation process is conceived of as a human/artificial multi-agent system. The actions of the agents are governed by system norms. These norms constrain the behaviour of the actors in the system. They may relate to the control of the system in maintaining availability, fault tolerance, adaptability etc. for quality of service or they may be concerned with the output results such as guideline compliance for quality of process. In any event the goal is complete system governance within safe limits, encompassing clinical governance. It is a complex process, in itself, to produce safety critical systems. However in needing to allow for the variability and variety of clinicians' usage adds another level of complexity. The autonomy that needs to be retained by the clinician is tempered by the constraints of clinical governance. Thus if the system is required to take control of management functions the autonomy of the system's agents is compromised. However the agents are autonomous, rational and social so computational economy is best served by the agents following the norms. Norms arise from a social situation in that they involve providing rules for a community of more than one individual (Boman 1999). The norms can be associated with producing decisions, as a practical application, or with the governance of the system and clinical processes. The provision of adjustable autonomous agent behaviour is necessary to deliver the system services in a way where the majority of the administrative tasks are handled within the software itself whilst the autonomy is reconciled with the governance of the system processes. This is achieved by taking an aspect oriented approach and separating the concerns of governance from the application service concerns.

In order to logically model and reason, within this approach, the formalism of the Situation Calculus (McCarthy 1968) was used. The Situation Calculus is especially suited to the analysis of dynamic systems because there is no need for the prior enumeration of the state space so unexpected external events can be accommodated. What-if scenarios can be modelled with branching time-lines and counterfactual reasoning. Thus a formalism is provided to model both system behaviour, in mapping a set of symptoms to a set of treatment options and deliberative governance strictures so that formal reasoning techniques, such as deduction, induction or abduction, can be applied to analyse the meta-control of the system.

Although a full explanation of the Situation Calculus is outside the scope of this paper a full specification can be found in Levesque 1998. Briefly stated, the Situation Calculus is based on actions. A situation is an action history emanating from an initial situation, defined by a set of function values called fluents. Successive situations are defined by effect axioms, for the action, on the fluents. Successor state axioms together with action precondition axioms give a complete representation that mostly solves the frame problem (Reiter 1991).

A custom scripting language, JBel (Miseldine, 2004) has been developed to facilitate the deployment of, the generated, Situation Calculus defined, self-governing decision agent in a grid based architecture – Clouds (figure 1). The Clouds concept is conceived as a system with fluid and flexible boundaries that can interact and merge with similar system architectures.

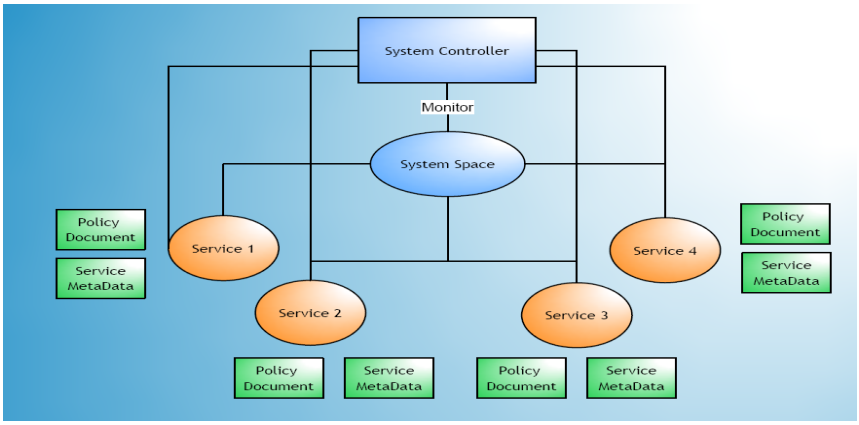


Figure 1. Clouds Architecture

The Cloud can be thought of as a federation of services (agents) and resources controlled by the system controller and discovered through the system space. The system space provides persistent data storage for service registration and state information giving the means to coordinate the application service activities. The system controller controls access to and from the individual services and resources, within a Cloud. It brokers requests to the system based on system status and governance rules, in JBel objects, derived from the logical normative deliberative process. The system controller acts as an interface to the Cloud. It can function in two roles, either as an abstraction that inspects calls between the System Space and services, or as a monitor that analyses the state information stored within the system space.

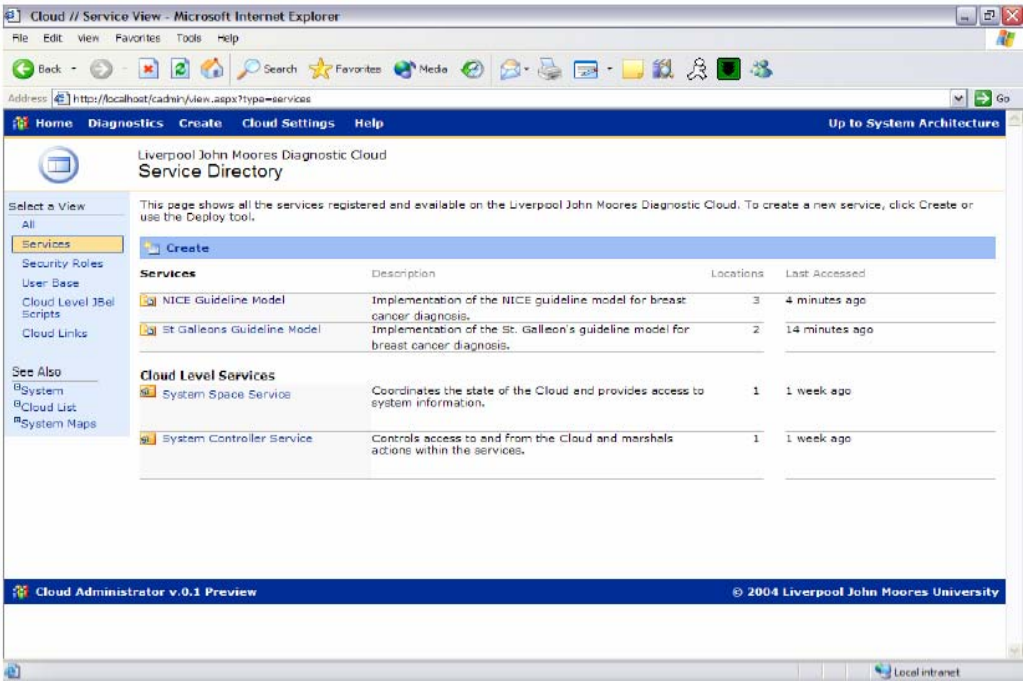


Figure 2. Clouds Decision Rule Governance Tool

In the current research prototype the Cloud environment (figure 2) produces an interactive dialogue in which the clinician enters the clinical data and is given one or more recommended treatments. At this stage of development we are looking at two approaches to critiquing the recommended outcomes. Firstly, by using a critiquing user interface (Gray 1996) where the clinician enters the data and the proposed treatment and the system returns with a recommended treatment. Secondly, where "borderline" or difficult treatment cases are explored using more than one set of clinical guidelines to see if they converge or diverge in their recommendations. This is a similar approach to voting systems used in safety critical systems (Mackie 2000). Critiquing at the user interface involves guidance by one knowledge base whereas the second form of critiquing requires the use of multiple *and* different knowledge bases.

The development of adaptive software, in healthcare systems, relies on the specification and abstraction of norms that collectively constitute a decision process as expressed logically through Situation Calculus. In addition, to allow safe re-engineering in the distributed context, the representation of such norms must be designed so that they can be adapted at runtime without alteration to their published interface signature to maintain modularity.

The methodology, used in JBel, to express behavioural rules within an introspective framework at runtime is structured in terms of conditional statements, and variable assignments which are parsed and evaluated from script form into object form, making it suitable for representing Situation Calculus rules. The resulting JBel object encapsulates the logic and assignments expressed within the script in object notation, allowing it to be inspected, modified, recompiled, and re-evaluated at runtime.

Object serialisation techniques, allowing an object to be stored, transferred and retrieved from information repositories, can be applied to the JBel objects, allowing the safe and efficient distribution of decision models.

The design of decision processes involves a process workflow and behavioural rules that determine the movement through the process within the model. Decisions within the workflow are linked to specific behaviour defined within the JBel script, with the behaviour determining the future flow of the model. Thus, with decision logic encapsulated within a JBel Script, changes to the structure of the model are separated from changes to its logical construction, yielding separate and abstracted decision model

architectures. This high level of runtime accessibility permits the complete separation of the presentation layer with the underlying process model.

Behavioural rules collectively assemble a profile of accepted behaviour for the system they describe by associating logical conditions with a set of actions to take place upon their success or failure. In the context of medical decision support, such profiling allows formal modelling of clinician concerns, introducing high level assurance for their fulfillment.

In conjunction with the rule-based approach we also have a data mining interface which looks at historical patient data. Our statistician colleagues have used a neural networks approach to find patterns in breast cancer survival data (Jarman 2004). The outcomes from this analysis have been fed into a rule induction package to produce a set of human-understandable rules. There are two intended usages of the data-mining interface in Clouds. Firstly, as a support for clinical governance, in that, we can compare the explicit rules from published guidelines with the induced rules and see if the published rules are being followed in practice. Secondly, we can perform "What-if?" analyses so that, within the patient/clinician dialogue about treatment, a doctor can illustrate to the patient that there are other patients in the historical data with a profile similar to theirs. Thus they can discuss alternative treatments in cases where either, the decision system outcomes are ambiguous, or the patient's situation requires a different treatment from that recommended. We are currently modelling the clinical-patient dialogue within Clouds so that we can record alternative outcomes and their reasoning.

As an example from the Situation Calculus a treatment decision may be handled via the successor state axiom:

$$NICEtreatment(patient, tamoxifen, do(a,s)) \Leftrightarrow [\, NICEtreatment(patient, tamoxifen, s) \wedge \neg \exists treatment(\, a = nice\_treament\_decision(patient, treatment) \wedge (\, treatment \neq tamoxifen))] \vee [a = nice\_treatment\_decision(patient, tamoxifen)]$$

with the action precondition axiom:

$$poss(nice\_treatment\_decision(patient, tamoxifen), s) \Rightarrow (oesreceptor(patient, s) = pos) \wedge (menostatus(patient, s) = post)$$

It is then possible to log compliance to provide data for system update and evolution:

$$compliance(patient, treatment, service\_decision, do(a, s)) \Leftrightarrow [\, compliance(patient, treatment, service\_decision, s) \wedge a \neq treament\_decision(patient, treatment1)] \vee [a = treatment\_decision(patient, treatment) \wedge service\text{-}decision(patient, s) = treatment]$$

This is the method by which the system representation and reasoning model can be realised for both self-governance and application level services. The service level concerns, the NICE guidelines in the above formalism, are handled separately from the governance concerns of process quality monitoring.

However in both cases the deliberation is modelled in the Situation Calculus then directly implemented into the JBel scripts (figure 3). The deliberation required for the governance allows the services (agents) to act autonomously within safe limits whilst the deliberation to produce a guideline-based decision is completely specified by the rule base derived from the NICE guidelines.

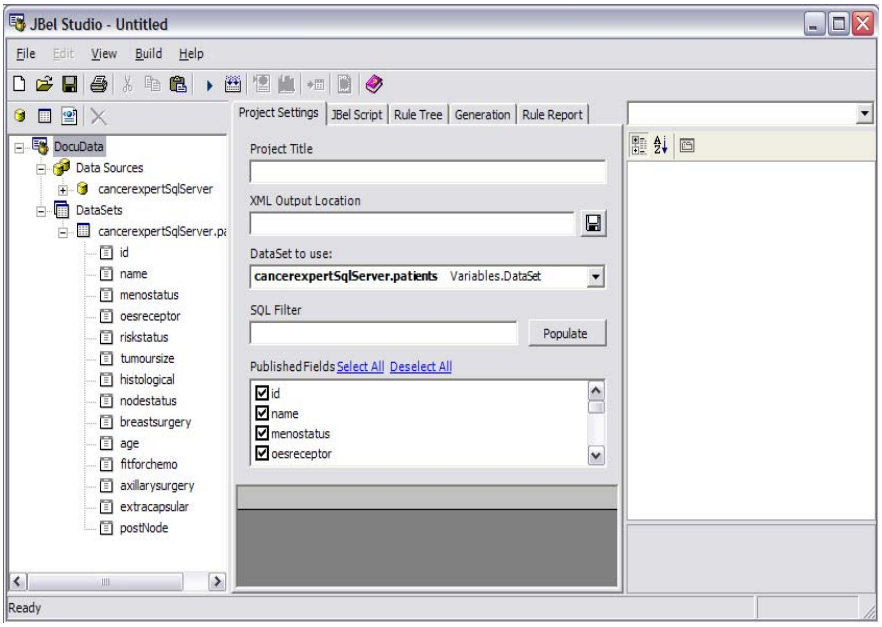


Figure 3 The JBel IDE

Conclusions

There are a number of remaining challenges to the successful deployment of the kind of system we are proposing. Practically there are implementation problems and issues with current access to IT by clinicians and their team members. Some of these issues may be resolved by the NPFit programme (NPFit 2004). However, we have already noted above problems with cost overruns, delays and issues with clinical acceptance of NPFit and its various constituent systems. From a safety viewpoint there are several issues with how we maintain the validity of our rules, whether induced or derived from guidelines, as NICE guidelines are update every five-year. Within the lifetime of the project there have been new medical developments which have been used in clinical trials. We need to ensure that our system continues to give safe recommendations after any rule updated, and the situation calculus can be used to specify and validate the dynamics of such open systems (Randles 2004).
In the wider medical viewpoint we can see applications for our approach in other areas of medicine, such as the diagnosis and treatment of lymphoma which has a similar staging model to breast cancer. We are also exploring the use of self-governing rule sets in Dentistry where the system would be used to module dentistry protocols. More broadly still we are looking at the use of our approach in the general area of self-governing and autonomous systems in a wide range of settings requiring ambient intelligence.

References

Boman M, "Norms in artificial decision making" AI and Law, 1999.

England D A. Taleb-Bendiab, A, Lisboa, P, Murphy, K, Jarman, I, "Decision Support for Post-Operative Breast Cancer Care", Coping with Complexity: Sharing New Approaches for the Design of Human-Computer Systems in Complex Settings, University of Bath, 2004

Gray PD, Draper SW, "A Unified Concept of Style and its Place in User Interface Design", Proceedings of BCS HCI 96 Conference, Cambridge University Press, 1996

Jarman, I, Lisboa, P J, "A comparative study of NPI and PLANN-ARD by prognostic accuracy and treatment allocation for breast cancer patients", ENRICH Deliverables,

Available on: http://www.cms.livjm.ac.uk/2nrich/deliverables2.asp, 2004.

Levesque H. J., Pirri F. and Reiter R. "Foundations for the situation calculus". Linköping Electronic Articles in Computer and Information Science, Vol. 3(1998): nr 18. http://www.ep.liu.se/ea/cis/1998/018/

McCarthy J. and Hayes P. "Some philosophical problems from the standpoint of artificial intelligence". Machine Intelligence 4, ed: B. Meltzer and D. Michie, pp 463-502, Edinburgh University Press, Edinburgh, Scotland 1968

Mackie J, Sommerville I, "Failures of Healthcare Systems", Proceedings of the First Dependability IRC Workshop, Edinburgh, United Kingdom, March 22-23, 2000

Miseldine, P. JBel Application Development Guide, 2nrich Project. http://www.cms.livjm.ac.uk/2nrich/deliverables.asp. 2004

NICE, National Institute for Clinical Excellence, www.nice.org.uk, 2005

NPFit, National Programme for IT in the NHS, www.npfit.nhs.gov.uk, 2004

PITAC - President's Information Technology Advisory Committee, Revolutionizing Health Care Through Information Technology, June 2004

Randles M, Taleb-Bendiab A, Miseldine P, "A Stochastic Situation Calculus Modelling Approach for Autonomic Middleware", Proceedings of PGNet 2004, Liverpool John Moores University, 2004

Reason, J, "Human Error", Cambridge University Press, 1990

Reason J, "Managing the Risks of Organisational Accidents", Ashgate, 1997

Reiter R. "The frame problem in the situation calculus: a simple solution (sometimes) and a complete result for goal regression". Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy, ed: V. Lifschitz, pp359-380, Academic Press, San Diego, California 1991

Rosenberg W, McDonald A, "Evidence based medicine: an approach to clinical problem-solving", BMJ 1995;310:1122-1126

Steve H Woolf, Richard Grol, Allen Hutchinson, Martin Eccles, Jeremy Grimshaw, "Potential benefits, limitations, and harms of clinical guidelines", BMJ 1999;318:527-530

This project is collaboration between Liverpool John Moores University, the Christie Hospital, Manchester and the Linda McCartney Centre of the Royal Liverpool Hospital. It is funded by the EPSRC.

Automation, Interaction, Complexity, and Failure: A Case Study

Robert L Wears, MD, MS and Richard I. Cook, MD*

Dept. of Emergency Medicine, University of Florida, Jacksonville, Florida, USA
and
Clinical Safety Research Unit, Imperial College, London W2 1NY
wears@ufl.edu

*Cognitive Technologies Laboratory, University of Chicago, Chicago, Illinois, USA
ri_cook@uchicago.edu

Abstract: Although proponents of advanced information technology argue that automation can improve the reliability of health care delivery, the results of introducing new technology into complex systems are mixed. The complexity of the health care workplace creates vulnerabilities and problems for system designers. In particular, some forms of failure emerge from the interactions of independently designed and implemented components. We present a case study of such an emergent, unforeseen failure and use it to illustrate some of the problems facing designers of applications in health care.

Keywords: health care, accidents, emergent properties, interactive complexity

Introduction

Efforts to improve patient safety often focus on automation (Leapfrog Group 1999) as a means for preventing human practitioner "error". Technological change in an ongoing field of activity, however, produces a complex set of organizational reverberations that are difficult to anticipate or predict and may go far beyond the expectations of designers (Cook and Woods 1996).

A difficult problem with the design of automation is the unanticipated interaction of multiple different automation systems. This paper discusses an archetypal case involving the failure of an automated drug-dispensing unit in an emergency department due to such an interaction, its local consequences and some of the implications for proposals to use automation to advance patient safety (Perry, Wears *et al* 2005). Our purpose is not to present a comprehensive analysis of this specific system, but to use this case to illustrate more general issues common in the introduction of advanced technology into the complex work environment of health care.

Case Description

A critically ill patient presented to a busy emergency department (ED) serving a large urban, indigent population. Intravenous access was obtained and a variety of pharmacologic agents were ordered. The resuscitation nurse went to obtain medications from an automated dispensing unit (ADU), part of a computer-based dispensing system in use throughout the hospital. He found an uninformative error message on the computer screen ("Printer not available") and an unresponsive keyboard. The system did not respond to any commands and would not dispense the required medications.

The ED nurse abandoned efforts to get the ADU to work and asked the unit clerk to notify the main pharmacy that the ADU was "down" and emergency medications were needed. He asked another nurse to try other ADUs in the ED. Other ED staff became aware of the problem and joined in the search for the sought after drugs. Some were discovered on top of another ADU in the ED, waiting to be returned to stock. Anticipating the patient's clinical deterioration, the ED physicians opened the resuscitation cart ("crash cart") and prepared to intubate the patient, using the medications and equipment stored there. A pharmacist came to the ED and examined the unresponsive ADU. He decided not to use the bypass facility for downtime access because neither the drawers nor the bins were labelled with the names of the medications they contained, and this information could not be obtained from a non-functioning unit. Instead, he arranged for the pharmacy staff to use runners to bring medications from the main pharmacy, one floor below, to the ED in response to telephone requests. The patient eventually received the requested medications; her condition improved; she survived and was later discharged from the hospital.

Reconstruction of the Chain of Events

A series of interviews with the ED staff, pharmacists, computer specialists and the ADU manufacturer's representative enabled a reconstruction of the complex sequence of events leading to this incident. (The sequence is summarized in schematic form in Table 1). The hospital had installed a popular computer-controlled automated dispensing system for drugs and supplies in 1994 to improve inventory tracking and reduce errors and pilferage, especially of controlled substances. The system was regarded as mature and reliable, and had been regularly upgraded. Other than a limited number of resuscitation drugs stored in "crash carts", all hospital medications were dispensed via this system. At the time of this incident, there were 40 ADUs linked to two centrally located computers by a general-purpose computer network that provided connectivity to the hospital information system (HIS).

To enhance safety within the hospital, the ADUs were programmed to deny access to a drug unless there was a current, valid, pharmacist-approved order for it in the HIS pharmacy subsystem. This safety feature was implemented by a software interlock mechanism between the HIS, the pharmacy computer, and the ADUs. When a user attempted to retrieve a drug for a patient from the dispensing unit, the ADU would query the HIS via the pharmacy computers and provide the medication only if a validated order could be found in the HIS. This feature was not activated in the ED because of the time constraints associated with ED drug orders and delivery.

About two weeks prior to the incident, the hospital began a major HIS software upgrade that was complicated by a sudden, unexpected hardware failure resulting in the complete loss of all HIS functions. In response, operators in the pharmacy disabled the safety interlock feature that required order checking before dispensing medications so that nursing staff on the wards could obtain drugs. As the HIS came back online, the pharmacy operators enabled this feature in order to restore normal operations. However, the HIS crashed repeatedly during this process, prompting the pharmacy operators to disable the safety interlock feature again.

The procedure for enabling and disabling the safety interlock feature entailed dialog between the pharmacy central computer and the ADU computers, which was conducted for each item in the inventory of each dispensing unit. When this procedure was started on the day of this incident, it unexpectedly created a storm of messages to and from the dispensing units. This message storm slowed the system response such that the individual units appeared to be unresponsive to keyboard commands from users. The pharmacy operators monitoring the system initially thought that network communication problems were causing the outage, but gradually came to realize that the network was functioning normally but that the ADUs were overwhelmed with messages. This phenomenon was essentially similar to denial-of-service attacks that have occurred on the internet (CERT Coordination Center 2001); the ADUs were unavailable to the users because they were busy with a large number of messages. Eventually most of the ADUs appeared to resume normal operation. The operators had assumed that ED units would not be affected by this procedure because they did not use the order checking feature. The specific reasons for the message storm, and for why the ED unit did not resume normal operation could not be determined, leaving a residual and unremovable mystery about the system.

Discussion

Many factors contributed to this near miss, at multiple levels. While the complexity of the work environment is high, and the design issues involved in anticipating what systems might interact and especially how they might be affected by transient failures are difficult, there are many additional dimensions that are important to consider in the larger picture of designing for resilience in complex worlds.

Organisational issues: The organisation conducted minimal planning for potential failure. No formal risk assessment was undertaken, and what planning occurred, occurred because of objections raised from relatively marginalized groups within the organization. It was acknowledged that a mechanical failure might prevent the ADU from dispensing drugs, but that eventuality was minimized because most drug administration is not highly time critical, and because a separate system, the "crash cart" was already in existence. The crash cart system is a manual chest, mounted on wheels, that contains drugs necessary for

the management of cardiac arrest. It did not occur to the planners that cases such as this one – not (yet) in cardiac arrest, but with highly time critical need for drugs – might occur. No scenario-based planning was done, which might have generated example cases that could have led to anticipatory changes in the crash cart (for example, stocking additional drugs that might *forestall* cardiac arrest). The organisation at this time was in a severe financial crisis, and the organisational leadership seemed blinded by the potential for savings represented by the ADU system. Objections on safety grounds tended to come from nurses or emergency physicians, who were not part of the formal planning team, and were tagged as obstructionist, non-team players, so their objections were treated as theoretical at best and specious or manipulative at worst.

The organisational response to the event is telling. Parts of the organisation believed the incident showed that the system was safe, since the nursing and pharmacy staff were able to overcome the problem and since no harm resulted. Nurses, on the other hand, began hoarding drugs as they did not trust the system, thus subverting one of its major organisational goals (inventory control). These disjoint views led to repeated conflict in the organization as more and more drugs and supplies in other locations were moved into similar controlled dispensing devices, often with little communication prior to changes.

Emergent phenomenon: The crux of this incident was the unanticipated interaction of two nominally separate computer systems. The HIS – ADU system was intentionally limited to certain parts of the hospital, but "spilled over" to involve ADUs in the ED, which never were part of the HIS – ADU axis. This, and the co-residence of all these systems on a common Ethernet backbone, was a source on inapparent coupling. By virtue of increased "coupling" between components of the system, automation generates opportunities for a complex systems failure, a "normal accident" (Cook and Woods 1994; Perrow 1999). The incident emerged from the interaction of major and minor faults which were individually insufficient to have produced this incident. The design problem here is that validation of individual device design is an insufficient basis from which to conclude that use in context will attain the design performance levels.

Properties of the health care sector: The case illustrates several aspects of the health care industry that make it peculiarly vulnerable at this stage in its history. First, the application of complex computer technology to certain aspects of technical work in health care is relatively new. Until recently, the majority of technological applications in healthcare were in the form of embedded systems that were highly task specific (*eg*, infusion pumps, or imaging devices). This has shifted in recent years, from systems that are narrowly focused on a specific (typically clinical) task, to systems that are more broadly aimed at solving organizational problems, such as billing, inventory control, accounting, *etc*, and are only secondarily (if at all) directed at supporting clinical work. The relative youth of the field means there is a relatively meagre infrastructure (people, procedures, resources) available for assessing the impact of technological change.

Second, these new types of systems, such as the one discussed here, are highly conflicted, because they advance organizational goals but impress an already beleaguered group of workers into servicing them, without providing the workers a commensurate benefit. Grudin's Law (Grudin 1994) still applies, although the managers and purchasers of such systems do not seem to be aware of it.

Third, health care has been historically a relatively insular, isolating field. There is a broad, general lack of awareness of large bodies of knowledge in design and engineering that might usefully be applied to health care problems. Thus, even if thoughtful managers in health care organizations wonder about the potential adverse effects of a new technology, they are generally unaware that methods and expertise are available upon which they might call; instead, they would more likely convene a group of their own workers, who might be well-intended but not well-versed in technology or risk assessment.

Fourth, health care organizations, at least in the US, are in a sense, barely organizations at all, but rather tense social webs of sometimes competing, sometimes cooperating groups, whose governance seems best modelled by the feudal system (Lorenzi, Riley *et al* 1997; Wears 2001). The relations among physicians, nurses, pharmacists, technicians, and administrators are complex and tense (Nemeth, Cook *et al* 2004). Because information about risk in such a setting might easily be subverted to advance a group agenda, it is frequently not sought, suppressed, or "interpreted in light of the source."

Finally, there is little or no formal, regulatory pressure to encourage the prior or ongoing evaluation of these systems. While the US Food and Drug Administration does evaluate medical devices, their definition of a medical device is purposely narrow, and would not include systems such as the ADU devices illustrated here. In addition, the FDA would not be well-positioned to evaluate a device such as an ADU in its environment; thus emergent vulnerabilities would likely be missed, even if such evaluations were mandated.

Conclusion

Automation offers a variety of tangible benefits and is often proposed as a means to increase patient safety. But, as this case demonstrates, automation also creates new vulnerabilities, some with substantial consequences. Emergent vulnerabilities, such as arise from the interaction among disparate, independently designed components, seem almost impossible to foresee in anything other than the most general terms. Health care seems especially vulnerable to these sorts of threats for several reasons: 1) The relative youth of complex computer application is the field; 2) The general unfamiliarity of health professionals and managers with methods for reducing vulnerabilities; 3) The fragmentary nature of health care "organizations"; 4) The potential subversion of risk information into internal, conflicting agendas; and 5) And a lack of formal or regulatory frameworks promoting the assessment of many types of new technologies. These factors are as much social-organizational as they are technological. As we consider increased automation in health care, we should pay as much attention to anticipating new vulnerabilities and the social component of the sociotechnical system, and to introducing well-established design and engineering risk assessment methods into the field as we do to the anticipated benefits (Nemeth, O'Connor *et al* 2004).

References

CERT Coordination Center. (2001). "Denial of Service Attacks." Retrieved 12 December 2001, 2001, from http://www.cert.org/tech_tips/denial_of_service.html.

Cook, R. I. and D. D. Woods (1994). Operating at the sharp end: the complexity of human error. Human Error in Medicine. M. S. Bogner. Hillsdale, NJ, Lawrence Erlbaum Associates**:** 255-310.

Cook, R. I. and D. D. Woods (1996). "Adapting to new technology in the operating room." Hum Factors **38**(4): 593-613.

Grudin, J. (1994). "Computer-supported cooperative work: history and focus." IEEE Computer **27**(5): 19 - 27.

Leapfrog Group. (1999). "Leapfrog initiatives to drive great leaps in patient safety." Retrieved 17 October 2000, from http://www.leapfroggroup.org/safety1.htm.

Lorenzi, N. M., R. T. Riley, *et al* (1997). "Antecedents of the people and organizational aspects of medical informatics: review of the literature." J Am Med Inform Assoc **4**(2): 79-93.

Nemeth, C., M. O'Connor, *et al* (2004). "Crafting information technology solutions, not experiments, for the emergency department." Academic Emergency Medicine **11**(11): 1114-1117.

Nemeth, C. P., R. I. Cook, *et al* (2004). "The messy details: insights from the study of technical work in health care." IEEE Transactions on Systems, Man, and Cybernetics: Part A **34**(6): 689 - 692.

Perrow, C. (1999). Normal Accidents: Living With High-Risk Technologies. Princeton, NJ, Princeton University Press.

Perry, S. J., R. L. Wears, *et al* (2005). "The role of automation in complex system failures." Journal of Patient Safety **xxxx (in press)**.

Wears, R. L. (2001). Oral remarks as symposium discussant: Challeges to building safe healthcare organizations: from the inside looking out. Washington, DC, Academy of Management.

Table 1. Time Sequence of Events

The time course of patient events, staff actions, and system event is outlined here. Times are approximate as they were not always documented and were estimated by participants during debriefing. Time zero was assigned to the point at which severe respiratory distress requiring resuscitation was recognized. Negative (-) times refer to events prior to this point and positive (+) to events afterward.

Approximate Time	Patient Events	Clinical Staff Actions	Automation Events
- 1 month	Sustains cardiac arrest and successful resuscitation in ED		
- 2 weeks			HIS software upgrade begins
- 11 days			Hardware failure stops HIS functions
			ADU drug order interlock disabled
- 2 days			HIS function re-established
- 1 day			ADU drug order interlock enabled
- 1 hour	Arrives in ED, placed in routine bed		HIS crashes
- 30 minutes		Initial orders written and given orally to nurses	ADU drug order interlock disable procedure started
- 20 minutes	Gradual deterioration in respiratory status		ADUs begin to appear off-line. ADU non-functional in resuscitation area
Time 0	Placed in resuscitation for severe respiratory distress		(ADU non-functional)
+ 3 minutes		Emergency drug orders given verbally	(ADU non-functional)
+ 6 minutes		Nurse finds ADU non-functional in resuscitation area	(ADU non-functional)
+ 8 minutes		Clerk notifies pharmacy of emergency need for drugs, non-functioning ADU	(ADU non-functional)
		Additional nurses try other nearby ADUs	
		Additional nurses attempt to locate drugs from "likely sites"	
+ 12 minutes		Physicians realize drugs will be delayed, open crash cart and prepare for emergency intubation if needed	(ADU non-functional)
+ 13 minutes		Pharmacist arrives in ED, investigates ADU, arranges for runners to bring drugs in response to telephone	(ADU non-functional)
+15 minutes		Albuterol found in another ED treatment area, given to patient	(ADU non-functional)

Approximate Time	Patient Events	Clinical Staff Actions	Automation Events
+ 17 minutes		Runner system established and functioning	(ADU non-functional)
+ 20 minutes		Pharmacy operator arrives to investigate ADU problem	(ADU non-functional)
+ 30 minutes	All medications received, respiratory status begins to improve		(ADU non-functional)
+ 45 minutes			ADU rebooted successfully, begins to function
+ 2 hours	Transferred to intensive care unit		
+ 4 hours	Intubated for respiratory failure		
+ 8 days	Discharged to home without sequelae		

What Makes Emergency Ambulance Command and Control Complex?

B.L. William Wong, Jared Hayes*, Tony Moore*,

Dept. of Computing Science, Middlesex University,
*Department of Informatics, University of Otage, New Zealand.
What Makes Emergency Ambulance Command and Control Complex?

Abstract: This paper reports initial findings from a comparative analysis of complexity between two emergency ambulance command and control centers in New Zealand. The two centers studied differ significantly in a number of areas including size, urban-ness of their areas of responsibility, and the dispatch technologies used. An earlier study focusing on the decision making strategies of the ambulance dispatchers in these centers led to the identification of 38 factors thought to contribute to complexity. To examine the effect of these factors, 24 ambulance dispatchers participated in a survey that assessed both the perceived complexity and frequency of occurrence for each factor. Our findings show that despite similarities in the task, many of the process-inherent complexities can be traced back to environmental and technology factors and their interaction. Surprisingly, the perception of complexity in process-inherent factors such as determining the next quickest ambulance appears higher in the control centre with a modern map-based dispatch system, than in the one without. This and other findings raise a number of questions which will be reported here but are still being investigated.

Keywords: complexity, emergency ambulance dispatch, emergency medical dispatch, command and control

Introduction

Emergency ambulance command and control, more formally known as Emergency Medical Dispatch, deals with the receipt of calls for medical emergencies, the medical prioritization of the calls, and the coordination and dispatching of medical assistance to the incident location (Clawson & Dernocoeur, 1998). Superficially, the process appears rather straightforward – answer a call, assess the urgency, locate the quickest available ambulance, and dispatch it – but is it that simple? Regularly, complications arise due to: uncertain, out of sequence or incomplete information; information arriving from different sources, such as multiple calls reporting on the same major accident, that needs integration; having to manage many incidents occurring simultaneously; having to allocate resources between competing incidents and thus having to know where each ambulance is so that decisions can be made about their most effective deployment; all combined with the presence of time pressure. For example, every delay of one minute can reduce the chances of a favorable outcome for a person suffering a heart attack by 10% (National Center for Early Defibrillation, 2002). In addition, (Vicente, 1999) describes several characteristics of complex socio-technical systems that are applicable to ambulance command and control. These include the need for social interaction between distributed system components, heterogeneous perspectives in regards to the goals of workers within the system, the dynamic and hazardous nature of the work environment, and the requirement for a computer interface to act as a window into the state of the system.

This paper reports on our initial findings from a survey to identify the specific factors that contribute to complexity in command and control at two ambulance control centers in New Zealand. We conducted a comparative analysis of the factors in order to identify those that negatively affect dispatch performance. We envisaged that by identifying these factors and by understanding how they affect dispatch performance, we might be able to help design systems that would reduce the complexity, reduce assessment and planning efforts, and therefore accelerate the process of getting medical assistance to people in need.

In an earlier study we used the Critical Decision Method to investigate the nature of decision making in the two control centers. We identified 18 decision strategies invoked by ambulance dispatchers in these two control centers. While this work has been reported in more detail elsewhere (Hayes, Moore, Benwell, & Wong, 2004), an example of one such strategy is described next: Dispatchers try to match the severity of an incident with the skills of a particular ambulance crew. For a priority one call when there are multiple

choices of vehicles to send, dispatchers will normally send the vehicles with a crew that will best match the requirements at the scene. However dispatchers are obliged to balance the workload of the crews and this often dictates the dispatch decisions.

Another example of a mental strategy identified in earlier studies of decision making in ambulance dispatching, is the "focus and compare" strategy for locating the next available ambulance resource. We found that the dispatcher combines information about the location of the incident and location of the stations to first determine the set of nearest ambulances. The dispatcher focuses only on those closest and eliminates from consideration those possible but further away, and compares between the candidate stations, spiraling outwards from the closest, and then dispatches the most appropriate based on availability, quickest to scene and crew qualification (Wong, 2000).

Whilst strategies such as the one above are useful in describing how dispatchers make decisions and the difficulties encountered in, for example, how they assess situations, collate information, and make trade-off decisions, they provided little insight into the severity of the problems dispatchers face in carrying out that strategy, and the extent to which such problems occur, and hence their impact on dispatch performance. These problems are often representative of the complexities that underlie the strategies. Therefore in order to appreciate this, we decided to conduct a follow-up study, asking the question of "What makes their dispatching jobs complex and therefore difficult, and to what extent do they occur?" We used the Emergent Themes Analysis, or ETA (Wong & Blandford, 2002) to extract from the decision strategies and observations in the communication centres, 38 factors that were considered to contribute to the complexity of the dispatch task were identified. These factors include the number of possible routes to an accident, traffic congestion, and determining the next quickest ambulance. Using an approach similar to that taken by (Koros, Rocco, Panjwani, Ingurgio, & D'Arcy, 2003) to study complexity in air traffic control, we administered a questionnaire to 24 dispatchers, who collectively represented over 50% of the total ambulance dispatch staff at the two centers. The following sections will provide some background to ambulance dispatching in New Zealand, a description of the methodology, the results and a discussion of those findings, highlighting some interesting issues for further investigation.

Background: Ambulance dispatching in New Zealand

The ambulance dispatchers surveyed in this study are based in Southern Regional Communications Centre (RCC) in Dunedin in the south of the country and the Northern RCC in Auckland in the north.

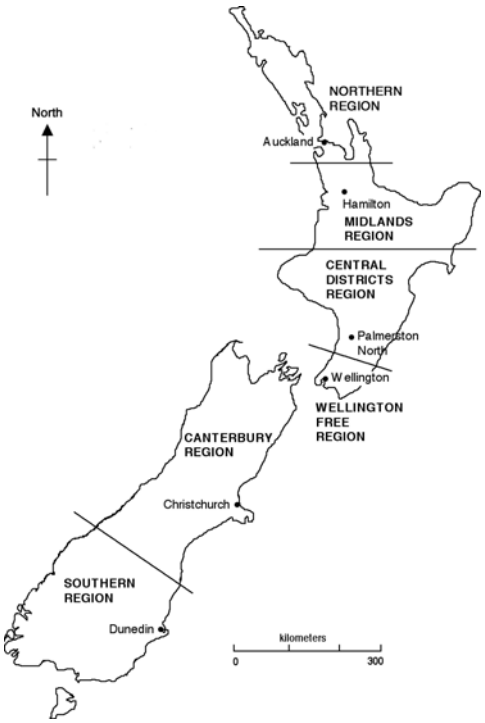


Figure 1 – Ambulance control regions.

The Southern RCC is responsible for a very large geographical area covering a significant part of the South Island of New Zealand. The land area is approximately 54,000 square km and encompasses mountain ranges, lakes and rugged coastline to the west and undulating coastal farming areas to the east. The area has a population of 273,500 people. The towns and cities are generally small and distributed across the region and connected by highways. Travel times between these towns is generally in the vicinity of one hours driving. The Southern RCC controls 48 ambulances, deployed in stations located within the towns and cities. The Southern RCC receives over 20,000 emergency calls a year, and uses an older text-based computer-aided dispatch (CAD) system to record calls, track jobs, and coordinate and dispatch ambulances. While the Advanced Medical Priority Dispatch System or AMPDS, is available, it is not integrated into the CAD system. The AMPDS is a set of medical protocols or structured questions that help a dispatcher determine the severity of a medical emergency. A simpler variant of the AMPDS is currently being used in a manual mode. The Southern RCC was one of the earliest Centers to incorporate computerized systems, but as plans are underway to rationalize the communications centers throughout the country, it has not been upgraded, and will be amalgamated with another communication centre in the near future.

The Northern RCC is based in the North Island city of Auckland, and is responsible for much smaller land area of approximately 21,000 sq km. While the region is hilly and also has a lot of farm land, the main bulk of the over 1.2 million residents in the region live within the main urban area of the city of Auckland. Within the urban areas, the road network is significantly more complex than the rural road network of the Southern RCC. The Northern RCC is responsible for controlling 74 ambulances and responds to over 100,000 emergency calls a year. The Centre is equipped with a newer generation map-based CAD system, which provides a spatial representation of the location of ambulances when they are at their stations, at emergency incidents, or other fixed points. The current system, due to be upgraded, does not incorporate real time vehicle tracking and as such when resources are in transit the dispatchers are not able to immediately determine their location. This Centre also has a computerized version of the AMPDS for prioritization of urgency.

The Dispatch Process: Whilst similar in many regards, there are a number of notable differences in terms of the dispatch process in each centre. This includes the team of dedicated call takers in the Northern centre. When an emergency call arrives in this centre it is usually answered by any one of a number of call takers. It is the job of the call taker to get the details from the caller regarding the incident. This is in the first instance the location of the incident and the condition of the patient so as the response priority can be determined. When this information has been entered into the CAD system, the job appears on the screen of the dispatcher responsible for the specific area within the region. They are then able to allocate resources to this incident. In many instances this information alone is not sufficient to make a dispatch decision and the dispatcher is able to view additional details about the incident as they are entered by the call taker and/or listen in as the call taker gathers additional details from the caller about the incident. In comparison the Southern centre does not have a team of call takers and instead the dispatchers work in unison. One dispatcher plays the role of the call taker and therefore collects details from the caller similar to those taken in the Northern centre, recording details such as location and patient injuries, followed by additional information such as telephone number, caller name, whilst the other dispatcher works in parallel to assess the resource situation and determine which resource is best suited to the needs of the patient.

In both centers, at the same time as making dispatch decisions – which ambulance is the nearest available? which ambulance is the next quickest? where are the ambulances in transit and returning that could be quicker to send? – the dispatcher will also be managing additional incidents and resources to provide the best level of ambulance coverage across the region. Gaps in coverage, or areas with poor ambulance coverage, are to be avoided as they will increase response times to callers within those gaps. The call taker (or dispatcher taking the call) can also end up engaged with the caller for a considerable amount of time. For instance, call takers and dispatchers have been known to keep in communication with distraught callers until the ambulance arrives, or to provide "telephone CPR" (instructions over the telephone to the caller on how to perform cardio-pulmonary resuscitation).

Dispatchers also often have to deal with multiple calls to the same incident, e.g. major crash on a highway, where each caller reports variations and the dispatcher has to determine if the many calls are for the same incident. Often during periods of high workload, the dispatchers have to deal with many simultaneous incidents with varying levels of urgency and size, in different parts of the region. Effective management of the situation will require a good awareness of many factors (Blandford & Wong, 2004), including knowledge of the jobs and their locations, the road network, traffic conditions, tracking the whereabouts of ambulances, intentions and planned next activities of each ambulance, e.g. upon completion of a job, ambulance A could be planned to be sent to job B. Such intentions are often not recorded in the system until it actually happens, as re-planning occurs quite frequently due to changing situations.

Collectively, all this makes the dispatch process difficult. In this study, we wanted to find out what these factors are and to what extent they contribute to dispatch task complexity.

Methodology

We conducted a questionnaire survey of 24 emergency ambulance dispatchers. 10 from the Southern centre and 14 from the Northern centre. This represents a sample size that is in excess of half the total number of dispatchers at the two Centers. In addition to collecting demographic data such as age, sex and experience in control, participants in the survey were asked to rate 38 factors thought to contribute to complexity in emergency ambulance command and control. Participants rated each factor on a 5-point Likert scale, assessing the factor's perceived contribution to complexity, and the frequency with which that factor was thought to have occurred. There were additional spaces for the participant to include other factors not in the list. The questionnaire approach was selected instead of other measures such as NASA TLX (Hart & Staveland, 1988), as while well established, the TLX is principally used to measure perceived mental workload, rather than to estimate the effect and frequency that a set of factors has on task complexity. Furthermore, one of the 38 factors considered to contribute to complexity was workload, and measuring workload only would not be representative.

A series of ANOVA (analysis of variance) tests were conducted on the data from the returned questionnaires using SPSS, a statistical analysis software. We tested the data for differences in the

Complexity, C, and Frequency, F, scores between the Centers. We also used a measure similar to Koros, et al (2003), called the Complexity Index (CI) where CI = C + F, to reflect each factors overall Complexity and Frequency scores, which was then tested for differences between centers. The CI was a convenient measure as it made it possible to easily compare factors that are say, high on complexity and high on frequency, with factors that are high on complexity but low on frequency. CI was also tested for differences between the two Centers. The next section will present some of the results.

Results

An ANOVA procedure was conducted on the data to determine if there were significant differences between what dispatchers at each Centre considered to contribute to task complexity, the frequency with which they occurred, and on CI. The results of the ANOVA procedure of the CI of the 38 variables are presented in Tables 1, 2 and 3. They have been organized according to three bands of complexity. Band 1 which we will call High Complexity factors, shows those factors where the CI is greater than 7.0; Band 2 Moderate Complexity, where CI scores are greater than 6.0 and less than 7.0; and Band 3, Low Complexity, where CI scores are less than 6.0.

In addition, the factors have been categorized into several groupings. These categories are listed and explained below.

a. Ambulance factors refer to the factors that relate to the ambulance dispatch process such as identifying the vehicle that can be expected to arrive at the scene the fastest, identifying the next quickest, and locating vehicles that are in transit.
b. Crew factors refer to keeping track of the ambulance crews, their workloads, planning for breaks, and their suitability for an incident in terms of skill levels.
c. Incident factors relate to factors like awareness of the location and what is happening with each incident and therefore being able to anticipate and cater for future needs. Also ensuring adequate resources have been dispatched fits within this category.
d. Patient factors include determining and prioritizing the medical conditions of the patients.
e. Geography factors refer to the nature of the terrain in which the ambulances operate, and includes access to incident scenes, dispatch alternatives, familiarity of the area, and traffic congestion.
f. Equipment factors relate to the systems in use in general, e.g. radio dead spots and hence the likelihood of non-responses to radio calls when an ambulance is in an area, or malfunctioning systems.

Table 1 – Band 1 High Complexity CI > 7.0 for each Centre.

Category	Southern RCC	CI	Northern RCC	CI	Mean	F	Sig.
General Factors	05. High workload	7.2	05. High workload	8.15	7.74	2.283	0.146
	36. Managing patient transfers	7	36. Managing patient transfers	7.62	7.36	0.706	0.411
			08. Time pressure	7.69	7.22	1.99	0.173
			06. Unpredictability of future events	7.25	6.95	0.376	0.547
Ambulance Factors			18. Quickest vehicle†	7.54	6.87	3.836	0.064
			19. Quickest Additional vehicles*	7.31	6.7	4.383	0.049
			28. Location of resources in transit*	8	6.57	12.913	0.002
	24. Determining next available resource	7.5	24. Determining next available resource	7.77	7.65	0.184	0.672
			27. Providing coverage*	7.69	5.32	42.821	0
Crew Factors			20. Matching crew to incident	7.15	6.78	1.357	0.257
			26. Determining crew status	7.54	6.87	3.148	0.091
			29. Managing crew workload	7.46	7.09	0.749	0.397
Incident Factors	12. Uncertain location	7.4	12. Uncertain location	7.23	7.3	0.07	0.794
	16. Determining priority*	7.9					
	17. Incident awareness	7.8					
	25. Ensuring required resources at incident	7					
Policy/Orgn Factors	09. Different resource allocation procedures for diff areas	7.5					

g. Policy and Organizational factors refer to how work needs to be done and includes the use of different terminology when dealing with different services e.g. police and fire, or different procedures for handling dangerous situations, e.g. dealing with patients at a domestic dispute versus dealing with patients at a chemical spill.

Table 1 Band 1 High Complexity (CI > 7.0) shows that for those factors considered to contribute significantly to complexity, i.e. high complexity and high frequency, dispatchers in the more urban Auckland centre have identified 13 factors as compared to the eight factors identified by dispatchers in the more rural Southern RCC. There are only four factors common to both Centers at this level. These were high workload, uncertain location, determining the next available resource, and managing patient transfers. ANOVA tests show no significant differences in these factors between Centers.

However ANOVA tests on another four factors within this band showed differences that were significant ($p<0.05$). These factors were determining priority for the Southern RCC dispatchers; and identifying the quickest additional vehicles, providing coverage, and locating ambulances in transit for the dispatchers at the Northern RCC.

The dispatchers at the Northern RCC also found their work further complicated by the unpredictability of future events, time pressure, identifying the quickest ambulance, matching an appropriate crew to an incident, and managing crew workload.

Table 2 – Band 2 Moderate Complexity 6.0 < CI < 7.0 for each Centre.

Category	Southern RCC	CI	Northern RCC	CI	Mean	F	Sig.
General Factors	07. Rate of change	6.78	07. Rate of change	6.62	6.68	0.023	0.882
	06. Unpredictability of future events	6.6					
	08. Time pressure	6.6					
			14. Information collation*	6.85	6.39	5.136	0.034
	35. Fatigue	6.5	35. Fatigue	6.38	6.43	0.016	0.9
Ambulance Factors	18. Quickest vehicle†	6					
Crew Factors	29. Managing crew workload	6.6					
	20. Matching crew to incident	6.3					
	26. Determining crew status	6					
Incident Factors			17. Incident awareness	6.69	7.17	2.012	0.171
			25. Ensuring required resources at incident	6.62	6.78	0.297	0.591
Patient Factors	15. Not talking with caller	6.4	15. Not talking with caller	6.77	6.61	0.466	0.502
	13. Uncertain patient condition	6.3	13. Uncertain patient condition	6.67	6.5	0.241	0.629
Geography Factors	04. Limited access†	6.2	23. Low number of dispatch alternatives	6.23	5.74	1.341	0.26
			02. Traffic congestion*	6.23	5.45	5.153	0.034
	21. Unfamiliar with area	6.2	21. Unfamiliar with area	6.15	6.17	0.008	0.929
Equipment Factors	37. Other distractions	6.5	37. Other distractions	6	6.23	0.387	0.541
			30. Radio dead spots	6.69	6.35	1.563	0.225
			31. Equipment malfunctions*	6.15	5.74	4.636	0.043
Policy/Orgn Factors	34. Different terminology between services†	6					
	10. Different response procedures for diff incident types	6.7	10. Different response procedures for diff incident types	6.85	6.78	0.037	0.849
			09. Different resource allocation procedures for diff areas	6.77	7.09	0.765	0.392

Table 2 Band 2Moderate Complexity (6.0 < CI < 7.0) shows those complexity factors that had a CI score of between 6.0 and 7.0. This CI indicates that these factors are less significant contributors to complexity, suggesting moderate complexity and moderate frequency ratings. Although the factors are different, there are approximately equal numbers of factors cited by both Southern and the Auckland RCCs. Dispatchers in the Southern RCC identified 15 factors while their Auckland counterparts identified 14 factors in this band, of which only six factors are common. Both groups felt that the rate of change or the tempo of events, the need for different procedures for different incident types, uncertainty in patient's condition, not being able to speak directly to the caller (when one is not taking the call), unfamiliarity with the area, fatigue and other distractions, contribute to complexity.

Other factors that were significantly different (*$p<0.05$)* were traffic congestion that was more likely to be experienced in the urban Auckland areas than in the more rural Southern region, the need to collate more information from more sources, and the apparently more consequences from equipment malfunctions in the Northern region. Dispatchers in the Southern centre rated unpredictability of future events, time pressure, determining the quickest ambulance, matching the appropriate crews to incidents, determining crew status, managing crew workload, and different terminology between services as other factors that contributed to complexity within this band.

Table 3 – Band 3 Low Complexity CI < 6.0 for each Centre.

Category	Southern RCC	CI	Northern RCC	CI	Mean	F	Sig.
General Factors	11. Low workload*	4.6	11. Low workload*	2.69	3.52	5.072	0.035
	14. Information collation*	5.8					
Ambulance Factors	19. Quickest Additional vehicles*	5.9					
	28. Location of resources in transit*	4.7					
	27. Providing coverage*	1.89					
Incident Factors			16. Determining priority*	5.77	6.7	4.521	0.046
Geography Factors	01. Road works/road conditions	5.3	01. Road works/road conditions	5	5.13	0.197	0.661
	03. Number of possible routes	4.6	03. Number of possible routes	5.17	4.91	0.7	0.413
			04. Limited access†	5	5.55	3.967	0.06
	02. Traffic congestion*	4.33					
	23. Low number of dispatch alternatives	5.1					
	22. High number of dispatch alternatives	4.56	22. High number of dispatch alternatives	5.31	5	1.02	0.325
Equipment Factors	30. Radio dead spots	5.9					
	32. Equipment distractions	5.8	32. Equipment distractions	4.23	4.91	2.868	0.105
	31. Equipment malfunctions*	5.2					
Policy/Orgn Factors	33. Joint responses with other regions	5.63	33. Joint responses with other regions	4.77	5.1	1.715	0.206
			34. Different terminology between services†	4.38	5.09	3.557	0.073
	38. Training exercises	4.3	38. Training exercises	3.85	4.04	0.472	0.5

Table 3 Band 3 Low Complexity (CI < 6.0). Dispatchers in the Southern RCC rated 15 factors as not contributing significantly to complexity, in comparison with 10 identified by their Northern counterparts. Of these, the following seven factors are common: low workload, road conditions, number of possible routes, having high number of dispatch alternatives (since it does not occur frequently as limited resources often limit the options available), equipment distractions such as poor interfaces, joint responses with other regions, and training exercises.

What is interesting is that the ambulance factors – identifying the quickest additional ambulances, locating ambulances in transit and providing coverage – were identified in this low significance band by the Southern dispatchers. In contrast, the Northern dispatchers ranked these factors as the among the highest contributors to complexity. While traffic congestion was highlighted as a high contributor by the Northern dispatchers, it is a low factor in the south. Determining the medical priority of emergency calls is a low contributor in the North as they do have a system, called ProQA, that automates the prioritization decisions.

Discussion

What do the results tell us? In this section, we will discuss some of the differences between the two Centers and the lessons that we can learn from them.

Partial solutions can add significantly to complexity. We mentioned earlier that the Northern Centre has a map-based CAD system which, graphically shows on the computer-generated map of the region, the location of the ambulances and the location of the emergency calls. We also mentioned that, due to a variety of reasons, the system only tracks the position of the vehicles at fixed points and not in transit. Our survey suggests that such partial implementations of systems can make the task more difficult than is necessary. Locating the quickest additional vehicles, determining which ambulances are likely to be available next, locating ambulances in transit, and providing coverage, are rated as high contributors to complexity by the Northern dispatchers. In contrast, the Southern dispatchers who do not have a computerized map-based CAD system, do not consider this aspect of the dispatch task a significant factor. Similarly, crew management – planning their workload, determining crew status – is also not a significant problem for the Southern dispatchers, but does present some challenges to the Northern dispatchers.

On the other hand, what the Southern dispatchers found complex was maintaining incident awareness. Being able to understand the nature of the emergencies and having a current awareness of the situation allows the dispatcher to anticipate the need for additional resources (Blandford & Wong, 2004). Without the computerised map-based CAD system, the Southern dispatchers appear to be able to focus on managing the incidents, a key aspect of dispatch work, which they find represents the greatest source of complexity for them. Knowing where their ambulances are, keeping track of their movements, estimating which will be the next quickest to an incident, are not significant issues to the Southern dispatchers who have to maintain this mental picture of the situation (Blandford and Wong 2004) in their heads. Whereas the Northern dispatchers who have only part of this task supported by the computer, find that the they have to focus significant effort to developing and maintaining that mental picture of both the vehicle and crew situation.

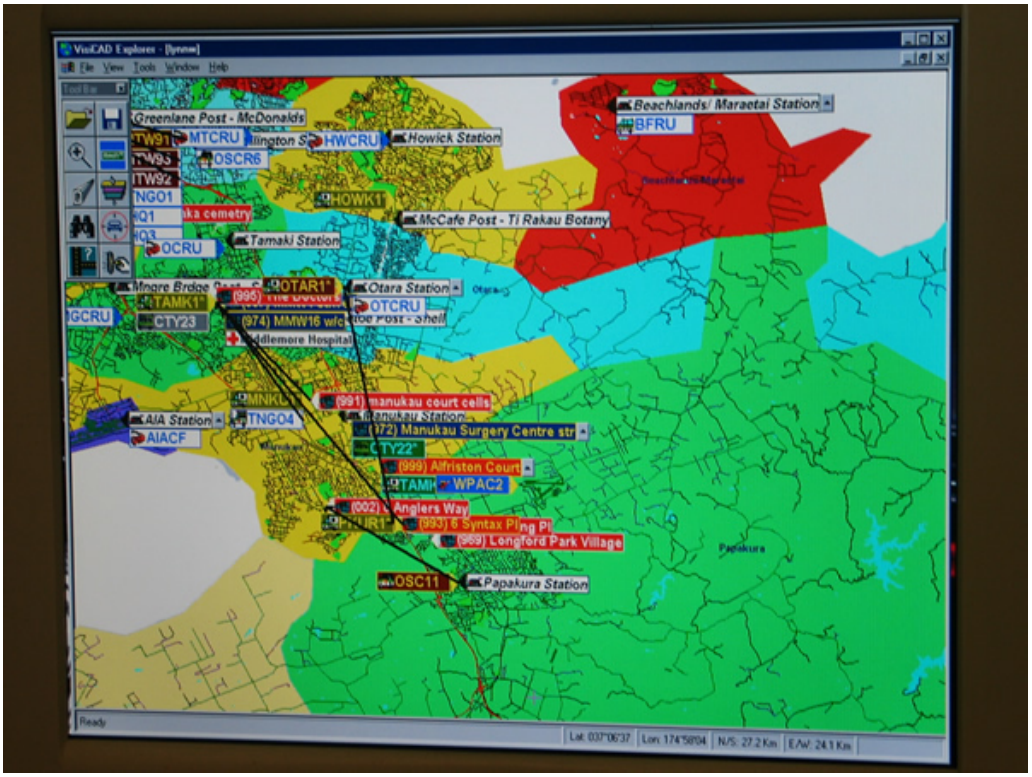


Figure 1 – Photograph of map-based CAD display in use in the Northern centre.

Figure 1 shows a photograph of the computer map-based CAD screen in use. The screen shows allocated and un-allocated jobs, ambulance stations, hospitals and standby points, and ambulances that have been assigned to a job, are on station, or at an incident. The straight black lines are used to show the job that an ambulance has been allocated or the destination of that ambulance. The display appears to suffer from the data overload problems of visual clutter leading to workload bottlenecks of the kind described by (Woods, Patterson, & Roth, 2002). The high rating of complexity in controlling the ambulances and crew, represents complexities that are imposed as a result of mismatches in the implementation of technology, rather than due to the inherent complexities in the dispatch task.

Interaction between workload, technology and task-environment characteristics. Workload is also a serious contributor to complexity. Its effects on complexity are due to having more parallel activities to co-ordinate, i.e. more simultaneous emergency calls to attend, more trade-offs to make within a given time, less time to give to each decision and activity. Under the same workload conditions as the dispatchers in the Southern centre the Northern dispatchers may be better able to manage their resources. Therefore it is perhaps not the partial technology solution alone that results in complexity but rather the interaction between the partial technology solution within the context of the characteristics of the Northern region, such as a 'tighter' road network and higher volume of jobs requiring, but the technology not providing an adequate level of support to the dispatchers to keep track of the resources. While the need to consider the human, task and environment in designing systems is not novel (Bailey, 1996), in isolation the factors may be manageable, but placed in context of each, these problems often multiply. More significantly, it can divert the attention from the prime task of managing incidents and the situation, forcing the dispatcher to attend more to the basics of the task, tracking vehicles and crew resources. Other aspects of the dispatch planning process that interact with the above are the planning and re-planning processes which result in many intermediate outputs, such as intentions to send ambulance A upon completion of job 1, to job 2 instead of returning to station. The firming up of such a decision may be dependent on other conditions becoming certain. In some ambulance centers, such fluid intermediate planning is catered for by dispatchers simply writing notes or placing the printed emergency call tickets in a particular semantically meaningful spatial arrangement on the desk. Such arrangements are easy enough to change, yet crucial for planning and tracking of very fluid situations (Wong & Blandford, 2004), but needs to be catered for in systems design. Their omission will lead to similar problems highlighted above.

Separating the roles can reduce complexity. Both Centers indicated that managing non-emergency patient transfers as one of their greatest sources of complexities. Managing patient transfer refer to the dispatching of usually single-crewed ambulances to ferry patients between hospitals or hospitals and home. The ambulances are largely drawn from the same fleet of vehicles used for emergency call-outs. Dispatchers have to balance off the need to ferry patients against unpredictable emergency calls. These are two very distinct roles which have very different time constraints. For emergency calls, ambulances need to arrive at the scene within eight minutes of the call in urban areas. Whereas for patient transfers, the time horizon is much longer at two hours. There are also staffing and vehicle equipment differences. Paramedic qualified staff would not be needed to ferry patients between hospitals. If for organizational or economic reasons, the patient transfers has to be managed by the same dispatchers, one solution is to segregate the information in a way that reflects the two roles. New representation design techniques being developed – information layering and segregation (Nees, Joyekurun, Villanueva, Amaldi, & Wong, 2005) using novel multi-layered display technology – could segregate the information regarding the two roles in separate and overlapping layers, but within the same visual field of view, so that when needed, an overall and integrated situation picture of the two roles can be presented.

Conclusion

This study has identified 38 factors that are thought to contribute to task complexity in emergency ambulance command and control. It has also provided a quantitative basis on which to assess the extent that the factors contribute complexity in ambulance command and control. What have we learnt? It provided a comparison of the level of complexities between the different regions, suggesting that the support systems might need to be configured differently to accommodate regional differences. We have also seen how factors can interact to create complexities which may not be apparent by themselves. Finally, we also discussed that if multiple roles cannot be separated, then, while yet to be tested, perhaps how new display

techniques and technology can be used to segregate the information about them, providing the system designers another avenue for addressing the complexities of the situation.

References

Bailey, Robert W. (1996). *Human performance engineering: Designing high quality professional user interfaces for computer products, applications and systems.* (3 ed.). Saddle Hill, NJ: Prentice Hall PTR.

Blandford, Ann, & Wong, B.L. William. (2004). Situation Awareness in Emergency Medical Dispatch. *International Journal of Human-Computer Studies, 61*(4), 421-452.

Clawson, Jeff J., & Dernocoeur, Kate Boyd. (1998). *Principles of Emergency Medical Dispatch* (2 ed.). Salt Lake City, Utah: Priority Press, The National Academy of Emergency Medical Dispatch.

Hart, Sandra G., & Staveland, Lowell E. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In P. A. Hancock & N. Meshkati (Eds.), *Human Mental Workload* (pp. 139-183). Amsterdam: Elsevier Science Publishers B.V. (North-Holland).

Hayes, J., Moore, A., Benwell, G., & Wong, B. L. W. (2004). Ambulance dispatch complexity and dispatcher decision strategies: Implications for interface design. In M. Masoodian, S. Jones & B. Rogers (Eds.), *Computer Human Interaction, Lecture Notes in Computer Science Vol. 3101, Proceedings of the 6th Asia Pacific Conference APCHI 2004, Rotorua, New Zealand, June/July 2004.* (pp. 589-593): Springer.

Koros, Anton, Rocco, Pamela S. Della, Panjwani, Gulshan, Ingurgio, Victor, & D'Arcy, Jean-Francois. (2003). *Complexity in Air Traffic Control towers: A field study. DOT/FAA/CT-TN03/14*. Atlantic City International Airport, NJ 08405: U.S. Department of Transportation, Federal Aviation Administration, William J. Hughes Technical Center.

National Center for Early Defibrillation. (2002). *What you need to know about Sudden Cardiac Arrest*. Retrieved 28 January, 2004, from http://www.early-defib.org/04_01advocacy.html

Nees, Anna, Joyekurun, Ronish, Villanueva, Rochelle, Amaldi, Paola, & Wong, William. (2005). Information layering, depth and transparency effects on multi-layered displays for command and control. In *Human Factors and Ergonomics Society 49th Annual Meeting* (pp. (submitted)): HFES.

Vicente, Kim J. (1999). *Cognitive Work Analysis: Toward safe, productive, and healthy computer-based work*. Mahwah, NJ: Lawrence Erlbaum Associates, Inc., Publishers.

Wong, B. L. William, & Blandford, Ann. (2002). Analysing ambulance dispatcher decision making: Trialing Emergent Themes Analysis. In F. Vetere, L. Johnston & R. Kushinsky (Eds.), *Human Factors 2002, the Joint Conference of the Computer Human Interaction Special Interest Group and The Ergonomics Society of Australia, HF2002* (pp. CD-ROM publication). Melbourne.

Wong, B. L. William, & Blandford, Ann. (2004). Information handling in dynamic decision making environments. In D. J. Reed, G. Baxter & M. Blythe (Eds.), *Proccedings of ECCE-12 , the 12th European Conference on Cognitive Ergonomics 2004, Living and Working with Technology, 12-15 September 2004, York.* (pp. 195-202). York: European Association of Cognitive Ergonomics.

Wong, B.L. William. (2000). The Integrated Decision Model in Emergency Dispatch Management and its Implications for Design. *Australian Journal of Information Systems, 2*(7), 95-107.

Woods, D. D., Patterson, E. S., & Roth, E. M. (2002). Can we ever escapte from data overload? A cognitive systems diagnosis. *Cognition, Technology & Work, 4*, 22-36.

V^2 : Using Violation and Vulnerability Analysis to Understand the Root-Causes of Complex Security Incidents

C.W. Johnson

Dept. of Computing Science, University of Glasgow, Glasgow, Scotland.
http://www.dcs.gla.ac.uk/~johnson, johnson@dcs.gla.ac.uk

Abstract: The US Department for Homeland Security has commissioned a number of recent reports into the 'root causes' of adverse events ranging from denial of critical infrastructure to barriers for security information transfer between Federal agencies. The US Department of Energy has also established the Information Security Resource Center to coordinate the 'root cause analysis' of security incidents. Harvard Business School (Austin and Darby 2003) highlighted several commercial initiatives to understand not simply what went wrong in any single previous incident but also to identify any further underlying vulnerability. All of these initiatives go beyond the specific events of a particular security incident to identify the underlying 'systemic' technical, managerial and organizational precursors. Unfortunately, there are relatively few established tools and techniques to support the 'root cause' analysis of such incidents. This paper, therefore, provides an introduction to V^2 (Violation and Vulnerability) diagrams. These are then used to provide a sustained analysis of Rusnak's fraudulent transactions involving the Allfirst bank. This case study is appropriate because it included failures in the underlying audit and control mechanisms. It also stemmed from individual violations, including the generation of bogus options.

Keywords: Root-cause analysis; Security violations; Accident analysis.

Introduction

A number of organizations already recognize the importance of this 'lessons learned' approach to security incidents. For example, the Los Alamos National Laboratory adopted this approach in the aftermath of a series of security related incidents involving information about nuclear weapons research. The mishandling' of two computer hard drives containing classified information led the director of the laboratory to report to the Senate Armed Services Committee. This report focused on the individual human failures that were identified as root causes. However, it also consider the contributing factors that included the 'government-wide de-emphasis on formal accounting of classified material that began in the early 1990s, which weakened security practices and created an atmosphere that led to less rigor and formality in handling classified material'(Roark, 2000). These and similar findings have led the US government to focus more directly on the different factors that contribute to the underlying causes of security vulnerabilities. The Government Security Reform Act (2001) transferred the Federal Computer Incident Response Capability (FedCIRC) from the National Institute for Standards and Technology (NIST) to the General Services Administration (GSA). As part of this move, the GSA was charged to identify patterns in the causes of security incidents (Law, 2001).

Similar trends can be observed in commercial organizations, especially business consultancies. For instance, Price Waterhouse Cooper (Skalak, 2003) recently issued a brief on understanding the root causes of financial fraud. They argued that 'the key for companies is to use a global risk paradigm that considers the root causes of financial fraud, corporate improprieties and potential regulatory malfeasance arising from different markets, and therefore different risk environments, in which global enterprises operate'. Although their focus is on the wider aspects of fraud and not simply of security, the Investigations and Forensic Services group within PWC have argued that a wider form of 'root cause' analysis represents a new paradigm for the investigation of security incidents. The intention is to probe beyond the specific violations of external agencies and junior staff members to look at the wider organizational problems that created the context and opportunities for these threats to be realized. Several accountancy firms in the US and Europe have adopted a similar perspective as they begin to examine the consequences of recent corporate scandals (Rabinowitz, 1996). It is clearly important that we learn as much as possible from those incidents that do take place if we are to reduce the likelihood and mitigate the consequences of security violations. Kilcrece et al's (2003) work on organizational structures highlights the consequences of the lack of methodological support for investigatory agencies. They argue "different members of the security team may conduct very different types of analysis, since there is no standard methodology".

The Allfirst Case Study

In 1983, the Allied Irish Bank (AIB) acquired a stake in Allfirst, then known as the First Maryland Bancorp. This stake grew until by 1989, AIB had taken acquired First Maryland through a merger. AIB planned to diversify its operations in North America (Promontory, 2002). They believed that this could best be achieved by allowing Allfirst a large amount of local autonomy. Allfirst continued have its own management team and board of directors. However, stronger control was retained over Treasury operations via the appointment of a senior AIB executive to oversee these operations. Prior to his appointment in 1989, there had only been a minimal history of currency trading at Allfirst with limited risks and a limited budget. In 1990, however, a trader was recruited to run proprietary trading. These operations continued relatively successfully until the first incumbent of this post had to be replaced in 1993. John Rusnak was recruited from a rival bank in New York, where he had traded currency options since 1989. One aspect of his recruitment was the desire by Allfirst to exploit a form of arbitrage that Rusnak specialized in. This took advantage of the differences in price between currency options and currency forwards. In simple terms, an option is an agreement that gives the buyer the right but not the obligation to buy or sell a currency at a specified price on or before a specific future date. If it is exercised, the seller must deliver the currency at the specified price. A forward is a contract to provide foreign exchange with a maturity of over 2 business days from the transaction date. Allfirst's treasury operations were divided into three areas. Rusnak's currency trading was part of the front office. The middle office was responsible for liability and risk management. The back-office was responsible for confirming, settling and accounting for foreign exchange and interest rate derivatives trades, including those initiated by Rusnak. Allfirst espoused the policy of having the back-office confirm all trades, following industry practice. The initial reports speculate that Rusnak may have put pressure on his colleagues not to confirm all of his options trades. Rusnak formed part of a relatively small and specialized group in the Foreign Exchange area. The Allfirst Tresurer was responsible both for ensuring profitable trading and for ensuring effective controls on that trading. Subsequent investigations also revealed concerns about the Treasury Funds Manager's position. Not only did they direct many of the Treasury operations but they also controlled many of the reporting procedures that were used to monitor operational risks. The Vice President for Risk Control, therefore, devised a plan so that asset and liability management reports as well as risk control summaries would be directed to senior management through his office. Unfortunately, this plan does not seem to have been implemented before the fraud was detected.

Violations and Vulnerability Analysis (V^2 Analysis)

Many different event-based techniques have been developed to support the root cause analysis of safety-related incidents. These include Events and Causal Factors charting (ECF), Multilinear Events Sequencing (MES) and Sequential Timed Event Plotting (STEP). Brevity prevents a detailed analysis of each of these approaches; the interested reader is directed to Johnson (2003). These techniques provide little specific support for the analysis of security incidents. Hence, the basic components in these event-based techniques are unchanged from their use in safety-related applications even though the details surrounding these 'dependability' failures can be very different. In contrast, Figure 1 provides an example of Violation and Vulnerability (V^2) analysis. This extends an event based modelling technique to deliberately support the identification of root causes for a wide range of security related incidents. The underlying approach is similar to the existing ECF, MES and STEP techniques, mentioned above. This V^2 diagram is constructed around a number of events that are denoted by rectangles. For example, 'AIB insert senior manager as Allfirst treasurer' and 'Treasurer is appointed to key AIB group marketing strategy committee' are both shown as events in Figure 1. These are made more likely by a number of contributory factors that are shown by ellipses. For instance, the decision to insert one of the AIB executives as the Allfirst Treasurer led to a situation in which some viewed the treasurer as a form of 'home office spy'. This contributed to the exclusion of the formed AIB executive from some senior management decisions at Allfirst. Figure 1 maps out a range of conditions that formed the background to the more detailed events mentioned in previous sections. An important objective behind the use of this modeling technique is to trace the roots of a security violation back into the underlying vulnerabilities within the operations of a company, such as Allfirst. Vulnerabilities can be thought of as a particular type of contributory factor. They create the opportunity for the violations that occur during security incidents. In Figure 1, vulnerabilities relate to the dual reporting structure between AIB and Allfirst. They weakened the supervision of the Treasurer's activities in the lead-up to the fraud. This vulnerability is denoted by the double ellipse at the bottom right

of figure 1. Subsequent V^2 diagrams can be used to map out the precise manner in which this particular contributory factor acted as a precondition for Rusnak's violations. Figure 1 illustrates the way in which V^2 diagrams can be used to look beyond the particular violations that lead to a fraud. This is important if investigations are to accurately identify the underlying managerial and organizational factors that might lead to future security problems. For instance, one response to the events at Allfirst would simply have been to focus legal retribution on the trader. This would, however, have ignored underlying problems in the relationship between AIB and Allfirst, including the supervision of key Treasury staff. This point is made more forcefully in the recommendations that emerged in the immediate aftermath of the fraud; 'In light of the foregoing considerations, AIB should consider terminating all proprietary trading activities at Allfirst, and all customer trading activities at Allfirst should be relocated to the AIB branch in New York. While the salespeople may continue to be located in Baltimore, any price-making and trade execution should be done in New York, under the direct supervision of AIB treasury' (Promontory, 2002).

Figure 2 continues the Violations and Vulnerability analysis by documenting the events leading to the hiring of Rusnak by Allfirst. Prior to 1989, Allfirst had only engaged in limited currency trading. This contributed to the decision to recruit a specialist to run their proprietary trading business. During this period, trading was focused on directional trading, in other words profits were dependent on forecasting the future price of a currency as it moved up or down on the markets. The senior trader left Allfirst and a further event in Figure 2 is used to show that the 'Treasury funds manager heads the search for a new trader'. This leads to an offer being made to Rusnak. The decision to make this offer was supported by recommendations from his previous employers at Chemical Bank. His appointment was also supported by the Allfirst Senior Management's interest in Rusnak's non-directional trading. This will be described in more detail in subsequent V^2 diagrams. Figure 2 also illustrates how these various events, together with a number of additional contributory factors lead to a further security vulnerability. Allfirst's efficiency committee suggested that the treasurer scale-back proprietary currency trading. However, the senior management interest in Rusnak's non-directional approach helped to focus the cutbacks in more conventional forms of currency trading. The senior management interest also created a situation in which the Treasury funds manager was highly protective of Rusnak and his activities. These various factors combined to weaken the monitoring and reporting procedures that were established to control the risks associated with his activities. When Rusnak's immediate trading manager resigned, his post was not filled. Lack of funds prevented a renewed appointment and so Rusnak now reported directly to the treasury funds manager who, as we have already seen, was protective of his non-directional trading strategies.

Rusnak initially created the impression that he specialized in a form of arbitrage by taking a profit from differences in the exchange rates between different markets. In particular, he claimed to make profits by holding a large number of options that were hedged by balancing positions in the cash market. These observations are denoted in Figure 3 by the contributory factors at the top-right of the diagram. The contributory factors at the top-left show that most of his trades were simpler than many at Allfirst had supposed. They involved linear trades based simply on predicted fluctuations in currency rates. This led him to buy significant quantities of Yen for future delivery. The subsequent decline in value of this currency prior to delivery left Rusnak with a loss. Combined with the image that he had fashioned for his trading activities, the loss may have created a situation in which he felt under pressure to hide the outcomes from his options on the Yen. This analysis of the top components in Figure 3 raises a number of important issues about the construction of V^2 diagrams. It can be argued that Rusnak's creation of a false impression about the nature of his trades should be 'promoted' from a contributory factor to either a violation, and therefore be linked to specific events, or vulnerability. The tension between his claimed trading techniques and his actual methods helps to explain many of his subsequent actions. It can equally well be argued that such tensions are widespread within many financial organizations. Several studies have pointed to the psychological characteristics and personality attributes of successful traders (Tvede, 1999). It has been argued, for instance in Oberlecher's (2004) study of the psychology of foreign exchange markets, that the same attributes that create these tensions between action and appearance may also be important ingredients in the makeup of successful traders. The meta-level point here is that V^2 analysis forces investigators to consider whether or not each contributory factor could be considered a potential vulnerability and also whether each event in the context of a security incident might also be labelled a violation. There is no automatic or algorithmic process to support this analysis.

Figure 3 also illustrates the mechanisms that Rusnak used to hide his losses from directional trading on the Yen. These have been briefly outlined in previous sections. Initially, he began by creating a bogus 'deep in the money' option. Recall that such an option has a price that is significantly below the current

spot-price and hence it is high risk for the vendor. Such options attract high premiums, especially if they can be exercised in the short term when the spot price is unlikely to fall below the level of the quoted option. Allfirst, therefore, had a significant potential liability. At the same time, he created a second balancing bogus option with the same counterparty. This is represented in Figure 3 by the violation labelled 'Rusnak creates balancing option as if Allfirst have paid a large premium to buy currency weeks later involving the same counterparty'. This made it look like Allfirst's original liability was offset by the asset value of the second option. Allfirst should have paid a correspondingly large premium to obtain this second option even though no cash would actually have changed hands because the two premiums balanced each other and were drawn against the same parties. The crucial difference between these options was that the first one, representing Allfirst's liability, was set up to expire within 24 hours. The second, representing Allfirst's fictitious asset, expired several weeks later. Rusnak knew that neither option would ever be exercised because they were bogus deals. However, for the period between the expiry on the first option and the end of the second, he was able to create the appearance of a genuine asset on the Allfirst books. This could be used to offset his genuine losses.

These deals made no sense for a number of reasons. Firstly, the risk exposure on each of the options was quite different given that one expired in 24 hours while the second typically lasted for several weeks. In such circumstances, the options should have attracted very different premiums and so were unlikely to balance each other out. Secondly, the 'deep in the money' options involved in the first bogus trade should have been exercised by the counterparty. A series of similar options failing to be acted upon should have alerted company management to potential fraud. However, as Figure 3 also shows, Allfirst managers did not have access to a list of those options that had expired without being exercised within 24 hours of them being placed. This is denoted by the vulnerability on the left hand side of the V^2 diagram. Prior to September 1998, Rusnak covered his tracks by creating bogus confirmations from the supposed counterparties to these transactions. The confirmations were intended to provide evidence that both parties had agreed upon these trade options. After that time, Rusnak managed to persuade the back-office staff not to pursue these confirmations for his trading activities. As can be seen from the V^2 diagram, their failure to confirm the transactions is partly explained by the difficulty of establishing contact with many of Rusnak's brokers who worked in the Asian offices of the counterparties. The trader's office hours often created considerable communications difficulties for Allfirst's back-office staff. Figure 3 also uses a triangle continuation symbol, labeled with a '2', to carry the analysis from the events surrounding Rusnak's appointment to the start of his fraud. As can be seen, flaws in the reporting and monitoring procedures for Rusnak's activities made it more likely that he would be able to persuade back-office staff not to confirm the matching pairs of bogus trades. These flaws stemmed in part from senior management's desire to support his 'novel' forms of arbitrage.

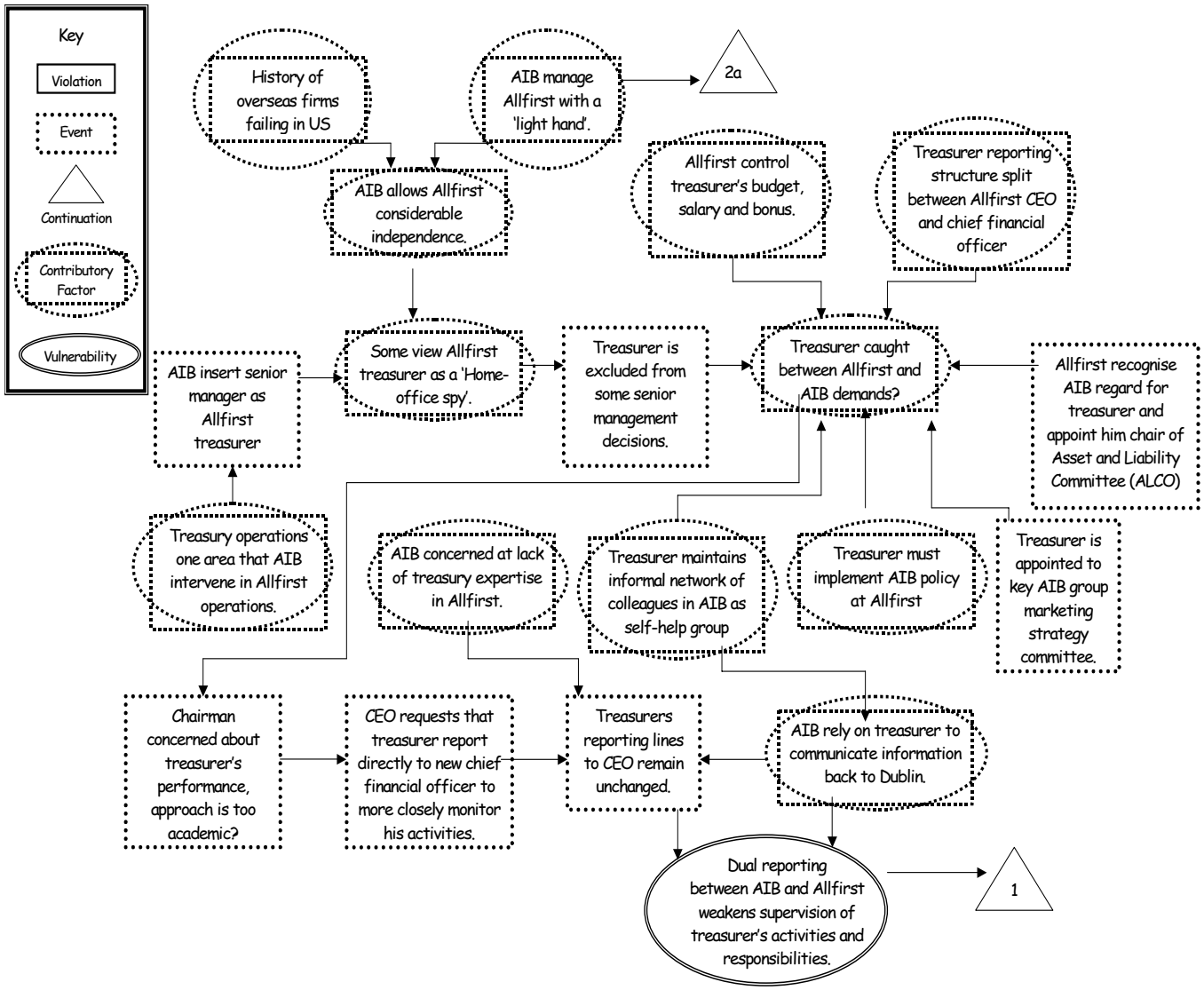


Figure 1: A V^2 Diagram of the Background to the Allfirst Fraud

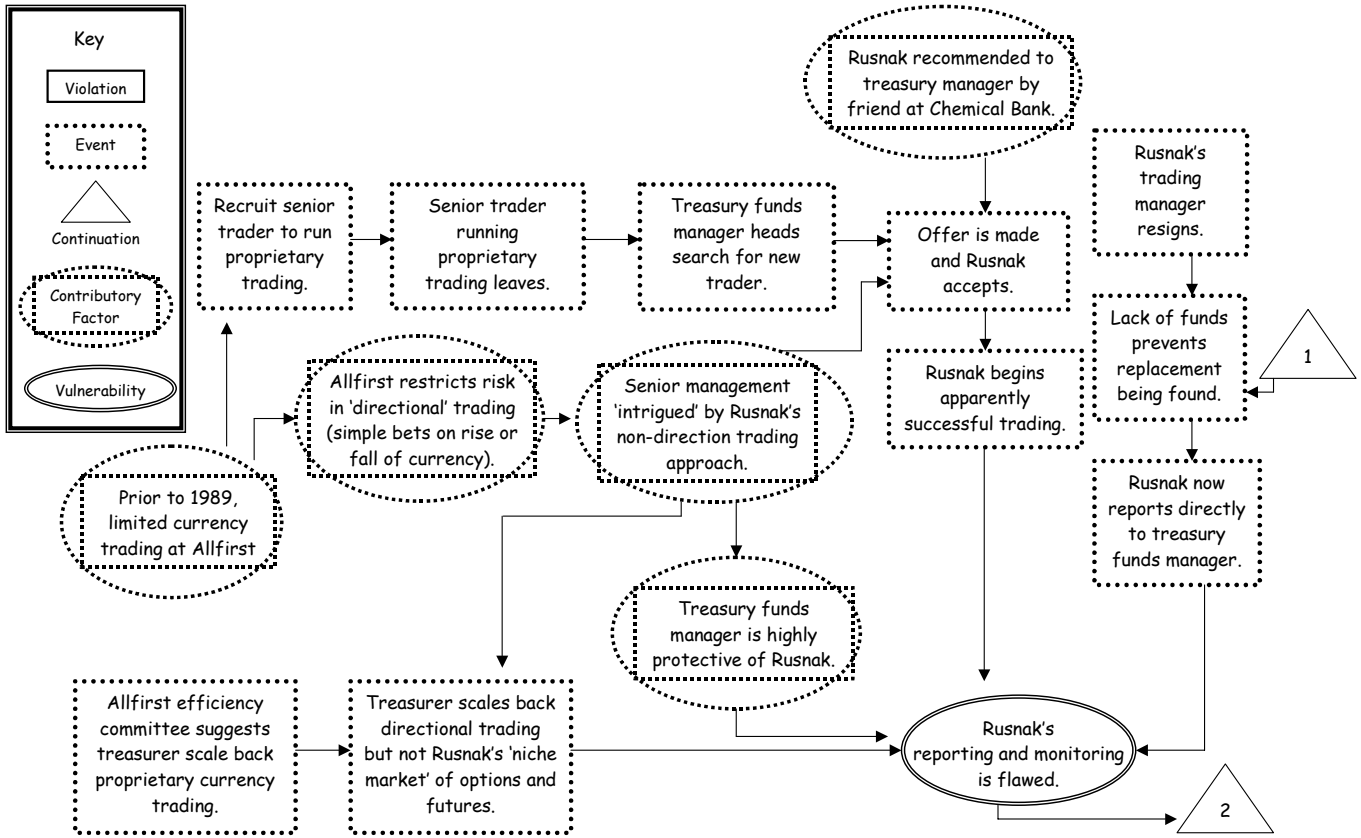


Figure 2: A V^2 Diagram of the Events Leading to Rusnak's Appointment and Flaws in his Reporting Structure

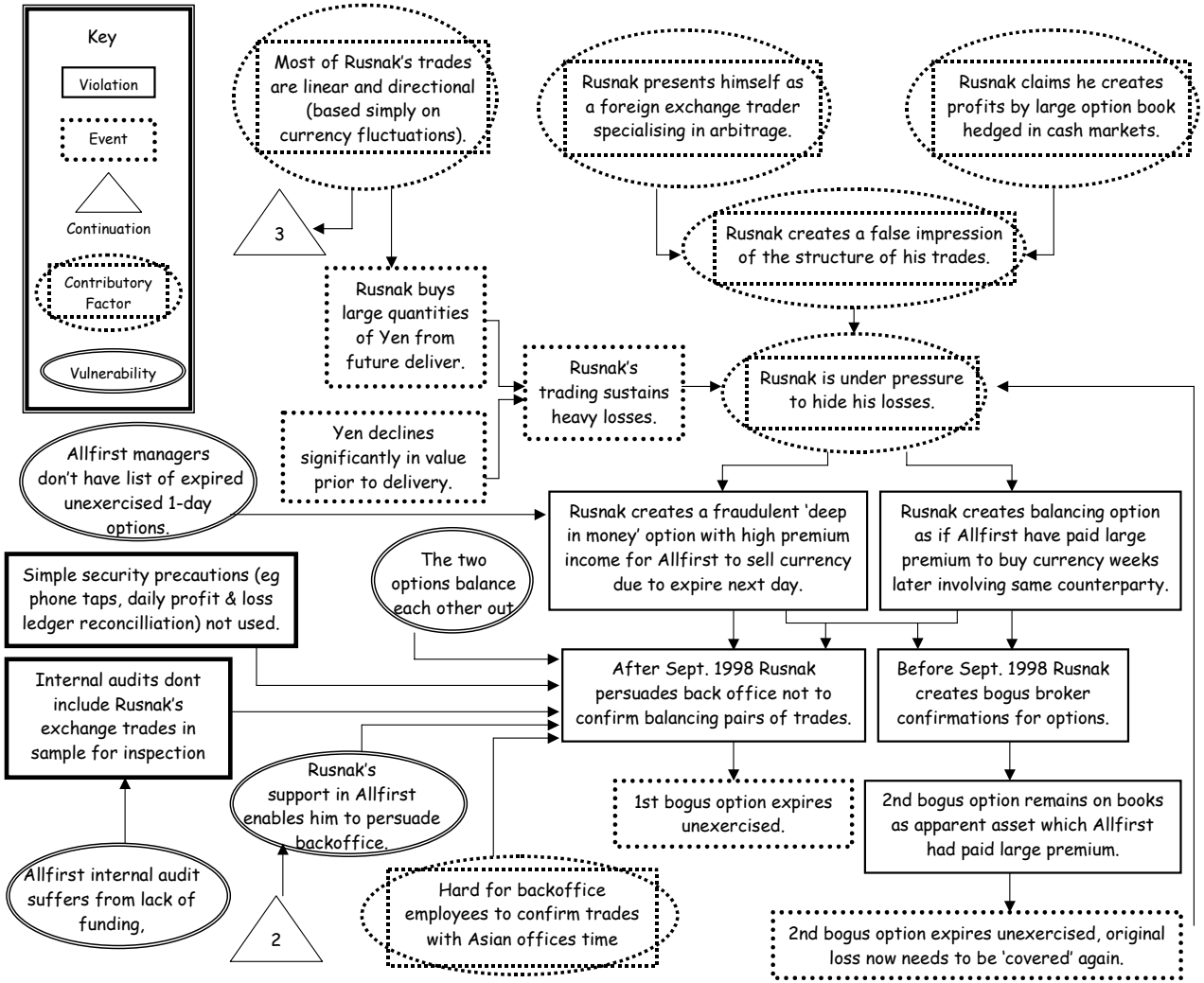


Figure 3: A V^2 Diagram of Rusnak's Initial Balanced-Options Fraud

Figure 4 shows how Rusnak exploited further opportunities to expand both his trading activities and the range of bogus trades that were required to conceal his mounting losses. The top right event in Figure 4 denotes that Rusnak was offered net settlement agreements with a number of financial institutions (Promontory, 2002). These eventually developed into 'prime brokerage accounts'. Such facilities enabled the broker to settle spot foreign exchange transactions with the counterparties. Each of these individual trades was then rolled together into as larger forward transaction between the broker and Allfirst that could be settled on a fixed date every month. As can be seen, these agreements simplified multiple transactions between Allfirst and the counterparties into a smaller number of larger transactions with the brokers. This simplification had two effects. Firstly it reduced the number of operations for the Allfirst back-office. Secondly, it made it difficult for the back-office and others within Allfirst from monitoring the individual trades that were being roller together within Rusnak's prime brokerage accounts. This potential vulnerability is represented half way down Figure 4 on the right hand side. The problems of monitoring transactions through the prime brokerage accounts together with the ability to roll together individual transactions for periodic settlement together combined to create a situation in which Rusnak could exceed the limits on his trading that were routinely insisted upon by Allfirst. His ability to increase the scope and scale of his trading is shown in Figure 4 to have increased the amounts of his loses in both forward and spot transactions. In order to cover his losses, another cycle emerged in which he generated more bogus transactions using the balancing options approach, described in previous sections. Rusnak was also able to exploit vulnerabilities in the DEVON software. This was used to track trades across the prime brokerage accounts. He was able to enter bogus transactions into the system and then reverse them before the monthly settlement period. As can be seen, however, Figure 4 does not provide sufficient details about the nature of the underlying problems with the DEVON application. The vulnerability symbol is annotated with the comment; 'DEVON system vulnerabilities (further analysis?)'. The V^2 notation could be revised to explicitly represent this need for additional analysis. More symbols could be used to show those events and contextual factors, violations and vulnerabilities that have only been partially analyzed. This has not been done, however, in order to minimize the amount of investment that must be made in training to both read and eventually develop these diagrams.

The right-hand, lower portion of Figure 4 illustrates a series of events that threatened Rusnak's activities. It began when the Allfirst treasurer decided to introduce a charge on those activities that used the bank's balance sheet. Such a change would provide greater accountability, for example by exposing whether the profits generated by an activity actually justified the work created for those who must maintain the balance sheet. Questions began to be asked about whether the apparent profits from Rusnak's activities could justify his use of the balance sheet. The total volume of currency traded had risen rapidly over the year to January 2001 but net trading income remained almost the same. A significant proportion of this rise can be attributed to Rusnak's various trading activities. He was, therefore, told to reduce his use of the balance sheet. This not only curtailed his legitimate trading activities but also placed tight constraints on many of the bogus trades, even if many of those trades only made a fleeting appearance on the Allfrist books before being reversed. He had to identify an alternate source of funds to offset his previous losses and those that continued to accrue from his legitimate trading activities.

Figure 5 traces the Allfirst fraud from the point at which senior management began to question Rusnak's use of the bank's balance sheet. This is denoted by the continuation symbol, labeled 4, connecting this image with the V^2 diagram in Figure 4. Rusnak's need to find an alternate source of funds led him to sell long-term options that were deep in the money. As mentioned previously, these options quoted a strike price that was far above the currency's current spot price. Hence, the options represented a relatively high-risk for Allfirst and attracted a corresponding premium. However, Figure 5 also uses a contributory factor to denote that these 'deep in the money options can be viewed as a form of loan' and that 'Rusnak would need to get these liabilities off the books'. Allfirst would have to redeem them when the options were redeemed. Figure 5 denotes a further violation as Rusnak created bogus transactions to indicate that the original options had been repurchased. These activities again involved Rusnak's use of the balance sheet and so the Allfirst treasurer placed a limit of $150 million on his trades.

Previous V^2 diagrams have shown how Rusnak was able to manipulate the DEVON system to conceal some of his transactions via the prime brokerage accounts. Figure 5 shows some of the consequences of these manipulations through the continuation symbol, labeled 5, that links back to the previous diagram. The misuse of the DEVON system, combined with the 'bogus' repurchasing of 'deep in the money' options distorted the Value at Risk (VaR) calculations that were introduced in previous sections. Figure 5 also illustrates further ways in which this risk assessment tool was undermined. Rusnak used 'holdover transactions' to disguise some of his trades. These transactions usually occurred after it was possible for them to be included in the day's accounts. They were, therefore, held over until they could be processed during the next trading day. Internal audit and risk control were aware that Rusnak was responsible for a large number of these transactions but they did not investigate. This observation is illustrated by the vulnerability at the top right of Figure 5. Holdover transactions were not entered directly onto the bank's trading software. There were no checks to determine whether transactions were actually entered into the following day's trading. All of these vulnerabilities can be seen as causal factors in a violation of audit procedures whereby Rusnak directly supplied risk group employees with on-line data for his holdover transactions.

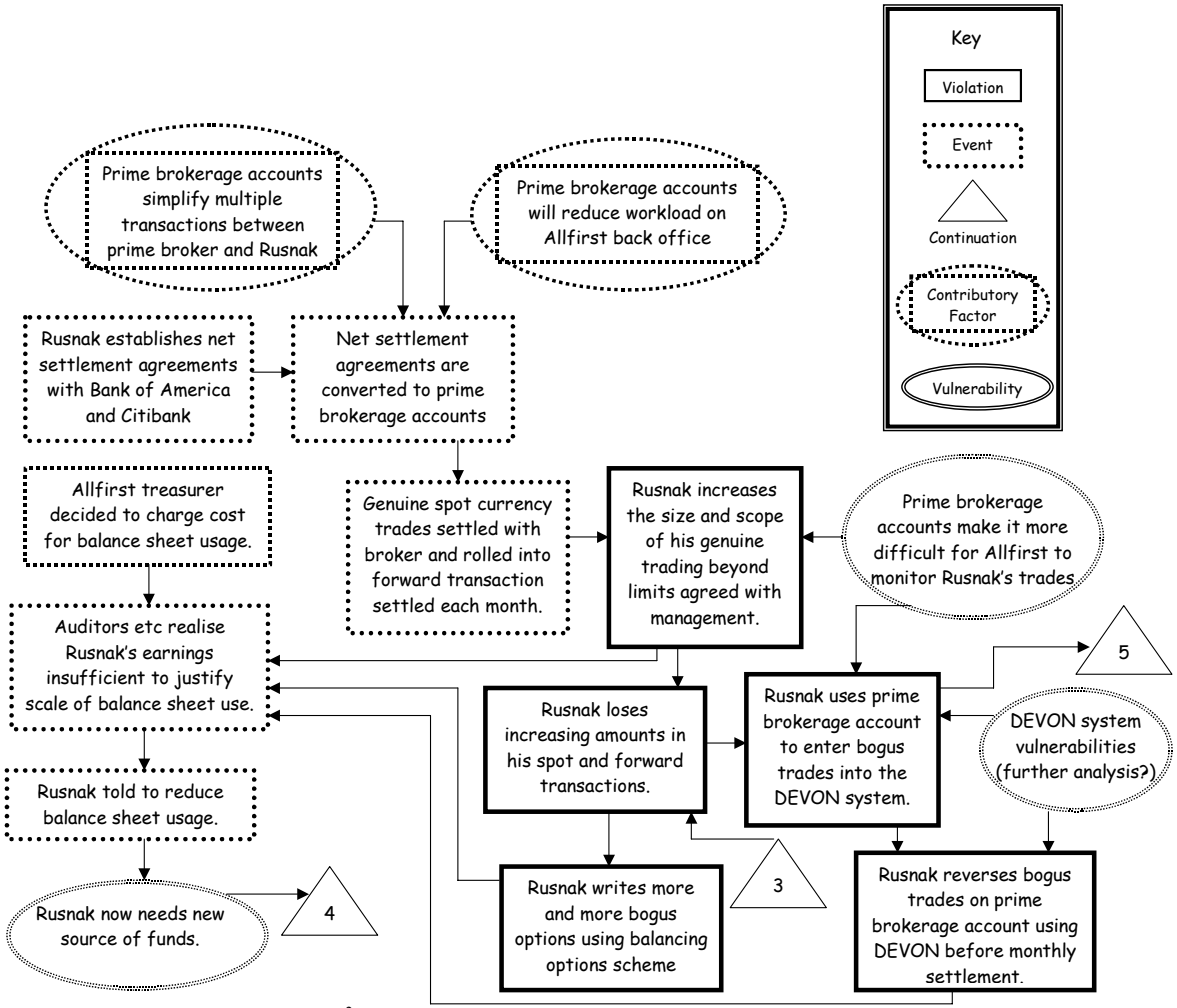


Figure 4: A V^2 Diagram of Rusnak's Manipulation of Prime Brokerage Accounts

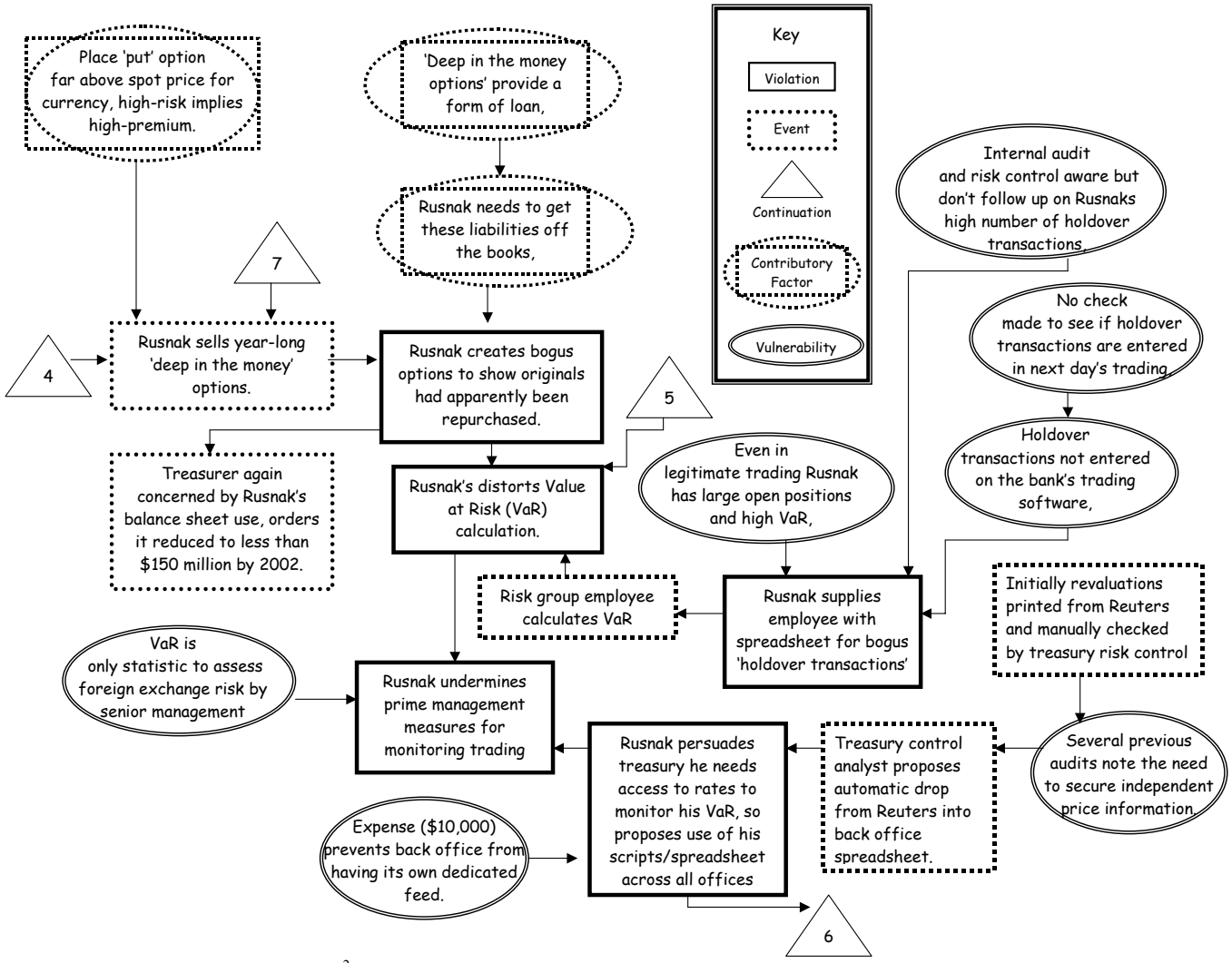


Figure 5: A V^2 Diagram of Rusnak's 'Deep in the Money' Options and the VaR Calculations

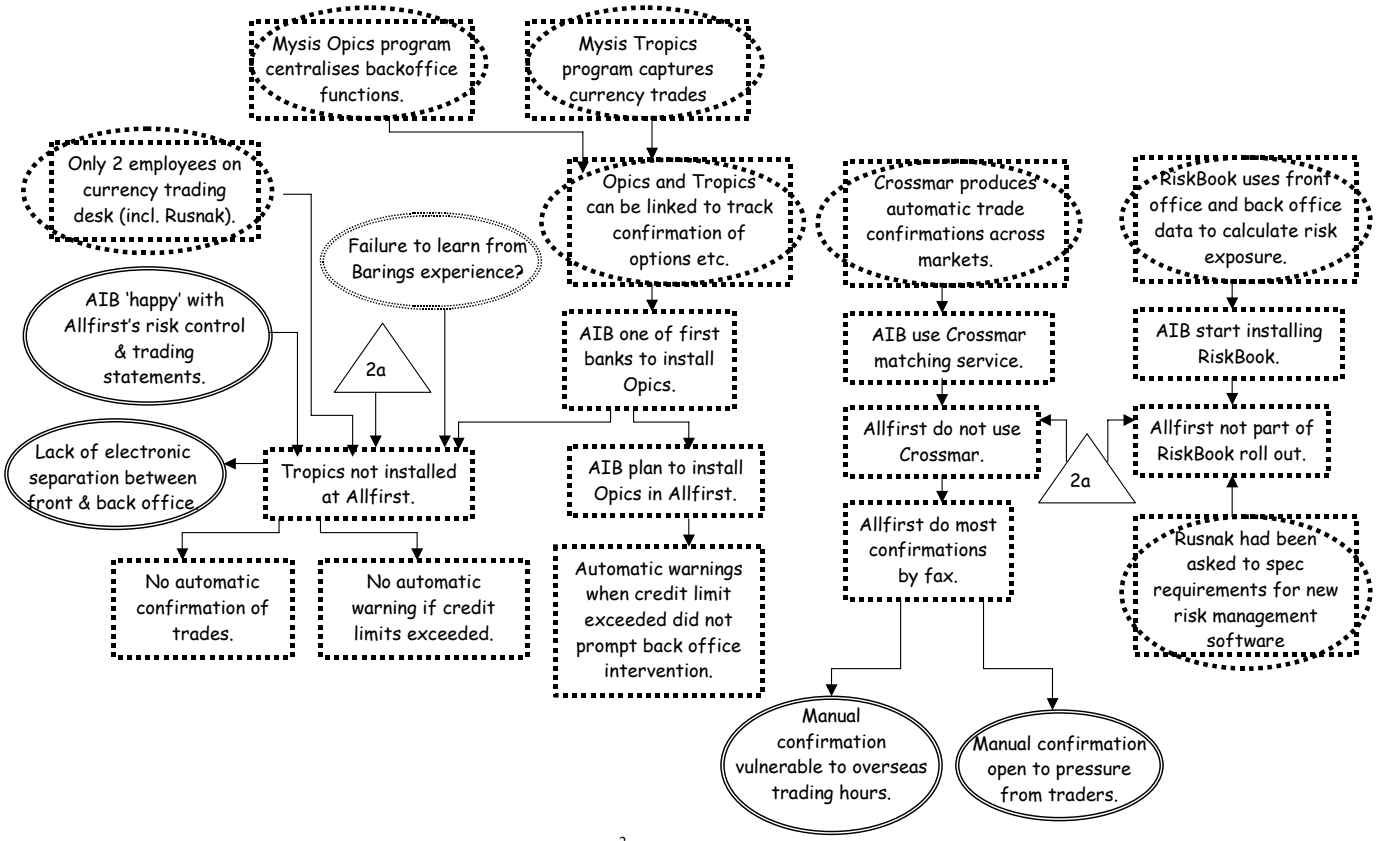


Figure 6: A V^2 Diagram of Software Issues

V^2 diagrams can also focus in on particular aspects of a security related incident. For example, Figure 6 shows how a V^2 diagram can be constructed to look more narrowly at the role that software based systems played in the fraud. This is particularly important given continuing concerns about the management and oversight of access provided by this class of applications. The continuation symbol labeled 2a refers back to Figure 1. This described some of the contextual factors that stemmed from the merger between Allfirst and AIB. In particular, it relates to AIB's decision that Allfirst should be allowed considerable independence and that the new acquisition should be managed with a 'light hand'. AIB had been one of the first banks to invest in a software system called Opics. The Opics application automates and centralizes a number of back-office functions. It can also be used in conjunction with a 'sister-application' known as Tropics that supports currency trading. An important benefit of using these applications together is that they can enforce a separation of back-office and front-office activities. They can also be used to trace the confirmation of options that were created by the front-office staff and should have been monitored by back-office employees. Tropics was not installed at Allfirst. Hence the software did not support the tracking and clear division of responsibilities that might have prevented many of the vulnerabilities and violations that were identified in previous V^2 diagrams. As can be seen in Figure 6, the decision not to install Tropics was justified on many grounds. Firstly, the costs of the software may not have been justified by the relatively small size of the trading desk. Also, at the time of merger AIB appeared to be happy with the Allfirst risk control and trading statements. They arguably did not see any justification for the additional monitoring facilities provided by the Tropics application. The decision to invest in Tropics can also be partly explained by a failure to learn from the Barings experience where a trader had managed to erode the separation between front and back office functions. Finally, there was no tradition for preserving this separation in terms of the electronic systems that support the work of Allfirst staff. The outcomes from the decision not to install Tropics included the lack of any automatic confirmation for trades. The decision not to install Tropics also prevented any automatic warnings for traders when their activities exceeded credit limits.

Figure 6 illustrates how V^2 diagrams can be used to gradually piece together more detailed information from a variety of sources. These included the official initial investigation (Promontory, 2002) as well as a number of subsequent reports (Gallager 2002, de Fontnouvelle, Rosengren, DeJesus-Rueff and Jordan, 2004). These sources reveal that Allfirst did go ahead with the installation of the Opics back-office modules associated with the Tropics front-office application. This did help to generate warnings when credit limits were exceeded. However, as we have seen, a host of technical and organizational factors persuaded the back-office staff that these warnings indicated numerous trader errors rather than significant alarms about bogus trading activities.

In addition to the Opics and Tropics systems, Allfirst might have been protected by the introduction of the Crossmar software that was used by AIB. This application also provided automated confirmation for trades using a matching service. Allfirst did not use the Crossmar software and so most of the confirmation relied upon back-office staff to fax requests to overseas markets. This manual confirmation was vulnerable to interruption and dislocation due to overseas trading hours. It was also open to pressure from traders such as Rusnak. Although we have not included it in the current analysis, Figure 6 might also be extended to illustrate the additional pressures that Rusnak's activities created for the back-office staff. His bogus options relied upon the continual generation of additional transactions beyond his legitimate trading activity. One side-effect of the fraud would, therefore, have been to increase the workload on back-office staff which in turn may have left them even more vulnerable to attempts to delay or ignore confirmations on a rising number of trades. AIB had also decided to exploit a software application known as RiskBook. This uses front and back-office systems to calculate the bank's risk exposure. Previous sections have described how Rusnak was able to affect the VaR calculations and there is reason to suppose that the use RiskBook might have offered some protection against these actions. Allfirst were not, however, part of the first roll-out for the RiskBook software within Allfirst. It is deeply ironic that Rusnak had been asked to specify the requirements for this new risk management software.

Conclusions and Further Work

A number of commercial and governmental organizations have recently argued that we must look beyond the immediate events that surround security-related incidents if we are to address underlying vulnerabilities (Austin and Darby, 2003). It is important to look beyond the immediate acts of 'rogue traders' or individual employees if we are to correct the technical and managerial flaws that provide the opportunities for security to be compromised. This paper has, therefore, provides an introduction to Violation and Vulnerability analysis using V^2 diagrams. The key components of this technique are deliberately very simple; the intention is to minimize the time taken to learn how to read and construct these figures. The paper has, in contrast, been motivated by a complex case study. The intention has been to provide a sustained example at a level of detail that is appropriate to an initial investigation into complex security incidents. Previous pages have provided a sustained analysis of Rusnak's fraudulent transactions involving the Allfirst bank. This case study is appropriate because it involved many different

violations and vulnerabilities. These included failures in the underlying audit and control mechanisms. They included individual violations, including the generation of bogus options. There were also tertiary failures in terms of the investigatory processes that might have uncovered the fraud long before bank personnel eventually detected it.

Much remains to be done. We are currently working with a number of organizations to extend and tailor the techniques in this paper to support security investigations in a range of different fields, including both financial and military systems. There is a common concern that the V^2 approach will provide a standard means of representing and modelling the outputs of an investigation into the causes of security-related incidents. In each case, however, we are being encouraged to extend the range of symbols represented in the diagrams. For example, these might be used to distinguish between different types of barriers that should have led to the identification of a violation or vulnerability. In terms of the Allfirst case study, the decision not to tell senior management about concerns over the Reuter's currency feed via Rusnak's PC would have to be represented using a different type of symbol. The intention is that analysts would then be encouraged to probe more deeply into the reasons why this potential warning was not acted upon. An important concern in this continuing work is, however, that the additional notational elements will increase the complexity of what is a deliberately simple approach. It is critical to avoid additional complexity in the analysis of what are almost always extremely complex events.

Further work also intends to explore the use of V^2 diagrams as a communication tool with wider applications. In particular, the outcomes of many security investigations must be communicated to diverse groups of stakeholders. These are not simply confined to security professionals and senior management in the target applications. In particular, it is often necessary to communicate findings about the course of an incident with members of the public who may potentially be called upon to act as jurors in subsequent litigation. The complexity of many recent security related incidents makes it vitally important that we find the means to help people understand the events and contributory factors that form the context for many adverse events. Similarly, political intervention is often triggered by incidents such as the Allfirst fraud. It can be difficult to draft effective legislation when key figures lack the necessary time and briefing material to fully follow the events that they seek to prevent.

References

R.D. Austin and C.A.R. Darby, The Myth of Secure Computing, Harvard Business Review, (81)6:120-126, 2003.

BBC News, Bank sues over $700m fraud, British Broadcasting Company, London, BBC On-Line, 23 May 2003.

Cisco, Network Security Policy: Best Practices White Paper, Technical report number 13601, Cisco Systems Inc., San Jose, USA, 2003.

US Department of Energy, Root Cause Analysis Guidance Document, Office of Nuclear Safety Policy and Standards, Guide DOE-NE-STD-1004-92, Washington DC, 1992.

US Department of Energy, DOE Standard Safeguard and Security Functional Area, DOE Defense Nuclear Facilities Technical Personnel, Standard DOE–STD–1171–2003, Washington DC, 2003.

P. de Fontnouvelle, E. Rosengren, V. DeJesus-Rueff, J. Jordan, Capital and Risk: New Evidence on Implications of Large Operational Losses, Federal Reserve Bank of Boston, Boston MA, Technical Report, 2004.

S. Gallacher, Allfirst Financial: Out of Control, Baseline: Project Management Information, Ziff Davis Media, March 2002.

G.L. Jones, Nuclear Security: Improvements Needed in DOE's Safeguards and Security Oversight, US General Accounting Office, Washington DC, Report GAO/RCED-00-62, 2000.

C.W. Johnson, A Handbook of Incident and Accident Reporting, Glasgow University Press, Glasgow, Scotland, 2003.

K. Julisch, Clustering intrusion detection alarms to support root cause analysis. ACM Transactions on Information and System Security, (6)4:443–471, 2003

G. Killcrece, K.-P. Kossakowski, R. Ruefle, M. Zajicek, Organizational Models for Computer Security Incident Response Teams (CSIRTs), Technical Report CMU/SEI-2003-HB-001, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, USA, 2003.

J. Lew, Guidance On Implementing the Government Information Security Reform Act, Memorandum for the Heads of Departments and Executive Agencies, Whitehouse Memorandum M-01-08, Washington DC, 2001.

C.A. Meissner and S.M. Kassin, "He's guilty!": investigator bias in judgments of truth and deception. Law and Human Behavior, 26(5):469-80, 2002.

Microsoft, Microsoft Solutions for Securing Windows 2000 Server, Microsoft Product & Technology Security Center, Redmond USA, 2003. Available from http://www.microsoft.com/technet/security/prodtech/win2000/secwin2k/default.mspx

Naval Surface Warfare Centre, Dahlgren, Computer Security Incident Handling Guidelines, Department of the Navy, Commanding Officer, Fleet Information Warfare Center, Virginia, USA, 2002.
T. Oberlechner, The Psychology of the Foreign Exchange Market, John Wiley and Sons, New York, USA, 2004.
Promontory Financial Group, Report to the Board and Directors of Allied Irish Bank PLC, Allfirst Financial Inc. and Allfirst Bank Concerning Currency Trading Losses Submitted by Promontory Financial Group and Wachtell, Lipton, Rosen and Katz, First published by Allied Irish Banks PLC, Dublin, Ireland, March 2002.
A.M. Rabinowitz, The Causes and Some Possible Cures: Rebuilding Public Confidence in Auditors and Organizational Controls
Certified Public Accountants Journal, 66(1):30-34 1996.
J. Reason. Managing the Risks of Organizational Accidents. Ashgate Publishing, Aldershot, 1997.
K. Roark, Los Alamos Director Testifies on Security Incident, Lawrence Livermore National Laboratory, Press Release, Livermore, California, USA, June 2000.
S. Skalak, Financial Fraud: Understanding the Root Causes, Price Waterhouse Cooper, Financial Advisory Services, Dispute Analysis & Investigations Department (2003).
P. Stephenson, Modeling of Post-Incident Root Cause Analysis, International Journal of Digital Evidence, (2)2:1-16, 2003.
L. Tvede, The Psychology of Finance, John Wiley and Sons, New York, 1999
M.J. West-Brown, D. Stikvoort, K.-P. Kossakowski, G. Killcrece, R. Ruefle, M. Zajicek, Handbook for Computer Security Incident Response Teams (CSIRTs), Technical Report CMU/SEI-2003-HB-002, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, USA, 2003.

Complexities of Multi-organisational Error Management

John Dobson*, Simon Lock, David Martin

Dept. of Computing Science, University of Lancaster, Lancaster LA1 4YW, U.K.

Abstract: In this paper we shall look at some of the problems in designing an information and communication (ICT) system for an organisation located in a complex multi-organisational setting. We shall look in particular at the handling of errors both within the ICT itself and in the complex multi-organisational activities which the ICT is designed to support. We shall conclude by offering some advice to system designers which should prevent them from repeating mistakes which have been made all too often before.

Keywords: responsibility modelling, organisational boundaries, error management

Introduction

This paper looks at organisational complexity and the problems it raises for the design of information and communication systems. Specifically, we shall look at problems arising from complex patterns of responsibility that are shared between separate organisations engaged in a joint enterprise or between whom some relationship exists, and the issues that arise when these shared responsibilities are called into play either to prevent failure or as a consequence of failure. We shall look at the stresses that shared responsibilities place on information and communication systems and indicate an approach to dealing with them.

One clarification is necessary at the start. When we use the term "information and communication system" we are not assuming anything about the extent to which it has been computerised. Information systems are taken to include not only paper records but also individual and organisational memory. Similarly, communication systems are taken to include teleconferencing and face-to-face meetings. To stress this point, we have deliberately chosen to illustrate our points by reference to an example in which serious failings in the information and communication systems were uncovered, though no computer systems were implicated. More will be said about this later, when the case study is introduced. However, our recommendations and conclusions are intended to be applied to systems built using information and communication technology; we are hoping to show how the kinds of problems that such systems have to deal with requires a certain reconceptualisation of information and approach to design when complex shared responsibilities are to be supported.

Many ICT systems are designed for a context which is restricted to the organisation that deploys them. This is often an oversimplification since organisations often do not work as a closed system with relationships confined to defined interfaces. Standard system design paradigms are not well adapted to designing for systems to be deployed in complex multi-organisational settings, often because the procurement process is specifically designed to exclude this. Procurement is thought of as a single contract between a single purchasing organisation and a single supplier (though the supplier may be a consortium of organisations).
This model works well when the nature of the goods to be supplied is well-understood ("Please supply half a ton of broken biscuits"), but fails when the relationship between the parties is more complex than a consumer-supplier one, or when it involves something more complex than goods, or when recovery from failure is problematical and involves society as a whole, not just the interested parties.

To make this clear, here are three examples of organisational relationships that are well-understood and standard system design paradigms can be made to work quite well: support for consumer-supplier relationships; implementation of straightforward financial transactions; licence-handling applications. Here, by contrast, are three examples of more complex multi-organisational systems where standard system design paradigms have been found not to work too well: systems to support healthcare for the citizen; integrated transport systems; military command and control systems.

These systems are all complex because they all include patterns of shared responsibilities which are implicit, negotiated and dynamic; and, as we shall see, it is often not until a failure occurs that the full complexity of these

* Contact address: 31 Wentworth Park, Allendale, Northumberland NE47 9DR, U.K.
email J.E.Dobson@ncl.ac.uk

shared responsibilities is appreciated, and the simplified assumptions about them that are implicit in the information and communication systems that support the joint enterprise are exposed and break down. This make such systems hard to design because more attention has to be paid to what happens when things go wrong. It is in the presence of failure that responsibilities are assumed, rearticulated and renegotiated; this requires flexibility of role definitions and organisational boundaries. Rigidity of boundaries and interface definitions so often serve to prevent recoverability.

Many system design methods start from the assumption that the functionality of the system can be defined in terms of activities that are expressed in terms of their behaviour when accessed through defined interfaces. Although this is a simplified model of the way real things work in the real world, it works well enough as a representation when things are working correctly in a stable well-understood environment.
But in the kinds of complex multi-organisational systems we are considering, when things do not work correctly, human ingenuity is often able to conceive some kind of workaround which may well require some extension or renegotiation of responsibilities, and this in turn may require some adaption of the information and communication system representation of the real world entities.

An Illustrative Example
One of the best simple examples of multi-organisational complexity is the relationship between Railtrack and the train operating companies as illustrated by the Ladbroke Grove disaster.[3]

> *On 5 October 1999, a train operated by Thames Trains and driven by a newly qualified driver passed a red signal (referred to as SN109) at Ladbroke Grove (just outside Paddington main line station, London) and continued for some 700 metres into the path of a high speed train. As a result of the collision and the ensuing fires, 31 people died and 227 were taken to hospital.*
>
> *A subsequent investigation identified a number of significant factors which resulted in the signalling in the Paddington area not always being compliant with relevant industry standards. Signal sighting experts came to the overall conclusion that the viewing conditions at the relevant signal presented an exceptionally difficult signal reading task.*
>
> *The reasons why the train passed the red light are complex. There were no indications that the driver deliberately set out to pass the signal at red, and the investigation concluded that any acts or omissions by him were just one group of contributory factors.*
>
> *A full report of the enquiry is available at*
> *//www.hse.gov.uk/railways/ladbrokegrove.htm*

As we shall show, the information and communication systems in these organisations, though partly manual, were deficient. There is no reason to believe that fully automated systems would have been any better, given the system design paradigms for computer-based systems prevalent at the time.

Dependability Basics
In this section we introduce, with examples, some basic vocabulary for talking about dependability. The terms used are standard in the domain of dependability of computer-based systems, but we will explain them in the context of any socio-technical system, including those in which the technology is not computer-based (or can be so regarded: computers are in fact used in signalling systems, but in the Ladbroke Grove case this was completely irrelevant).

FAULT: a fault is something that is not as it should be; a state. It may be a state of a human or of a machine. It may be latent (not visible) and it may be benign (does not cause an error).

ERROR: an error is the manifestation of a fault, and is a behaviour of something. Often an error is a manifestation of an interaction between two or more faults.

CONSEQUENCE: a consequence is the observable effect or outcome of an error.

FAILURE: a failure has occurred when a undesirable consequence is experienced. It is a judgement about erroneous behaviour, based on its consequences.

[3] [For readers not familiar with the UK railway system it will help to know that at the time of the disaster, railway responsibility was divided between a single organisation (Railtrack) responsible for the entire infrastructure of track, signalling, bridges and property, and a number of train operating companies (of whom Thames Trains was one) responsible for operating and maintaining rolling stock and the conveyance of passengers or goods. A regulator was appointed to oversee that the whole system worked in the public interest, to award franchises to the operating companies, and to enforce public policy in the areas of competition and cooperation.]

We shall endeavour to be quite consistent in our usage, which is based on strong typing: a fault is a state, an error is a behaviour, a consequence is a result (a causal concept), a failure is a judgement.

We shall also introduce some terms associated with the achievement of dependability. Faults can be *avoided* during the creation or implementation of system components. Faults can be *removed* from components after they have been created or implemented. Faults can also be *tolerated*, which means that if an error occurs and is detected as such, some recovery or workaround is initiated which prevents the error from causing a consequence judged to be a failure. This implies the need for monitoring and exception handling. The risk of failure can also be *accepted*, with the cost of failure (if it occurs) being met, for example through insurance or compensation or writing off. Acceptance of risk can be transferred to users or operators through the use of disclaimers and warning signs.

Example: There are many possible accounts of an incident which leads to an adjudged failure, taken from different viewpoints. Indeed, though not here, some accounts may lead to the view that a consequence was not, in fact, a failure — since a different judge is making the judgement.

One possible account of Ladbroke Grove is that which places the responsibility with the train operator (there are others, equally valid):

FAULT	*a poorly trained driver*
ERROR	*a signal passed at danger (called a SPAD)*
FAILURE	*a crash*

Another possible account is that which places the responsibility with the infrastructure provider:

FAULT	*a badly designed and/or positioned operational signal*
FAULT	*Inadequate monitoring and countermeasure guidelines and practice*
ERROR	*SN109 not identified as dangerous due to poor monitoring and countermeasure processes*
CONSEQUENCE	*the continued use in operation of SN109*

We now briefly look at the mechanisms in place that were intended to achieve dependability of the system.

Operational faults

Removal	It was assumed that (re)training and information would remove driver errors due to faults in insufficient skill and knowledge
Tolerance	It was assumed that an automatic warning system in the driver's cab and a 700-yard run-on (between the signal and the points it controlled) would be sufficient to allow error recovery and 700 yard run on would allow error recovery to avoid failure

Signal design and Placement Faults

Removal	Procedures were in place to identify and rectify problematic signals but a solution had not been found/agreed upon for SN109
Tolerance	In addition to the assumed tolerance of driver error, procedures were in place in the signal control room to detect SPADs and to take appropriate action in signals on the other line(s)

Multiple Faults: We provide a brief summary of the factors which lead to complexity. These points will be expanded upon as we proceed.

A single failure may be the consequence of multiple faults, all acting together. The removal or tolerance (recovery) from a single fault may prevent a subsequent failure occurring. The danger is (especially with multi-organisational systems) that the faults which are not removed of protected against will remain latent and may later become reactivated by changing conditions or the injection of further faults. For example, if the infrastructure provider (namely Railtrack) did all that they could to remove faults from the system, this would at best improve the positioning of the signal, removing only one fault from the system. The adequacy of driver training would not be

affected; indeed, the deficiencies in training might go unnoticed as the improved signal positioning would be likely to reduce or prevent failures.

This brings us to the issue which is at the heart of this paper: the complexity arising from multiple faults situated in different organisations. Examples of this complexity are:

* With different organisations, how do different possible faults interact? Whose responsibility is it to work this out? Who is responsible for the interaction?
* What is the model of the relationship between companies? What is the nature of the contract between them?
* Is the peer relationship of very loose cooperation adequate for creating a safety structure?
* How do faults, error and failure in the system that creates a given system undermine the effectiveness of fault avoidance strategies? In a similar way, how are fault-error-failure chains associated with the other failure management schemes (fault removal, fault tolerance, failure acceptance)?

Responsibilities for Handling Errors

In this section, we shall expand a theme mentioned in the introduction: responsibilities for handling errors. We maintain that in complex multi-organisational settings, failures often occur because mechanisms for sharing responsibilities are inadequate or absent; and this is particularly true for responsibilities for preventing or managing failure. Our paper is concerned with design considerations for such mechanisms.

Causal and consequential responsibility: There are many meanings of the word 'responsibility', which we will not discuss here. (Good books to read on the nature and importance of responsibility are Lucas (1995) and Jonas (1984), respectively.) However, for our present purposes it is useful to distinguish between *causal responsibility*, when an agent has an obligation to make something happen or prevent it from happening or to maintain a state, from *consequential responsibility*, when an agent is answerable when something happens or does not happen or a state is not maintained. These different responsibilities do not always rest on the same agent (the doctrine of 'ministerial responsibility') and consequential responsibility may be held to rest with an organisation as a whole whereas causal responsibility most usually can be traced to an individual or the fact that no particular individual at the time held the responsibility. causal responsibility may sometimes be delegated, though some responsibility remains with the delegating agent (i.e. the responsibility for having chosen to delegate), whereas consequential responsibility is not normally capable of delegation, though it may sometimes be transferred. We shall refer to these different responsibilities in our discussion of Ladbroke Grove, and deal with the complexities they pose for system design in a later section.

Lifecycle and responsibilities: In preparation for mapping out the responsibilities implicated in a failure, it is useful to start by looking at the major life-cycle phases of an operational system as a way of distinguishing different responsibilities. There are four major phases (defined by processes) in the life cycle of an operational system: procurement; operation; maintenance; decommissioning (in the case of Ladbroke Grove, decommissioning was not an issue). It is easier to deal with particular faults in particular ways at particular points in the life-cycle:

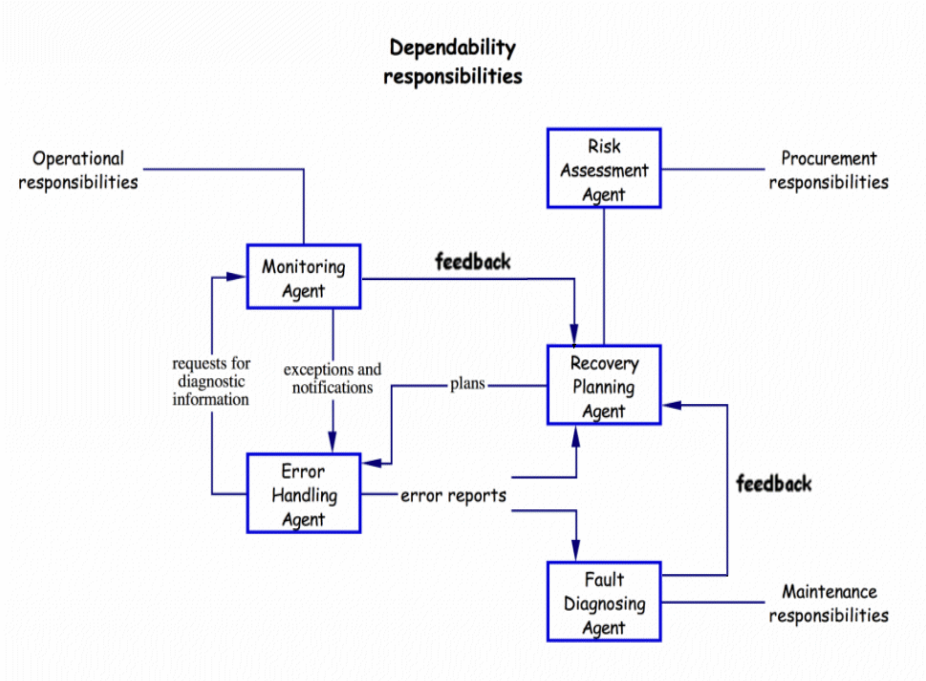
Procurement includes making assessments of the risks and consequences of operational failures.

Operation includes monitoring errors and following plans for recovering from the errors so as to prevent them from giving rise to failures.

Maintenance includes taking retrospective action to prevent subsequent occurrences of F-E-F chains.

Decommissioning includes ensuring that documentation concerning the (in)accuracy of the failure mode assumptions and (un)successful ways discovered of managing failures is preserved for posterity.

The previous analysis leads to the following articulation of overall responsibilities:



The use of the word 'agent' here indicates a responsibility for doing something or seeing that it gets done – the actual execution could be performed by a machine or delegated to humans. An agent is always a person or group of people sharing a responsibility.

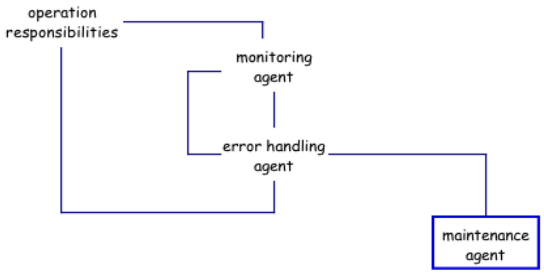
The lines in the diagram represent not just information flows but conversations. A conversation is a possibly extended series of exchanges, distributed in time and space, between two agents. The information exchanged can also be seen as a partial state of that conversation as it exists at any instant. More details about the modelling here presented will appear in a forthcoming book (Clarke and Hardstone 2005).

The picture is intended to be normative. Its use is in performing a comparison with a description of the responsibilities as they are articulated in the actual setting, in order to identify such things as missing or ill-defined responsibilities, or shared responsibilities that cross inter- or intra-organisational boundaries, as it is these that so often give rise to failures, and in particular in failures in failure prevention or management. This comparison can be used, as in the soft systems methodology (see Checkland (1981) and Checkland and Scholes (1990) for the theory and practice of soft systems methodology), as a way of determining expectations on a new information system.

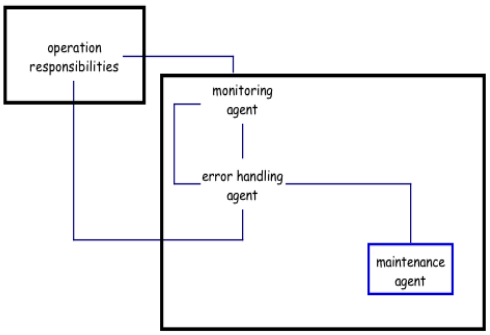
The positioning in this model of (intra- and inter-) organisational boundaries is key to effective error recovery. This will be discussed in the next section.

Organisational Boundaries

In order to discuss the problems arising when responsibilities cross organisational boundaries, we start by taking a slight simplification of the previous figure.

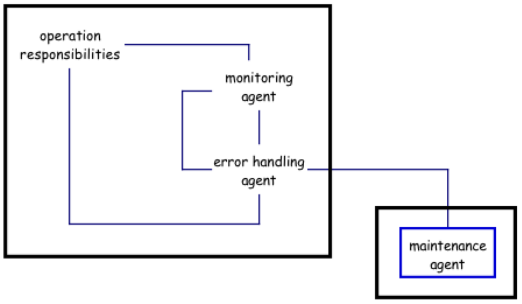


If maintenance responsibilities are in a different enterprise from the operation responsibilities, where exactly does the boundary lie? It could, for example, be like this:

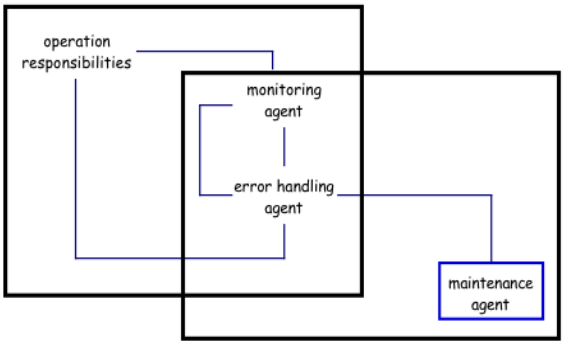


Here, system maintenance is carried out either by a separate organisation or by a separate division within the operating enterprise. As part of the maintenance, all the monitoring responsibilities can be transferred, but the operator is then dependent on another organisation for critical management information; there are a number of possible organisational failures associated with such a critical dependence.

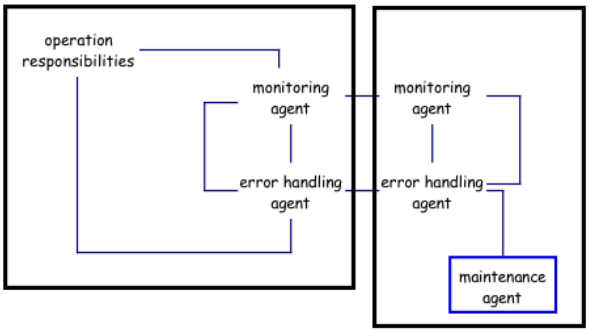
An alternative that is theoretically possible but in practice would be defective, is shown below:



but in practice, maintenance will include at least some monitoring and therefore some error handling:



so that monitoring and error handling responsibilities are shared between the operational organisation and the maintenance organisation, Such shared responsibilities require good communications channels and some way of resolving conflicts in priorities, because this model is equivalent to the following:



The problems here are clear. Inter-organisational conversations are required to coordinate shared responsibilities; but the media and channels required for such co-ordination may be unclear and the supposedly communicating processes may be mutually opaque as indeed they were at Thames Trains and Railtrack, as the Ladbroke Grove enquiry shows.

Boundary Objects

There has been much discussion on the concept and nature of boundary objects, and indeed whether they can easily be determined. In simple cases, however, the idea is a useful one. A boundary object is one which is visible on both sides of an organisational boundary, but which has different connotations on each side. For example, to Railtrack a train is something which uses their infrastructure; to Thames Trains, a train is something which delivers their transport services to their customers. So information about boundary objects is generated in two distinct contexts. Normally, such information is interpreted in the context in which it is generated, though parts of the context may be shared (e.g. the whereabouts on the line of the train).

In the presence of failure, however, when shared responsibilities are actually called upon to be exercised, information generated on one side (e.g. about the training of drivers) has to be interpreted on the other (e.g. was their driver appropriately trained *from our point of view?*) In addition, two other things tend to happen:

i) what constitutes the boundary object is re-articulated (e.g. trains are now seen to have drivers)

ii) things previously not seen as boundary objects now take on that significance (e.g. signals have now to be treated as boundary objects; as indeed have drivers because Railtrack now realises that it has an interest in the driver's training and experience).

There are three distinct, but related, information management problems that now arise:

1) What one party judges to be a failure of, or implicating, a boundary object might not be so judged by the other party (e.g. the fact that drivers had difficulty in reading a particular signal was initially treated by Railtrack as a form of driver failure, not signal failure).
This is a point which is not always appreciated by those who consider a failure a state or behaviour (i.e. something which all competent observers can agree upon), because although it is a mismatch between what actually occurred and what a specification says should occur, there might be more than one valid specification. Clearly, a train crash is a failure, as is a SPAD: but wherein lies the error(s)? And of what is the failure that gave rise to the error(s)? This is the problem of socially determined failures, i.e. consequences whose subsequent characterisation as failures is a process of social agreement.
2) Information on one side of the boundary – including its context of interpretation and generation – might not be visible on the other side. This undoubtedly occurred at Ladbroke Grove. The report comments unfavourably again and again on the way that information passed across the boundary but was not acted on for reasons that were obscure.
3) A shared approach to recoverability or repair might well be hampered by the invisibility of relevant information or its processing.

These problems are deep-rooted and give rise to a number of issues in the procurement and design processes for an information management system, to which we shall now turn.

Some Implications for Design
Agency: The binding between individuals and responsibilities is a complex many-to-many relationship. We structure this using two distinct concepts. We have introduced the concept of *role* to classify the relationships held by an individual: an individual can hold many roles simultaneously, and a role may imply many related responsibilities. The concept of role is related to the structure of an organisation and is defined in organisational terms. *Agency*, on the other hand, is abstracted away from any actual organisational structure and is simply a collection of related responsibilities; how it maps onto work roles is a matter of organisational design and will change as the organisation changes. In particular, one particular agency may span inter-organisational boundaries, such as the consequential responsibility for a collision.

The concept of agency allows a conceptual separation to occur as organisations change. For example, small organisations often combine the sales agency and the marketing agency into the same role; as the organisation grows, this is often a desirable separation to make. Since agency is a more stable concept than role, an information system based on agency rather than role is more likely to be capable of change as the organisational structure changes.

Conversations: Dealing with multi-organisational failure and its consequences requires communication and cooperation. This implies that information, as well as being about the state of a boundary object, is also the (partial) state of a conversation between the communicating and cooperating parties. This means that an information system is sometimes better reconceptualised as a communication system ,and this in turn requires a reconceptualisation of communication and conversation), one that provides a basis for undestanding failure modes.

Conversations and the relationships that they define, sometimes fail. The purpose of a theory of conversations is to explain the failures associated with the intentions of the participants. It is clear that the bringing together of obligations and responsibilities can create conflicts of interest as well as synergies. It can also create overloads and imbalances which could lead to failure in operation. In addition to failures of organisational policy and design, we have operational failures due to a lack of correspondence between the expectations of the participants. We have developed a theory of the attributes of roles and conversations that provide a basis for analysing such situations. We have also developed an analysis of failures to perform the intended role by failing to generate correct information, by misinterpreting presentations or by proffering incorrect or inappropriate resources. These failures would be accounted for in a theory of instruments. Finally, failures in reading, writing or transporting data are the province of a theory of communication.

The need to record: Responsibility modelling raises three important information questions: What do I need to know? What do I need to do? What do I need to record to show what I have done? It is this last that becomes of importance when the possibility of failure raises questions of answerability. Recording can be seen as an anticipated conversation between a responsibility holder and an investigator. For example, one possible organisational policy is that consequential responsibility following a technical malfunction rests with the person

who chose to deploy the malfunctioning device. This answerability could be mitigated by providing recorded evidence about the soundness of the choice process.

Boundaries and boundary objects: Problems with systems are particularly likely to arise if the systems are intended to cut across organisational or professional boundaries. One reason why such problems arise is that the responsibilities on each side of the boundary are either undefined or are incompatible.

One design implication is that need for explicit representation of the nature of relationships across the boundary, identifying boundary objects, conversations and communication, and shared and non-shared responsibilities. Because boundary objects have differing interpretations on the two sides of the boundary, there is often a need for two distinct representations of the object. For example, to the train operator, a train is a unit of schedulable resource, which requires other schedulable resources, such as a trained driver. Essentially the need is for a static representation. But for the infrastructure provider, a train is a dynamic unity defined by a head code and a moving point on the track; the need is for a dynamic representation. Tying together the two different representations is not, to be sure, an insuperable problem, but it does present a certain complexity in system design.

Monitoring: Not everything need be monitored. Obviously if failure is likely to be catastrophic, fault tolerance and recoverability measures are important. But if the consequences of failure are likely to be merely inefficiencies, resources for planning for and implementing monitoring are best spent where the structures of the system or its associated responsibilities cross organisational boundaries, since it is there that disputes are both more likely to arise and difficult and costly to resolve.

Audit trail: It is unusual for information systems to have the capability to record things that happen in the social domain, such as delegation. It is in the interest of agents who hold consequential responsibility that the audit trail is correct, reconstrucable and complete. For example, one possible organisational rule is that consequential responsibility following a technical malfunction rests with the agent who chose to deploy the malfunctioning device. This answerability could perhaps be mitigated by providing evidence about the soundness of the choice process, including those aspects of it that took place in he social domain.

Design for recoverability: There are two main classes of strategy for recoverability: backward recovery is the restoration of a previous state known to be safe, usually saved as a checkpoint or archive; forward recovery is the forcing into a known safe state. Backward recovery is not always possible for systems that are closely coupled to the world, since although the system can be rewound, it is not always possible to rewind the world.

One important strategy for recovery after a failure is diversity: trying not to put all you eggs in one basket is as important during recovery as it is during normal operation. Remember that independent systems of the same functionality may well not fail independently (e.g. having a second driver in the cab may not help if both have been on the same defective training course).

Summary and Conclusions

Focussing on responsibility is to make a distinction between who is responsible for performing a role and who (or what) actually executes a role. We advocate that looking at responsibilities is a better guide for designing information systems than looking at executions, since it allows analysis of problems that can potentially arise when responsibility has not been clearly allocated, those responsible do not or can not actually perform the role, responsibility cannot be enforced because of lack of corresponding authority, communication between jointly responsible actors is difficult or impossible, and many other causes of organisational failure. In this section we look at the acquisition of information about responsibilities. Clearly one way of finding out about responsibilities is direct enquiry: asking informants (and their managers), looking at their job descriptions and contracts and so on. But the direct approach, although necessary, is also limited. People's interpretation of their responsibilities are often nuanced, and this nuancing is often better determined as a result of observation and subsequent elaboration, since direct questions are usually answered directly.

It is one of the roles of ethnography to observe the manifestation of interpretation of responsibility. It can do this by explicating social aspects of work and considering the relationship between local practice and formal procedure – how procedures are enacted and how practice is related to or explained as or accounted for in terms of formal process. It can probe into aspects of safety culture as these are enacted in the organisation. Because ethnographic interpretation considers systems in a broad socio-technical sense, it is particularly useful for analyses of 'systems'

where computers are minor players. Ethnography is also useful in identifying boundary objects and the ways their interpretations differ on each side.

Ethnography can also be useful in failure mode analysis. One particular use is in situations where response to potential or actual failure is a preventative or recoverable action, ethnography provides a description of what actions actually occurred — as opposed to what actions were supposed to occur. (It is hardly necessary to stress how often these differ.) It can show how fault-error-failure chains are investigated and examine the nature of interactions across organisational boundaries – how processes are brought into alignment, and who or what does the job of translation. (This is particularly important for recoverability.)

So far, three possible uses for models of responsibility seem to be emerging:
1. During planning/procurement/requirements when there is a need to clarify the responsibilities of the different actors in the system, especially where multiple organisations are involved.
2. During an enquiry, when there is a need to find out who takes the blame and (perhaps) who should have done something.
3. During system operation, when a problem arises and there is a need to find out who needs to know and what they need to know.

Organisational complexity requires an ICT system design method which recognises that multi-organisational systems need to extend current methods in the way they deal with failure. Three examples seem particularly important.
1. Procurement processes which are based on the single organisation assumption may not work too well.
2. Failures which can be traced back to errors in the sharing of responsibility are going to occur and the recovery procedures also have to be designed for a multi-organisational context of use. Where consequential responsibility is unclear, the social and legal processes require more information than just that immediately prior to the triggering event. the nature of the contract between the parties may have implications for existing (or non-existing) systems.
3. Information is often best regarded as a partial state of a conversation and understanding the nature of the conversation is needed to construct the multiple contexts of generation and interpretation.

Acknowlegements
We wish to thank all those who have participated in discussions with us at Lancaster (particularly Mark Rouncefield Guy Dewsbury and Ian Sommerville) and Newcastle (particularly Mike Martin and Ros Strens). This work is currently supported by the EPSRC through the Interdisciplinary Research Collaboration in Dependability (DIRC).

References
Checkland, P. (1981). *Systems Thinking, Systems Practice*, Chichester, John Wiley.
Checkland, P. and J. Scholes (1990). *Soft Systems Methodology in Action*, Chichester, John Wiley.
Clarke, K.M. and Hardstone, G., *Trust in Technology*, Kluwer, 2005 (in press).
Jonas, H. (1984) *The Imperative of Responsibility*, Chicago, University of Chicago Press
Lucas, J. R. (1995). *Responsibility*, Oxford, Clarendon Press.

Capturing Emerging Complex Interactions - Safety Analysis in ATM

Massimo Felici

LFCS, School of Informatics, The University of Edinburgh, Edinburgh EH9 3JZ, UK
http://homepages.inf.ed.ac.uk/mfelici/

Abstract: The future development of Air Traffic Management (ATM), set by the ATM 2000+ Strategy, involves a structural revision of ATM processes, a new ATM concept and a systems approach for the ATM network. This requires ATM services to go through significant structural, operational and cultural changes that will contribute towards the ATM 2000+ Strategy. Moreover, from a technology viewpoint, future ATM services will employ new systems forming the emergent ATM architecture underlying and supporting the European Commission's Single European Sky Initiative. Introducing safety relevant systems in ATM contexts requires us to understand the risk involved in order to mitigate the impact of possible failures. This paper is concerned with some limitations of safety analyses with respect to operational aspects of introducing new systems (functionalities).

Keywords: Safety Analysis, ATM, Complex Interactions, System Evolution.

Introduction

The future development of Air Traffic Management (ATM), set by the ATM 2000+ Strategy (EUROCONTROL, 2003), involves a structural revision of ATM processes, a new ATM concept and a systems approach for the ATM network. The overall objective (EUROCONTROL, 2003) is, *for all phases of flight, to enable the safe, economic, expeditious and orderly flow of traffic through the provision of ATM services, which are adaptable and scalable to the requirements of all users and areas of European airspace*. This requires ATM services to go through significant structural, operational and cultural changes that will contribute towards the ATM 2000+ Strategy. Moreover, from a technology viewpoint, future ATM services will employ new systems forming the emergent ATM architecture underlying and supporting the European Commission's Single European Sky Initiative. ATM services, it is foreseen, will need to accommodate an increasing traffic, as many as twice number of flights, by 2020. This challenging target will require the cost-effectively gaining of extra capacity together with the increase of safety levels (Matthews, 2002; Overall, 1995). Enhancing safety levels affects the ability to accommodate increased traffic demand as well as the operational efficiency of ensuring safe separation between aircrafts. Suitable safe conditions shall precede the achievement of increased capacity (in terms of accommodated flights). Therefore, it is necessary to foreseen and mitigate safety issues in aviation where ATM can potentiality deliver safety improvements.

Introducing safety relevant systems in ATM contexts requires us to understand the risk involved in order to mitigate the impact of possible failures. Safety analysis involves the activities (i.e., definition and identification of system(s) under analysis, risk analysis in terms of tolerable severity and frequency, definition of mitigation actions) that allow the systematic identification of hazards, risk assessment and mitigation processes in critical systems (Leveson, 2005; Storey, 1996). Diverse domains (e.g., nuclear, chemical or transportation) adopt safety analyses that originate from a general approach (Leveson, 2005; Storey, 1996). Recent safety requirements, defined by EUROCONTROL (European organization for the safety of air navigation), imply the adoption of a similar safety analysis for the introduction of new systems and their related procedures in the ATM domain (EUROCONTROL, 2001a). Unfortunately, ATM systems and procedures have distinct characteristics (e.g., openness, volatility, etc.) that expose limitations of the approach. In particular, the complete identification of the system under analysis is crucial for its influence on the cost and the effectiveness of the safety analysis. Some safety-critical domains (e.g., nuclear and chemical plants) allow the properly application of conventional safety analyses. Physical design structures constrain system's interactions and stress the separation of safety related components from other system parts. This ensures the independence of failures. In contrast, ATM systems operate in open and dynamic environments where it is difficult completely to identify system interactions. For instance, there exist complex interactions between aircraft systems and ATM safety relevant systems. Unfortunately, these complex interactions may give rise to catastrophic failures. The accident (1 July 2002) between a BOING B757-200 and a Tupolev TU154M (BFU, 2004), that caused the fatal injuries of 71 persons, provides an instance of unforeseen complex interactions. These interactions triggered a catastrophic failure, although all aircraft systems were functioning properly. Hence, safety analysis has to take into account these complex interaction mechanisms (e.g., failure

dependence, reliance in ATM, etc.) in order to guarantee and even increase the overall ATM safety as envisaged by the ATM 2000+ Strategy.

This paper is concerned with some limitations of safety analyses with respect to operational aspects of introducing a new system (functionality). The paper is structured as follows. Firstly, it introduces safety analysis in ATM domain. The EUROCONTROL Safety Regulatory Requirement (EUROCONTROL, 2001a), ESARR4, requires the use of a risk based-approach in ATM when introducing and/or planning changes to any (ground as well as onboard) part of the ATM System. Unfortunately, ATM systems, procedures and interactions expose limitations of safety analyses. This paper proposes a framework for capturing complex interactions. The framework supports the iterative aspects of safety analyses. It, finally, discusses the proposed framework and draws some conclusions.

Safety Analysis in ATM

ATM services across Europe are constantly changing in order to fulfill the requirements identified by the ATM 2000+ Strategy (EUROCONTROL, 2003). Currently, ATM services are going through a structural revision of processes, systems and underlying ATM concepts. This highlights a systems approach for the ATM network. The delivery and deployment of new systems will let a new ATM architecture to emerge. The EUROCONTROL OATA project (Skyway, 2004) intends to deliver the Concepts of Operation, the Logical Architecture in the form of a description of the interoperable system modules, and the Architecture Evolution Plan. All this will form the basis for common European regulations as part of the Single European Sky.

The increasing integration, automation and complexity of the ATM System requires a systematic and structured approach to risk assessment and mitigation, including hazard identification, as well as the use of predictive and monitoring techniques to assist in these processes. Faults (Laprie et al, 1998) in the design, operation or maintenance of the ATM System or errors in the ATM System could affect the safety margins (e.g., loss of separation) and result in, or contribute to, an increased hazard to aircrafts or a failure (e.g., a loss of separation and an accident in the worst case). Increasingly, the ATM System relies on the reliance (e.g., the ability to recover from failures and accommodate errors) and safety (e.g., the ability to guarantee failure independence) features placed upon all system parts. Moreover, the increased interaction of ATM across State boundaries requires that a consistent and more structured approach be taken to the risk assessment and mitigation of all ATM System elements throughout the ECAC (European Civil Aviation Conference) States (EUROCONTROL, 2001). Although the average trends show a decrease in the number of fatal accidents for Europe, the approach and landing accidents are still the most safety pressing problems facing the aviation industry (Ranter, 2003; Ranter, 2004; van Es, 2001). Many relevant repositories[4] report critical incidents involving the ATM System. Unfortunately, even maintaining the same safety levels across the European airspace would be insufficient to accommodate an increasing traffic without affecting the overall safety of the ATM System (Enders, Dodd, and Fickeisen, 1999).

The introduction of new safety relevant systems in ATM contexts requires us to understand the risk involved in order to mitigate the impact of possible failures. The EUROCONTROL Safety Regulatory Requirement (EUROCONTROL, 2001a), ESARR4, requires the use of a risk based-approach in ATM when introducing and/or planning changes to any (ground as well as onboard) part of the ATM System. This concerns the human, procedural and equipment (i.e., hardware or software) elements of the ATM System as well as its environment of operations at any stage of the life cycle of the ATM System. The ESARR4 (EUROCONTROL, 2001a) requires that ATM service providers systematically identify any hazard for any change into the ATM System (parts). Moreover, they have to assess any related risk and identify relevant mitigation actions. In order to provide guidelines for and standardize safety analysis EUROCONTROL has developed the EATMP Safety Assessment Methodology (SAM) (EUROCONTROL, 2004) reflecting best practices for safety assessment of Air Navigation Systems.

The SAM methodology provides a means of compliance to ESARR4. The SAM methodology describes a generic process for the safety assessment of Air Navigation Systems. The objective of the methodology is to define the means for providing assurance that an Air Navigation System is safe for operational use. The methodology

[4] Some repositories are: Aviation Safety Reporting Systems - http://asrs.arc.nasa.gov/ -; Aviation Safety Network - http://aviation-safety.net/ -; Flight Safety Foundation: An International Organization for Everyone Concerned With Safety of Flight - http://www.flightsafety.org/ -; Computer-Related Incidents with Commercial Aircraft: A Compendium of Resources, Reports, Research, Discussion and Commentary compiled by Peter B. Ladkin et al. - http://www.rvs.uni-bielefeld.de/publications/Incidents/ -.

describes a generic process for the safety assessment of Air Navigation Systems. The process consists of three major steps: *Functional Hazard Assessment (FHA)*, *Preliminary System Safety Assessment (PSSA)* and *System Safety Assessment (SSA)*. Figure 1 shows how the SAM methodology contributes towards system assurance. The process covers the complete life cycle of an Air Navigation System, from initial system definition, through design, implementation, integration, transfer to operations, to operations and maintenance. Moreover, it takes into account three different types of system elements (human, procedure and equipment elements), the interactions between these elements and the interactions between the system and its environment.

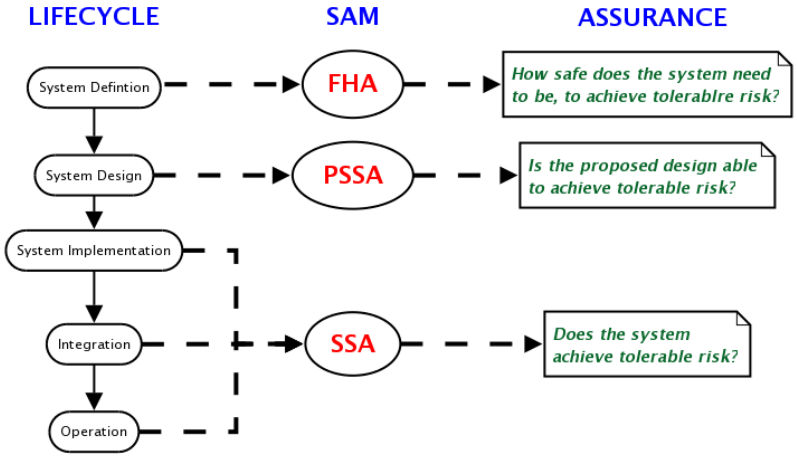


Figure 1 - Contribution of the Safety Assessment Methodology towards system assurance

The FHA is a top-down iterative process, initiated at the beginning of the development or modification of an Air Navigation System. The objective of the FHA process is to determine the overall safety requirements of the system (i.e., specifies the safety level to be achieved by the system). The process points out potential functional failures modes and hazards. It assesses the consequences of their occurrences on the safety of operations, including aircraft operations, within a specified operational environment. The FHA process specifies overall Safety Objectives of the system. The PSSA is another top-down iterative process, initiated at the beginning of a new design or modification to an existing design of an Air Navigation System. The objective of performing a PSSA is to demonstrate whether the assessed system architecture can reasonably be expected to achieve the Safety Objectives specified in the FHA. The PSSA process the Safety Objectives into Safety Requirements allocated to the system elements. That is, it identifies the risk level to be achieved by each system element. The SSA is a process initiated at the beginning of the implementation of an Air Navigation System. The objective of performing a SSA is to demonstrate that the implemented system achieves an acceptable (or at least tolerable) risk and consequently satisfies its Safety Objectives specified in the FHA. Moreover, the SSA assesses whether each system element meets its Safety Requirements specified in the PSSA. The SSA process collects evidences and provides assurance throughout the system life cycle (i.e., from implementation to decommissioning).

Although the SAM methodology describes the underlying principles of the safety assessment process, it provides limited information to applying these principles in specific projects. The hazard identification, risk assessment and mitigation processes comprise a determination of the scope, boundaries and interfaces of the constituent part being considered, as well as the identification of the functions that the constituent part is to perform and the environment of operations in which it is intended to operate. This supports the identification and validation of safety requirements on the constituent parts.

Modeling: The definition and identification of the system under analysis is extremely critical in the ATM domain and can have a significant influence on the safety analysis. System Models used during design phases provide limited support to safety as well as risk analysis. This is because existing models defined in the design phases are adapted and reused for safety and risk analysis. Organizational and cost-related reasons often determine this choice, without questioning whether models are suitable for the intended use. The main drawback is that design models are tailored to support the work of system designers. Thus, system models capture characteristics that may be of

primary importance for design, but irrelevant for safety analysis. On the contrary, models should be built as working-tools that, depending on their intended use, ease and support specific cognitive operations of users, for instance, by highlighting some aspects and neglecting others. The level of granularity of the model should be adaptable to the safety relevance of the part under analysis. Modeling has attracted a substantial effort from research and practice in system engineering. In spite of quality and effective development processes, many system faults are traced back to high-level requirements. This has motivated the increasing use of modeling in system engineering. The aim of modeling is twofold. On the one hand modeling contributes towards correctness and completeness of system requirements. On the other hand modeling supports validation and verification activities. The overall goal of modeling is mainly to reduce the gap between system requirements and design. The requirements-design gap represents a major source of (requirements) changes. Although this gap is one of the sources of requirements changes, research on (requirements) evolution clearly points out other origins of changes (PROTEUS, 1996). Modeling tackles two main issues. The first is that translations from requirements to design are error-prone. The second is that stakeholders (e.g., system users, system engineers, etc.) have often contradicting understandings about which system. These problems have motivated the blossom of many modeling methodologies and languages, e.g., UML (Rumbaugh, Jacobson, and Booch, 1999), used in practice.

Modeling incorporates design concepts and formalities into system specifications. This enhances our ability to assess safety requirements. For instance, *Software Cost Reduction* (SCR) consists of a set of techniques for designing software systems (Heitmeyer, 2002; Hoffman and Weiss, 2001). The SCR techniques support the construction and evaluation of requirements. The SCR techniques use formal design techniques, like tabular notation and information hiding, in order to specify and verify requirements. According to information hiding principles, separate system modules have to implement those system features that are likely to change. Although module decomposition reduces the cost of system development and maintenance, it provides limited support for system evolution. *Intent Specifications* provide another example of modeling that further supports the analysis and design of evolving systems (Leveson, 2000). Intent Specifications extend over three dimensions. The vertical dimension represents the intent and consists of five hierarchical levels[5]. Along the horizontal dimension, the Intent Specifications decompose the whole system in heterogeneous parts: Environment, Operator, System and Components. The third dimension, Refinement, further breaks down both the Intent and Decomposition dimensions into details. Each level provides rationale (i.e., the intent or "why") about the level below. Each level has mappings that relate the appropriate parts to the levels above and below it. These mappings provide traceability of high-level system requirements and constraints down to physical representation level (or code) and vice versa. In general, the mappings between Intent levels are many-to-many relationships. In accordance with the notion of semantic coupling, Intent Specifications support strategies (e.g., eliminating tightly coupled, many-to-many, mappings or minimizing loosely coupled, one-to-many, mappings) to reduce the cascade effect of changes. Although these strategies support the analysis and design of evolving systems, they provide limited support to understand the evolution of high-level system requirements[6]. The better is our understanding of system evolution; the more effective are design strategies. That is, understanding system evolution enhances our ability to inform and drive design strategies. Hence, evolution-informed strategies enhance our ability to design evolving systems.

Modeling methodologies and languages advocate different design strategies. Although these strategies support different aspects of software development, they originate in a common *Systems Approach*[7] to solving complex

[5] Level 1, system purpose; Level 2, system principles; Level 3, blackbox behavior; Level 4, design representation; Level 5, physical representation or code. Note that a recent version of Intent Specifications introduces two additional levels: Level 0 and Level 6. Level 0, the management level, provides a bridge from the contractual obligations and the management planning needs to the high-level engineering design plans. Level 6, the system operations level, includes information produced during the actual operation of the system.

[6] Leveson in (Leveson, 2000) reports the problem caused by Reversals in TCAS (Traffic Alert and Collision Avoidance System): *"About four years later the original TCAS specification was written, experts discovered that it did not adequately cover requirements involving the case where the pilot of an intruder aircraft does not follow his or her TCAS advisory and thus TCAS must change the advisory to its own pilot. This change in basic requirements caused extensive changes in the TCAS design, some of which introduced additional subtle problems and errors that took years to discover and rectify."*

[7] *"Practitioners and proponents embrace a holistic vision. They focus on the interconnections among subsystems and components, taking special note of the interfaces among various parts. What is significant is that system builders include heterogeneous components, such as mechanical, electrical, and organizational parts, in a single system. Organizational parts might be managerial structures, such as a military command, or political entities, such as a government bureau. Organizational components not only interact with technical ones but often reflect their characteristics. For instance, a management organization for presiding over the development of an intercontinental missile system might*

problems and managing complex systems. In spite of common grounds, modeling methodologies and languages usually differ in the way they interpret the relationships among heterogeneous system parts (e.g., hardware components, software components, organizational components, etc.). A common aspect is that models identify the relations between the different system parts. On the one hand these relations constrain the system behavior (e.g., by defining environmental dependencies). System (architectural) design partially captures these relations. On the other hand they are very important for system management and design. Among the different relations over heterogeneous system parts and hierarchical levels is *Traceability*. Although traceability supports management, traceability often faces many issues in practice. In particular, traceability faces evolution.

Research and practice in system engineering highlight critical issues. Among these issues evolution affects many aspects of the system life cycle. Unfortunately, most methodologies provide limited support to capture and understand system evolution. This is often because the underlying hypotheses are often unable to capture system evolution. Although requirements serve as basis for system production, development activities (e.g., system design, testing, safety analysis, deployment, etc.) and system usage feed back system requirements. Thus system production as a whole consists of cycles of discoveries and exploitations. The different development processes (e.g., V model, Spiral model, etc.) diversely capture these discover-exploitation cycles, although development processes constrain any exploratory approach that investigates system evolution. Thus system-engineering methodologies mainly support strategies that consider changes from a management viewpoint. In contrast, system changes, like the ones occurring in the ATM System, are emerging behaviors of combinations of development processes, products and organizational aspects.

Limitations: Conventional safety analyses are deemed acceptable in domains such as the nuclear or the chemical sector. Nuclear or chemical plants are well-confined entities with limited predictable interactions with the surroundings. In nuclear and chemical plants design stresses the separation of safety related components from other plant systems. This ensures the independence of failures. Therefore, in these application domains it is possible to identify acceptable tradeoffs between completeness and manageability during the definition and identification of the system under analysis. In contrast, ATM systems operate in open and dynamic environments. Hence, it is difficult to identify the full picture of system interactions in ATM contexts. In particular:

- There is a complex interaction between aircrafts' controls and ATM safety functions. Unfortunately, this complex interaction may give rise to catastrophic failures. Hence, failure separation (i.e., understanding the mechanisms to enhance failure independence) would increase the overall ATM safety.
- Humans (Flight Safety Foundation, 2003; Pasquini and Pozzi, 2004) using complex language and procedures mediate this interaction. Moreover, most of the final decisions are still demanded to humans whose behavior is less predictable than that of automated systems. It is necessary further to understand how humans use external artifacts (e.g., tools) to mediate this interaction. Moreover, this will allow the understanding of how humans adopt technological artifacts and adapt their behaviors in order to accommodate ATM technological evolution. Unfortunately, the evolution of technological systems often corresponds to a decrease in technology trust affecting work practice.
- Work practice and systems evolve rapidly in response to demand and a culture of continuous improvements. A comprehensive account of ATM systems, moreover, will allow the modeling of the mechanisms of evolution. This will enhance strategies for deploying new system configurations or major system upgrades. On the one hand modeling and understanding system evolution support the engineering of (evolving) ATM systems. On the other hand modeling and understating system evolution allow the communication of changes across different organizational levels. This would enhance visibility of system evolution as well as trust in transition to operations.

Capturing Emerging Complex Interactions

Heterogeneous engineering[8] provides a different perspective that further explains the complex interaction between system (specification) and environment. Heterogeneous engineering considers system production as a whole. It provides a comprehensive account that stresses a holistic viewpoint, which allows us to understand the underlying

---

be divided into divisions that mirror the parts of the missile being designed.", INTRODUCTION, p. 3, (Hughes and Hughes, 2000).

[8] *"People had to be engineered, too - persuaded to suspend their doubts, induced to provide resources, trained and motivated to play their parts in a production process unprecedented in its demands. Successfully inventing the technology, turned out to be heterogeneous engineering, the engineering of the social as well as the physical world."*, p. 28, (MacKenzie, 1990).

mechanisms of evolution of socio-technical systems. Heterogeneous engineering involves both the systems approach (Hughes and Hughes, 2000) as well as the social shaping of technology (MacKenzie and Wajcman, 1999). On the one hand system engineering devises systems in terms of components and structures. On the other hand engineering processes involve social interactions that shape socio-technical systems. Hence, stakeholder interactions shape socio-technical systems. Heterogeneous engineering is therefore convenient further to understand engineering processes.

The most common understanding in system engineering considers requirements as goals to be discovered and design solutions as separate technical elements. Hence system engineering is reduced to be an activity where technical solutions are documented for given goals or problems. Differently according to heterogeneous engineering, system requirements specify mappings between problem and solution spaces. Both spaces are socially constructed and negotiated through sequences of mappings between solution spaces and problem spaces (Bergman, King, and Lyytinen, 2002; 2002a). Therefore, system requirements emerge as a set of consecutive solution spaces justified by a problem space of concerns to stakeholders. Requirements, as mappings between socio-technical solutions and problems, represent an account of the history of socio-technical issues arising and being solved within industrial settings.

The formal extension of these mappings (or solution space transformations) identifies a framework to model and capture evolutionary system features (e.g., requirements evolution, evolutionary dependencies, etc.) (Felici, 2004). The resulting framework is sufficient to interpret system changes. Therefore, the formal framework captures how design solutions evolve through subsequent releases. Hence, it is possible to define system evolution in terms of sequential solution space transformations. Moreover, it is possible to capture evolution at different abstraction levels with diverse models. This defines evolutionary cycles of iterations in the form: solutions, problems and solutions. This implies that engineering processes consist of solutions searching for problems, rather than the other way around (that is, problems searching for solutions). This holistic viewpoint of systems allows us to understand the underlying mechanisms of evolution of socio-technical systems, like the ATM System.

Capturing cycles of discoveries and exploitations during system design involves the identification of mappings between socio-technical solutions and problems. The proposed framework exploits these mappings in order to construct an evolutionary model that will inform safety analyses of ATM systems. Figure 2 shows the proposed framework, which captures these evolutionary cycles at different levels of abstraction and on diverse models. The framework consists of three different hierarchical layers: *System Modeling Transformation (SMT)*, *Safety Analysis Modeling Transformation (SAMT)* and *Operational Modeling Transformation (OMT)*.

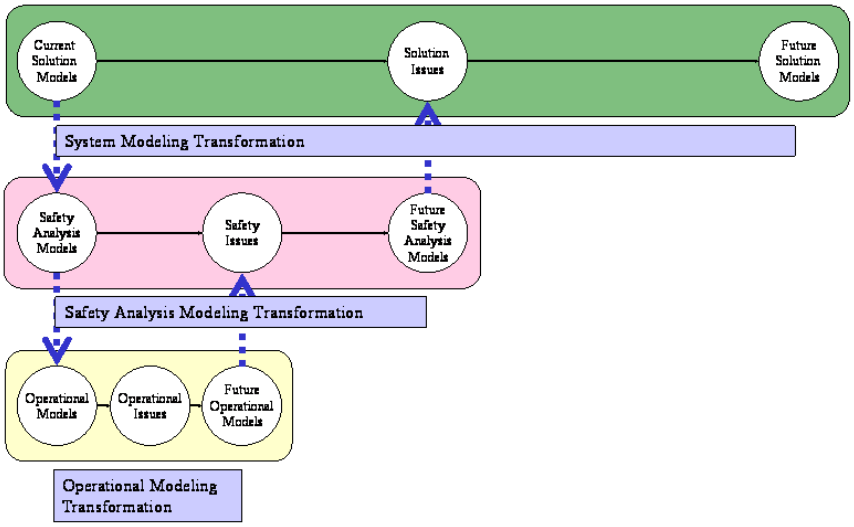


Figure 2 - A framework for modeling evolutionary safety analyses

The SMT layer captures how solution models evolve in order to accommodate design issues or evolving requirements. Therefore, an SMT captures system requirements as mappings between socio-technical solutions and

problems. This allows the gathering of changes into design solutions. That is, it is possible to identify how changes affect design solution. Moreover, this enables sensitivity analyses of design changes. In particular, this allows the revision of safety requirements and the identification of hazards due to the introduction of a new system. Therefore, the SMT supports the gathering of safety requirements for evolving systems. That is, it supports the main activities occurring during the top-down iterative process FHA in the SAM methodology (EUROCONTROL, 2004). The FHA in the SAM methodology then initiates another top-down iterative approach, i.e., the PSSA. Similarly, the framework considers design solutions and safety objectives as input to Safety Analyses. Safety analyses assess whether the proposed design solution satisfies the identified safety objectives. This phase involves different methodologies (e.g., Fault Tree Analysis, HAZOP, etc.) that produce diverse (system) models. System usage or operational trials may give rise to unforeseen safety issues that invalidate (part of) safety models. In order to take into account these issues, it is necessary to modify safety analyses. Therefore, safety analysis models evolve too. SAMT, the second layer of the framework, captures how safety analysis models evolve in order to accommodate raising safety issues. Although design models serve as a basis for safety models, they provide limited supports to capture unforeseen system interactions. Therefore, SAMT supports those activities involved in the PSSA process of the SAM methodology (EUROCONTROL, 2004). Note that although the SAM methodology stresses that both FHA and PSSA are iterative process, it provides little supports to manage process iterations as well as system evolution in terms of design solution and safety requirements. The framework supports these evolutionary processes.

Finally, operational models (e.g., structured scenarios, patterns of interactions, structured procedures, workflows, etc.) capture heterogeneous system dynamics. Unfortunately, operational profiles often change with system usage. For instance, system users often refine procedures in order to integrate different functionalities or to accommodate system failures. OMT, the bottom-layer of the framework, captures how operational models change in order to accommodate issues arising. The evolution of operation models informs safety analyses of new hazards. Therefore, OMT supports the activities involved in the SSA process of the SAM methodology.

Discussion and Conclusions

The proposed framework addresses three main points in order effectively to support evolutionary safety analyses. Firstly, the model questions the system boundaries and the required level of details. These aspects considerably vary from design models to risk analysis models, since system parts that need to be specified in details for the design may be much less relevant from a safety point of view. The typical drawback experienced in most cases is that resources for risk analysis may be consumed in investigating detailed aspects of every system part, instead of trying to identify unknown risks that may be related to elements not central in the design model. Furthermore it is often the case that system boundaries can be more neatly defined in respect to the design objectives, whilst risk analysis often requires the adoption of a larger focus. All the recent major incidents occurred in the civil aviation domain proved to stem from unexpected interactions from a large variety of elements, differently located in space and time. Those elements were often judged as outside of the system boundaries (or outside of normal operating conditions) when safety analysis has been conducted. For instance, the investigation report (BFU, 2004) of the accident between two aircrafts highlights that although individual ATM systems and procedures work properly, the ATM socio-technical interactions may, unfortunately, result in a catastrophic event.

The second point directly addresses these unexpected interactions between system elements as main source of incidents. Best practices and standards in safety analysis prescribe that mutual impact between different risks be analyzed. A system model is a key support to perform this task effectively, but the possible interactions need to be represented explicitly. On the contrary, models defined for design purposes usually outline the relationship between system elements by a functional (or physical) decomposition. In all the cases when design models are exploited for the safety analysis, the functional decomposition principle many unduly provide the structure for the analysis of incident causal dynamics (Johnson, 2003; Leveson, 2004), thus failing to acknowledge their different underlying nature. Furthermore, a correct model should not only ensure that interactions and mutual impact between different risks be analyzed, but also outline interactions between everyday productive processes in “normal operating conditions”, since risk factors are likely to interact along these lines.

The third characteristic of the model refers to the possibility of effective re-use of (part of) the model to inform other safety analyses. This would ensure that part of the safety feedback and experience related to a system can be beneficial when introducing major changes to the current system or when developing new similar systems. In the same way, the effective reuse of the model would result in safety analyses that have better means to achieve a good balance between exhaustiveness and costs, as findings of closely related analysis could be easily considered.

In order realistically and cost-effectively to realize the ATM 2000+ Strategy, systems from different suppliers will be interconnected to form a complete functional and operational environment, covering ground segments and aerospace. Industry will be involved as early as possible in the life cycle of ATM projects. EUROCONTROL manages the processes that involve the definition and validation of new ATM solutions using Industry capabilities (e.g., SMEs). In practice, safety analyses adapt and reuse system design models (produced by third parties). Technical, organizational and cost-related reasons often determine this choice, although design models are unfit for safety analysis. Design models provide limited support to safety analysis, because they are tailored for system designers. The definition of an adequate model and of an underlying methodology for its construction will be highly beneficial for whom is performing safety analyses. As stated before, currently the model definition phase cannot be properly addressed as an integral part of safety analysis, mostly because of limited costs and resources. This paper is concerned with problems in modeling ATM systems for safety analysis. The main objective is to highlight a model specifically targeted to support safety analysis of ATM systems. Moreover, the systematic production of safety analysis (models) will decrease the cost of conducting safety analyses by supporting reuse in future ATM projects.

Acknowledgments: I would like to thank Alberto Pasquini and Simone Pozzi for their information about the ATM domain. This work has been supported by the UK EPSRC Interdisciplinary Research Collaboration in Dependability, DIRC - http://www.dirc.org.uk - grant GR/N13999.

References

Bergman, M., King, J.L., and Lyytinen, K. (2002). Large-scale requirements analysis as heterogeneous engineering. Social Thinking - Software Practice, pages 357-386.

Bergman, M., King, J.L., and Lyytinen, K. (2002a). Large-scale requirements analysis revisited: The need for understanding the political ecology of requirements engineering. Requirements Engineering, 7(3):152-171.

BFU (2004). Investigation Report, AX001-1-2/02.

Enders, J.H., Dodd, R.S., and Fickeisen, F. (1999). Continuing airworthiness risk evaluation (CARE): An exploratory study. Flight Safety Digest, 18(9-10):1-51.

EUROCONTROL (2001). EUROCONTROL Airspace Strategy for the ECAC States, ASM.ET1.ST03.4000-EAS-01-00, 1.0 edition.

EUROCONTROL (2001a). EUROCONTROL Safety Regulatory Requirements (ESARR). ESARR 4 - Risk Assessment and Mitigation in ATM, 1.0 edition.

EUROCONTROL (2003). EUROCONTROL Air Traffic Management Strategy for the years 2000+.

EUROCONTROL (2004). EUROCONTROL Air Navigation System Safety Assessment Methodology, 2.0 edition.

Felici, M. (2004). Observational Models of Requirements Evolution. PhD thesis, Laboratory for Foundations of Computer Science, School of Informatics, The University of Edinburgh.

Flight Safety Foundation (2003). The Human Factors Implication for Flight Safety of Recent Developments in the Airline Industry, (22)3-4 in Flight Safety Digest.

Heitmeyer, C.L. (2002). Software cost reduction. In John J. Marciniak, editor, Encyclopedia of Software Engineering. John Waley & Sons, 2nd edition.

Hoffman, D.M. and Weiss, D.M., editors (2001). Software Fundamentals: Collected Papers by David L. Parnas. Addison-Wesley.

Hughes, A.C. and Hughes, T.P., editors (2000). Systems, Experts, and Computers: The Systems Approach in Management and Engineering, World War II and After. The MIT Press.

Johnson, C.W. (2003). Failure in Safety-Critical Systems: A Handbook of Accident and Incident Reporting. University of Glasgow Press, Glasgow, Scotland.

Laprie, J.-C. et al (1998). Dependability handbook. Technical Report LAAS Report no 98-346, LIS LAAS-CNRS.

Leveson, N. (2004). A new accident model for engineering safer systems. Safety Science, 42(4):237-270.

Leveson, N.G. (2000). Intent specifications: An approach to building human-centered specifications. IEEE Transactions on Software Engineering, 26(1):15-35.

Leveson, N.G. (2005). SAFEWARE: System Safety and Computers. Addison-Wesley.

MacKenzie, D.A. (1990). Inventing Accuracy: A Historical Sociology of Nuclear Missile Guidance. The MIT Press.

MacKenzie, D.A. and Wajcman, J., editors (1999). The Social Shaping of Technology. Open University Press, 2nd edition.

Matthews, S. (2002). Future developments and challenges in aviation safety. Flight Safety Digest, 21(11):1-12.

Overall, M. (1995). New pressures on aviation safety challenge safety management systems. Flight Safety Digest, 14(3):1-6.

Pasquini, A. and Pozzi, S. (2004). Evaluation of air traffic management procedures - safety assessment in an experimental environment. Reliability Engineering & System Safety, 2004.

PROTEUS (1996). Meeting the challenge of changing requirements. Deliverable 1.3, Centre for Software Reliability, University of Newcastle upon Tyne.

Ranter, H. (2003). Airliner accident statistics 2002: Statistical summary of fatal multi-engine airliner accidents in 2002. Technical report, Aviation Safety Network.

Ranter, H. (2004). Airliner accident statistics 2003: Statistical summary of fatal multi-engine airliner accidents in 2003. Technical report, Aviation Safety Network.

Skyway (2004). Working towards a fully interoperable system: The EUROCONTROL overall ATM/CNS target architecture project (OATA). Skyway, 32:46-47.

Rumbaugh, J., Jacobson, I., and Booch, G. (1999). The Unified Modeling Language Reference Manual. Addison-Wesley.

Storey, N. (1996). Safety-Critical Computer Systems. Addison-Wesley.

van Es, G.W.H. (2001). A review of civil aviation accidents - air traffic management related accident: 1980-1999. Proceedings of the 4th International Air Traffic Management R&D Seminar, New-Mexico.

Extending Small Group Theory for Analysing Complex Systems

Alistair Sutcliffe

Centre for HCI Design, School of Informatics, University of Manchester
PO Box 88, Manchester M60 1QD, UK
a.g.sutcliffe@manchester.ac.uk

Abstract: This paper introduces a social psychological theory – Small Groups as Complex Systems – as a contribution to the design of CSCW and CMC systems. Small Group Theory is composed of local dynamics which model the internal view of a group; global dynamics that represent whole group emergent properties; and contextual dynamics that model the influences of the group's environment on its composition, coherence and behaviour. The potential contribution of Small Group Theory to the design of CSCW systems is investigated by model-based analysis of group members, supporting technology, and design principles motivated by the theory.

Keywords: cognitive theory, modelling framework, socio-technical systems

Introduction

Activity Theory (Nardi, 1996) and Distributed Cognition (Hutchins, 1995) have provided insight for informing the design of collaborative systems, but do not provide a detailed model of systems. In contrast, task modelling approaches have been extended for CSCW systems (Van der Veer, Lenting & Bergevoet, 1996), while Cognitive Work Analysis (Vicente, 1999) also provides a model-based approach for design of human activity that takes social and ecological context into account. However, there has been little convergence between task modelling and theory-driven approaches in CSCW. Instead, researchers in CSCW have evolved design principles from a combination of ethnographic study and design exploration (Abowd & Mynatt, 2000; Olson & Olson, 2000). However, such design principles focus on the technology for collaborative systems and tend to ignore the need for socio-technical solutions for collaborative systems.

One of the weaknesses of applying theories to HCI, e.g. Activity Theory and Distributed Cognition, is that design recommendations are indirect, i.e. they require a theory-knowledgeable expert to provide design recommendations. For example, the influence of Activity Theory analysis in design of the exemplar case study for a Coloured Petri Net tool (Bertlesen & Bødker, 2003), while plausibly explained, is not easy to generalise to other domains and examples. Perhaps, given the complexity of phenomena at the social level, it is unrealistic to expect theory to prescribe design in complex systems.

This paper introduces a social-psychological theory that does account for a wide range of phenomena and investigates its power for analysing requirements for the design of socio-technical systems. The theory of Small Groups as Complex Systems (hereafter SGACS theory: Arrow, McGrath & Berdahl, 2000), is a successor to Social Dependency Theory (McGrath, 1993). The paper is structured as follows: SGACS theory is introduced and briefly explained. The use of the theory as an analytic instrument is investigated and limitations of applying it are discussed. Principles are derived from SGACS theory that might be used to influence design of collaborative systems. The paper concludes with a brief discussion of the contributions that social-psychological theories might make to interactive systems design.

Small Groups as Complex Systems Theory

The SGACS theory comes from a social psychological heritage which takes an eclectic approach of theory synthesis building on research into groups as information processing systems (McGrath, 1998), bringing together 13 streams of social psychology research (Arrow et al., 2000). SGACS theory limits its scope to small groups, i.e. <=20 members. The theory contains a taxonomy of groups, a timeline view of group evolution, intra-group modelling called local dynamics, whole group modelling referred to as global dynamics, and assessment of the environment in contextual dynamics. The theory is based on a set of seven propositions that govern the influences at the local, global and context levels.

The theory classifies groups into task forces (single project, short duration); teams (many projects, longer duration); crews (strong role models for collaboration); and social groups (non-work goals, member-driven collaboration); see Figure 1. A set of dimensions classifies groups according to their duration, goal directedness, and mode of creation (external setup/internal motivation). SGACS theory explains how groups of different types develop over a life cycle of formation, emergence, operation, maturity, senescence; and describes qualities for

successful group interaction and pathologies in structure and behaviour that can disrupt achievement of common goals and destabilise the group.

	external organiser	no external organiser
Work related goals	teams, task forces, crews	ad hoc task forces
Socially motivated goals	social clubs, societies	social friendships, clans

Figure 1 - Taxonomy of Groups (adapted from Arrow et al., 2000)

The theory is composed of two layers, a bottom-up analysis driven from modelling the composition of groups, and an upper layer of emergent properties that characterise the group as a whole. Contextual dynamics describes the influence of the group's environment on both levels. The lower level, local dynamics, provides an internal view of the group composed of agents, goals, tasks, tools and communication channels. Key success factors include member participation, leadership, authority/hierarchy v. autonomy/democracy, etc. For task forces, formation of a sound task and labour network is important for effective sharing of work, while social relationships are less important given the short duration of the group. In contrast, a member and role network is vital for teams where social relationships are crucial to success. Crews require sound tools and job networks so they can function effectively, employing role knowledge even if individual members change. In SGACS theory, tools refers not only to hardware tools such as computer systems but also to tools as collective knowledge, i.e. shared strategies, procedures and norms which are important ingredients for teams and crews. The group-level, global dynamics view describes emergent properties of whole groups such as social cohesion, motivation, shared beliefs, image, goals, satisfaction of members, effectiveness in achieving tasks. Some concepts and heuristics describe how internal properties of groups might contribute to emergent properties at the global dynamics level. An overview of SGACS theory is given in Figure 2, which illustrates its components and models.

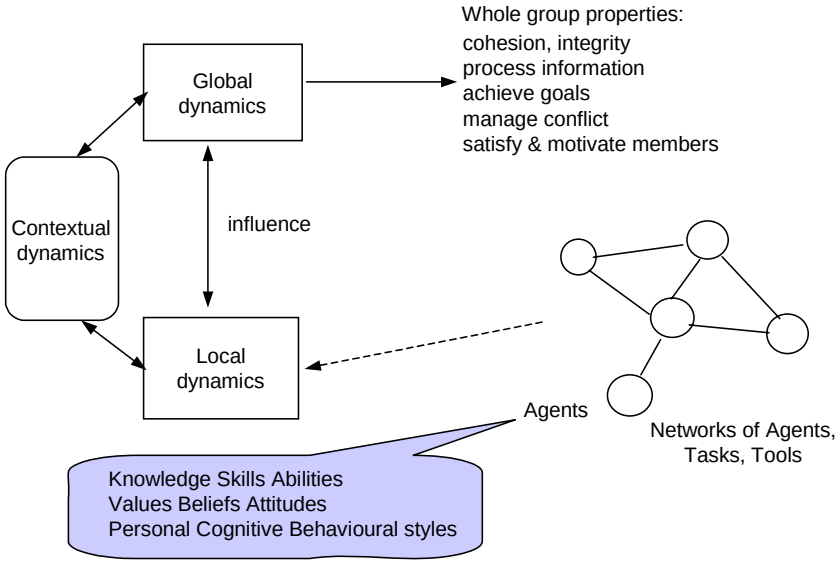


Figure 2 - Overview of the Components of Small Groups as Complex Systems Theory

Clearly, a few paragraphs of description can hardly do justice to a sophisticated and complex theory; however, further partial description is interleaved with a discussion of how to apply SGACS theory, and the reader is referred to Arrow et al. (2000) for an authoritative description.

Modelling Socio-Technical Systems

In this section we discuss how analytic techniques could extend SGACS theory to model socio-technical systems, specify requirements for supporting technology and diagnose potential problems in group working. Local dynamics models phenomena which HCI researchers are familiar with: tasks, agents and roles; however, the theory needs to be elaborated to demonstrate how modelling individuals in groups and their work can be used to predict group behaviour and performance.

Tasks, Agents and Roles: SGACS theory creates three complementary networks showing the relationships between agents and tasks, tasks and tools, and agents and tools. Tools may be software systems, information or other resources used for completing tasks. A key concern in task-agent-tool network analysis is the level of support provided by tools to the agent for achieving the task. Assessing task-agent-tool networks needs a measure of task-tool fit which could be taken from subjective ratings, combined with usability metrics from evaluation studies. In collaborative systems, even though the task-tool fit for individual roles might be reasonable, support for group integration may be more critical, so a separate group-level analysis of tool support will be necessary. Support for communication, coordination, and collaborative work could be assessed by expert judgement, questionnaires or by evaluation studies. Analysis techniques assess how well a task-goal could be developed using descriptions of supporting tools and agents' qualifications. The i* modelling language provides techniques for reasoning about relationships between task-goals, agents and supporting resources (Mylopoulos, Chung & Yu, 1999; Yu & Mylopoulos, 1994). Although i* has been augmented with socially oriented relationships, such as capability and commitment of agents and trust between them (Castro, Kolp & Mylopoulos, 2002), it does not explicitly consider interactions between agents or properties of whole groups. SGACS theory can supply such concepts via the local dynamics analysis. KSA (knowledge, skills and abilities) analysis could augment the concepts of capability and commitment to assess whether a group has the necessary human resources for a project. However, detailed assessment of how effectively tasks might be carried out by agents and supporting resources implies considerable domain knowledge. Using abstract categories of tasks, combined with appropriate knowledge of requirements and claims, might provide a viable approach (Sutcliffe 2000, 2002); otherwise, human expertise will be necessary for

interpreting models. The value of task-agent-tool modelling will depend on the insight gained in analysing pathologies in systems, compared with the expense of creating detailed models.

SGACS theory predicts that the task-agent-tool network should become more integrated over time and that a good fit between tasks, people and technology will enhance group effectiveness. Furthermore, individual goals and roles of group members should concord with the group's objectives. The social aspect of agent interaction could be analysed by studying the channels and patterns of communication between group members. SGACS theory does not deal with communication patterns; however, these could be studied empirically by discourse analysis (Clark, 1996) or simpler measures such as recording messages passed between agents to create network graphs of inter-agent communication frequencies. Pathologies may be posited if certain agents are isolated from communication networks. If a more detailed discourse analysis were carried out, other pathologies in communication patterns, e.g. arguments, disagreement, could be diagnosed. Unfortunately, discourse analysis is time consuming so a more economic approach might be using subjective ratings of communication frequency and effectiveness by each individual with other group members. In summary, task-tool support and communication analysis techniques could predict where problems might arise in cohesion of collaborating groups. In the following section we turn to analysis of individual group member attributes.

Knowledge, Skills and Abilities: One of the tenets of SGACS theory is that network integration at the affective level, i.e. trust, social familiarity and friendship, should deepen as the group matures in the formation and operation phase, leading to improved effectiveness. For this analysis, SGACS investigates the knowledge, skills and abilities (KSA) of agents to determine how well the group's human resources fit the needs of the tasks and group objectives. Then the values, beliefs and attitudes of individual members are evaluated, followed by personal, cognitive and behavioural styles. The second set of measures bears on group cohesion, since groups composed of members with very different attitudes and beliefs are less likely to develop the deep shared understanding that is necessary for effective collaboration.

SGACS theory does not specify how KSA analysis should be performed; however, knowledge can be interpreted as domain and task knowledge held by individuals that is relevant to the collective task. Skills may be interpreted in their cognitive sense, i.e. pre-compiled, internalised procedures for carrying out tasks. Abilities can be considered as capabilities or resources that contribute to the collective goal. Hence knowledge and skills are individual-level attributes ascribed to people, based on expert judgement or measures (e.g. skills tests), whereas abilities reflect capabilities of a person's role discerned by expert judgement.

In highly trained domains KSA analysis should show that all personnel have the necessary knowledge and capabilities to carry out their individual and collective tasks. KSA analysis should also show critical weak points in a collaborative system if training has been less than adequate. But deciding just when a deficit in KSA analysis might signal a dangerous flaw is not obvious. Individuals might collaborate to remedy deficits in knowledge and skills; however, ability problems should be easier to diagnose by comparing task requirements and agents' capabilities. KSA analysis, therefore, may complement task, tool and communication analysis for diagnosis of local dynamics problems.

Values, Beliefs and Attitudes: SGACS theory predicts that development of a network of personal relationships (member network) and a role network that connects people to shared group norms, resources and procedures, is important for establishing an emergent group-level culture and structure. VBA (values, beliefs and attitudes) analysis may indicate how cohesive a group might be with respect to its members' shared goals, culture and social norms. As with KSA analysis, SGACS theory does not specify how to conduct a VBA analysis. This analysis presents several difficulties. First, values are a nebulous concept usually not directly accessible in interviews, although questionnaires coupled with statistical cluster analysis can detect value-related concepts. Beliefs could be treated as knowledge and information that the core members held to be true about the domain, over the medium to long term. Attitudes can be viewed as a sub-set of this information where stronger valency prevails, so attitudes are construed to affect laden belief. This concurs with theories of emotion which distinguish emotional reactions and hence affective memory in reaction to agents, objects and events (Ortony, Clore & Collins, 1988). A further problem with attitude analysis is tacit knowledge (Rugg, McGeorge & Maiden, 2000). Individuals may articulate an "officially" held attitude at a meeting while holding a possibly contradictory attitude that they only voice in private (Goffman, 1976; Kahnemann & Tversky, 1982). Attitudes and beliefs could be captured by interviews or questionnaires, the latter being more reliable as they reduce the subjective interpretation of interview data. Unfortunately, development of questionnaire instruments takes time and resources to refine an appropriate set of questions from initial pilot studies. Hence capture of VBA data is likely to be time consuming. However, VBA analysis might be able to predict pathologies in group cohesion if mismatches between group members' attitudes

and beliefs were apparent. But this would require considerable further extension of the theory to predict which types of beliefs and attitudes might clash.

The next analysis, PCB (personal, cognitive and behaviour) styles, presents even more complexity. Such measures require personality-style inventories to be completed by the group members, e.g. for personality profiles (McCrae & John, 1992) or cognitive styles (Kelly, 1963). Personality testing is a reasonably mature area in psychology with many questionnaire instruments; however, how to diagnose pathological mixes of personal styles in groups is less certain. While PCB data could be collected via questionnaire inventories, it is less obvious how such data could be interpreted. There is some guidance in the personality styles literature about compatibilities between personal styles, but few firm guidelines exist to decide whether a certain mix of personality types would impair collaborative behaviours. Furthermore, interpreting the impact of cognitive styles on group cohesion is even less sure. Hence even though detailed descriptions of groups' members could be made in terms of traits and styles, predicting the impact on global dynamics may well be informed guesswork. It may be more useful to assess individual motivations and how these influence collective goals and group cohesion, but these aspects are not covered by SGACS theory.

Predicting Emergent Properties

SGACS theory describes emergent properties of whole groups as global dynamics and indicates that these should be a consequence of local dynamics; however, no procedures are given for establishing global group properties from lower-level local dynamics analysis. Ideally the theory should have predictive power to assess the potential success of groups given a detailed model of their participants. Desirable goals that groups should achieve are to fulfil members' needs, motivate members, process information, generate knowledge and achieve collective goals, while managing conflict, maintaining group integrity and developing agreement to complete group projects. The conjectured influences of local dynamics on global dynamics are illustrated in Table 1. It should be noted that these are hypotheses, not stated explicitly in SGACS. These hypotheses might be realised if effective analytic instruments could be developed and the posited effects were validated by experimental and empirical study. There is clearly considerable further research to realise these aims, but they do illustrate how detailed analysis might reveal emergence properties of groups.

Analysis treatment	Achieve group goals	Generate knowledge	Maintain integrity	Promote agreement	Motivate members
Task-agent role capabilities	+++	+++	++	++	+
Task-agent-tool support	+++	+++	+	+	+
Inter-agent communications	+++	++	++	+++	+
Task-agent KSA	+++	+++	++	+	+
Agent VBA	++	++	+++	+++	+++
Group goal-agent motivations	+++	+	+++	++	+++
Agent PCB	+	+	++	++	++

Table 1 - Implications of local dynamics analysis for group emergent properties (global dynamics) and analysing group performance

Some impact heuristics are posited by SGACS theory to link local and global dynamics. For instance, poor development of the task-agent-tool networks and lack of opportunity to develop close personal relationships indicates poor group cohesion and impaired information processing. Task-agent networks with inadequate knowledge and skills held by groups' members could indicate poor performance and increased errors, possibly leading to social tension within the group caused by inability to achieve collective goals.

VBA analysis might be able to predict potential conflict within groups, assuming that widely divergent beliefs and attitudes are a source of conflict. Conflict may be indicated by clashes between group members' attitudes, values and personality styles, if these are reported candidly in interviews or by questionnaires. Ethnographic or discourse analysis of conversation may reveal member attitudes; however, even if no open disagreements were observed, covert disagreements may be present. Agreement over the goals and achieving the group project is likely to be a function of how well formed the task-agent-tool network was, coupled with how well motivated group members are towards achieving a shared goal. Good motivation and sound task-agent tool support could counteract operational difficulties and tension between group members. A certain level of perturbation might be tolerated in

concocted groups since the external formation and authority of the founder might suppress dissent. If the members shared a common motivation then the chance of achieving group projects may be increased. External pressure may also motivate members of the group if there is pressure to conform to a shared view. Analysis of agents' roles, goals and responsibilities may be one approach to evaluate the influence of local dynamics on fulfilling members' needs and motivating members. However, SGACS theory does not make this association explicitly; furthermore, it does not distinguish between individual and organisational-level motivations. This suggests that a motivation analysis needs to be added to the theory.

This section has discussed the potential use of SGACS theory for analysis of socio-technical systems and demonstrates how it might pinpoint potential barriers to success. While SGACS theory places considerable emphasis on contextual dynamics and the influence of embedding context on group structure and behaviour, space precludes discussing these further, although some influences such as the role of external authority on group formation have been introduced.

Case Study Application

Space precludes an extensive description of our experience using SGACS theory, so, a high-level "lessons learned" summary of its application is reported in this section. The theory was applied to a case study in local government partnership between organisations in the London Borough of Havering. Data was collected from several interviews with council, police and other members of partnership task groups. Partnerships in the London Borough of Havering involved public and private sector organisations and were instigated in response to government policy set out in the Crime and Disorder Act of Parliament 1999, which aimed to create multi-agency cooperation to tackle crime. Eight task groups were targeted on different crime and community safety concerns. Each task group had members from Havering Police and the council (LBH). Council members were drawn from the appropriate departments, e.g. the Vulnerable Persons Group was attended by officials from the housing, education and social services departments. Other members of the task groups were drawn from the probation service, health authority, private sector organisations and charities.

Membership of the task groups was initiated by the two Community Safety Managers and creation of the groups was a tribute to their persuasive powers. Government organisations had a duty to become members of the task groups; however, several non-government organisations were also recruited as volunteers. Each organisation had to supply at least one individual as a group member. The task groups and community management team were supported by a GIS (geographic information system) owned by Havering Council. The police have an extensive crime reporting system and depersonalised data was transferred from the police system to the GIS, so that the distribution of crimes by type, and offender demographics (age, sex, background, etc.) could be investigated. The council and police maintained websites with limited information on partnerships and community safety. One task group on vehicle crime was analysed in depth. This group was given the mission of reducing vehicle crime in the borough with a particular emphasis on young offenders who had been responsible for most of the problems.

The Vehicle Crime Task Group was composed of four organisations: the borough police force, the local authority, schools and colleges in the area, and a charity which specialised in remedial education for young offenders who had been involved in car crime. The charity matched the cure to the crime by teaching young offenders car maintenance skills. The organisational structure is illustrated in Figure 3.

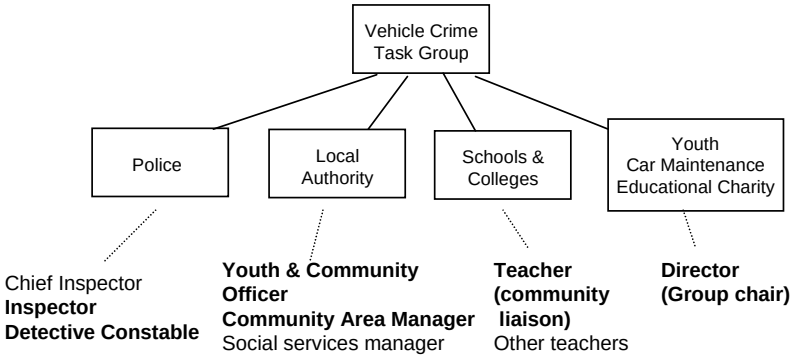


Figure 3 - Organisations belonging to the Vehicle Crime Task Group with the principal group members **in bold,** and other peripheral group members

The core individual members of the group who attended all meetings in the study period were two representatives from the local authority (youth and community officer, community area manager, a police inspector with responsibility for youth opportunities and a community sergeant from the police, one teacher who had community liaison responsibilities from the local colleges, and the chair who was the director of the educational-crime prevention charity. Other peripheral members of the group attended some meetings, including the chief inspector who was one of the group's founders. The task group was composed of four organisations but more were implicit members. For instance, the cooperation of the probation service was necessary to target persistent offenders of car crime, but this relationship was hindered by under-resourcing of the probation service, so they had not entered into a partnership agreement even though this was the intention in the Crime and Disorder Act. Claims and damage information from insurance companies was required for analysing crime patterns but it was not clear if this could be obtained.

According to SGACS theory, the Vehicle Crime Group was a concocted team. Members were brought together by external agency, initially by government policy and more directly by the initiative of the community management team who persuaded non-government members to volunteer their services and motivated government employees (the council, police, education and social services) to participate. The group's objective was to make recommendations and take action to reduce vehicle related crime in Havering. This group fits the SGACS theory's definition of a team since the group's mission can be decomposed into several sub-projects, and it had an anticipated lifespan of two to three years. Local dynamics analysis demonstrated poor coordination in the task-agent-tool network; furthermore the simple e-mail technology employed actually hindered communication due to problems with firewalls in each organisation prohibiting exchange of document attachments. The KSA analysis was limited by the subjects' time to be interviewed and fill in questionnaires to capture skills and abilities; however, a less formal analysis indicated that the group had a reasonable complement of skills and knowledge for the task in hand. We represented the KSA analysis as agents' capabilities in the i* requirements modelling frameworks (Mylopoulos, Chung & Yu, 1999; Yu & Mylopoulos, 1994) which enable simple type-level checking of human skills and abilities against the skills necessary to complete the tasks. The VBA (values, beliefs, attitude) analysis was more problematic to quantify. Consequently we restricted this to assessing individual group members' attitudes toward the collective group goal, rated on a 1-7 Likert scale. This demonstrated considerable differences between individuals in their commitment to the group goals. Global and contextual dynamics revealed that conflicts between organisational loyalties and lack of commitment to a shared goal had serious implications for the group's potential to succeed.

The case study illustrated use of SGACS theory for analysis of socio-technical systems and demonstrates how it can pinpoint potential barriers to success. The key findings from the case study which indicated an unsuccessful outcome of the Vehicle Crime team are summarised as follows:

1. Limited access to information from the police database because of security concerns over depersonalised data. Access was further inhibited by the under-resourced workload of preparing depersonalised data.
2. Poor technology support for collaborative work. This was caused by lack of a shared interactive GIS system, which restricted information access to a slow batch-request approach. The problem was compounded by restrictions on e-mail communication due to the police firewall.
3. The group had inadequate information resources about offender profiles, their behaviour and vehicle crime. This was partly a technology problem and partly a structural flaw in the group which did not have all the local schools or the probation service as members.
4. Infrequent meetings and poor social contact between members of the task group who worked in different locations. This hindered development of a richer task-agent-tool network.
5. Motivation analysis showed that some individual members and their organisations had poor motivation, which did not augur well for completing the group project.
6. Shared goals in the task group, while superficially in agreement, hid considerable disagreement about how to tackle the problem. This hindered information processing, and coordination in the task-agent network.
7. The VBA analysis indicated a considerable divergence in culture and norms between the participating organisations, which suggested that effective co-working may be hindered by conflicting mental models of the problem.
8. The local dynamics analysis showed that formation of an effective role network of members with social relationships and shared knowledge of procedures and norms was unlikely. This is one of the critical success factors for teams.

It was necessary to extend SGACS theory to explicitly model individuals' goals, their motivations, and attitudes towards the collective goal. Furthermore, SGACS theory assumes that groups have a clear membership boundary. We found that we needed to analyse conflicts between individuals' goals, their loyalty to their parent organisation, and attitude and motivation towards the task group. This three-way analysis of individual/group identity exposed many conflicts and resource problems which did not augur well for collective action. Our experience demonstrated that SGACS theory did provide a good conceptual framework for analysis of complex systems; however, it needed considerable extension to add measures and analysis techniques for the variables contained in its models.

Implications for Design

Most of the problems that may be uncovered by the SGACS analysis pertain to the social system; for instance, the poor construction of a team which did not have the appropriate resources or management to develop a well formed member network. We propose two major contributions which could be developed: first, by task-agent-tool modelling of local dynamics; and secondly, analysis of group composition using properties of individual members to expose potential pathologies in group cohesion and effectiveness. In this section we propose how principles derived from SGACS theory might inform design in combination with modelling approaches.

Modelling Socio-technical Systems: Local dynamics modelling may also contribute more directly towards specification of CSCW technology. In i*, functional requirements for task support are modelled as goals, while quality requirements (non-functional requirements) are called soft goals. Goals can be decomposed so functional requirements can be expressed for individuals, the whole group, or within-group collaborations. Tasks become computer, manual or semi-automated procedures that fulfil goals. SGACS theory emphasises that "soft" tools, i.e. shared knowledge of procedures, roles and norms, are a key success factor for local dynamics of teams, so i* models enhanced by the theory could provide a template for specifying requirement of shared artefacts, communication and workflow processes in CSCW. Dependency relationships between agents and tasks indicate the need for task support either targeted at individual agents or collaborations between them. KSA analysis can point out where information and decision support requirements need to be targeted in the task-tool network. Analysis of agent-task-tool and communication networks may point to the need for shared awareness support and knowledge management facilities such as aide-memoire lists of procedures, concept maps of issues discussed at meetings, and workflow tools for allocation of responsibilities among members.

Design Principles: To complement modelling socio-technical systems with SGACS-augmented i* models, design principles are proposed based on global dynamics criteria and dependency analysis in task-agent-tool networks. Requirements for collaboration support tools may be expressed as CSCW principles based on global dynamics, such as shared awareness, negotiation support, shared artefact control, etc. The principles may also act as heuristics to critique i* models and collaborative systems designs as a form of expert evaluation.

- *Shared views*: knowledge held by individual group members and their attitudes needs to be visible to other group members. This implies a need for knowledge management and visualisation tools.
- *Collective goal awareness*: the group's collective goal should be communicated to all, with sub-goals and responsibilities of members towards achieving the collective goal. This principle can be supported by shared visualisation of goal trees and progress tracking tools.
- *Support knowledge processing:* information processing and collective knowledge creation should be supported by communication and group decision support systems. Support for this principle will be interpreted in light of the group's collective activity, e.g. managing, design, social activities.
- *Maintain integrity*: by shared goals and information displays, task checklists, goal priorities. This principle may also be supported by shared awareness via social proxies (Erickson & Kellogg, 2000; Viegras & Donath, 1999).
- *Manage conflict*: make decisions and their rationale explicit via design rationale notation or an equivalent. Provide notification, depersonalised as necessary, for group members to express concerns. While we acknowledge that conflict management is a complex topic that requires human negotiation skills to resolve, collaborative technology has a role to play in making the issues visible and shareable.
- *Support agreement*: sorting, prioritising and voting functions, coupled with shared displays of decisions (e.g. gIBIS: Conklin & Begeman, 1988) can help negotiation.

- *Task and agent compatible support*: system functions and communication processes should be based on a local dynamics model of the group's procedures and norms, and support members' behaviour.
- *Relationship support*: the system should support relationship-building appropriate for the group type (crew/team/task force), with shared awareness, representation of shared knowledge and communication.
- *Protect privacy*: allow member contributions to be anonymised according to their wishes. The social identity concepts in SGACS theory indicate a trade-off between shared identity in groups and protecting privacy of their members.

While these principles do not add radically new concepts to CSCW design, they do focus on how technology can support social aspects of collaboration, which constitutes another contribution that SGACS theory adds to design. We could point to other influences such as that the taxonomic view of task forces, crews, teams and social groups may lead to quite different treatment of collaborative support, customisation of member identity, and forms of communication; however, space precludes expansion of these issues.

Discussion

The main contribution of this paper has been to introduce the theory of Small Groups as Complex Systems (Arrow et al., 2000) as a new resource for HCI and CSCW design. The strength of SGACS theory lies in its eclectic foundations in social psychology research, and its formalisation of sociological issues with a model theoretic approach. In contrast, Distributed Cognition (Hutchins, 1995) and Activity Theory (Bertelsen & Bødker, 2003; Bødker, 1991) both place more emphasis on human interaction with artefacts in the world. While SGACS theory can account for these issues in the task-agent-tool network, the theory does not place much emphasis on the role of technology in groups. Instead it provides the means of modelling the contribution of technology within a much richer social view of group interaction.

SGACS theory could provide a modelling framework within which concepts drawn from Activity Theory and Distributed Cognition could be expressed. For example, the knowledge-skills-attributes aspect of local dynamics can be adapted to consider the distribution of knowledge in the world that is emphasised in Distributed Cognition. Conflict is a key concern in Activity Theory; it could be analysed to ascertain whether it may threaten group cohesion or, at a more tolerable level, provoke productive exchanges. We argue that HCI needs to synthesise design influences from several theories and that SGACS theory provides a new set of concepts and models that augment previous contributions.

Activity Theory and Distributed Cognition are claimed to influence user interface design, even though authors admit that influence is indirect. We propose SGACS theory as a semi-formal framework which can function productively as a designer's "tool for thought". It functions first as a diagnostic instrument to find potential problems in socio-technical systems, and secondly, as a source of design principles that can be combined with SGACS-i* models to provide critical insight for improving design. Furthermore, it provides a collection of social psychology knowledge that can be applied to CSCW design to augment the perspectives of other theories. Its particular strengths lie in explicit consideration of social relationships that other theories do not consider.

SGACS lends itself as a modelling-based approach for socio-technical systems analysis and design. Moreover, SGACS theory could be augmented with concepts drawn from principles from Distributed Cognition to develop a design method for collaborative systems. The limitation of complex modelling approaches is the effort required to create models in comparison to the design insight gained. As yet no judgement can be given about SGACS theory on this trade-off. Groups in engineering and design domains are more likely to be task oriented teams, which may place more emphasis of the task-agent-tool network analysis in the theory. However, extensions to SGACS theory may be necessary to model the commitment to individuals to a collective goal and how authority might influence group members' motivation and behaviour. Another limitation is the assumptions made when developing theoretical concepts into models and measurable techniques. Considerable interpretation which depends on human judgement is necessary when transforming explanatory theory into prediction about designs.

References

Abowd, G. D., & Mynatt, E. D. (2000). Charting past, present and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction*, *7*(1), 29-58.

Arrow, H., McGrath, J. E., & Berdahl, J. L. (2000). *Small groups as complex systems: Formation, coordination, development and adaptation*. Thousand Oaks CA: Sage.

Bertelsen, O. W., & Bødker, S. (2003). Activity theory. In J. M. Carroll (Ed.), *HCI models, theories, and frameworks: Toward a multidisciplinary science* (pp. 291-324). San Francisco: Morgan Kaufmann.

Bødker, S. (1991). *Through the interface: A human activity approach to user interface design*. Hillsdale NJ:

Lawrence Erlbaum Associates.
Castro, J., Kolp, M., & Mylopoulos, J. (2002). Towards requirements-driven information systems engineering: The Tropos project. In *Information Systems*. Amsterdam: Elsevier.
Clark H.H. (1996), Using Language, Cambridge University Press.
Conklin, J., & Begeman, M. L. (1988). GIBIS: A hypertext tool for exploratory policy discussion. *ACM Transactions on Office Information Systems*, *64*, 303-331.
Erickson, T., & Kellogg, W. (2000). Social translucence: An approach to designing systems that mesh with social processes. *ACM Transactions on Computer-Human Interaction*, *7*(1), 59-83.
Goffman, E. (1976). Replies and responses. *Language in Society*, *5*, 257-313.
Hutchins, E. (1995). *Cognition in the wild*. Boston MA: MIT Press.
Kahnemann, D., & Tversky, A. (1982). Intuitive prediction: Biases and corrective procedures. In D. Kahnemann, P. Slovic, & A. Tversky (Eds.), *Judgement under uncertainty: Heuristics and biases*. Cambridge: Cambridge University Press.
Kelly, G. A. (1963). *A theory of personality*. W.W. Norton.
McCrae, R. R., & John, O. P. (1992). An introduction to the five factor model and its applications. *Journal of Personality*, *60*, 175-215.
McGrath, J. (1993). Time, task and technology in work groups: The JEMCO workshop study. *Small Group Research: special issue*, *24*(3), 283-421.
McGrath, J. E. (1998). A view of group composition through a sub-theoretic lens. In D. H. Gruenfeld (Ed.), *Research on managing groups and teams: Vol. 1 Composition* (pp. 225-272). Stamford CT: JAI Press.
Mylopoulos, J., Chung, L., & Yu, E. (1999). From object-oriented to goal-oriented requirements analysis. *Communications of the ACM*, *42*(1), 31-37.
Nardi, B. (ed.) (1996). *Context and consciousness: Activity theory and human computer interaction*. Cambridge MA: MIT Press.
Olson, G. M., & Olson J.S. (2000). Distance matters. *Human-Computer Interaction*, *15*(2), 139-178.
Ortony, A., Clore, G. L., & Collins, A. (1988). *The cognitive structure of emotions*. Cambridge: Cambridge University Press.
Rugg, G., McGeorge, P., & Maiden, N. A. M. (2000). Method fragments. *Expert Systems*, *17*(5), 248-257.
Sutcliffe, A. G. (2000). On the effective use and reuse of HCI knowledge. *ACM Transactions on Computer-Human Interaction*, *7*(2), 197-221.
Sutcliffe, A. G. (2002). *The Domain Theory: Patterns for knowledge and software reuse*. Mahwah NJ: Lawrence Erlbaum Associates.
Van der Veer, G. C., Lenting, B. F., & Bergevoet, B. A. J. (1996). GTA: Groupware Task Analysis: Modeling complexity. *Acta Psychologica*, *91*, 297-322.
Vicente, K. J. (1999). *Cognitive work analysis: Toward safe, productive, and healthy computer-based work*. Mahwah NJ: Lawrence Erlbaum Associates.
Viegras, F. B., & Donath, J. S. (1999). Chat circles. In M. G. Williams, M. W. Altom, K. Erhlich, & K. Newman, (Eds). In *Human Factors in Computing Systems: CHI 99 Conference Proceedings, Pittsburgh PA*, (pp. 9-16). New York: ACM Press.
Yu, E., & Mylopoulos, J. (1994). Towards modelling strategic actor relationships for information systems development, with examples from business process reengineering. *Proceedings: 4th Workshop on Information Technologies and Systems, Vancouver*, (pp. 21-28).

A Systems Approach to Resolving Complex Issues in a Design Process

Emad Marashi, John P. Davis

Dept. of Civil Engineering, University of Bristol, Bristol, BS8 1TR, UK.
{Emad.Marashi, John.Davis}@bristol.ac.uk

Abstract: Engineers dealing with large-scale, highly interconnected systems such as infrastructure, environmental and structural systems have a growing appreciation that they deal with complex adaptive systems. We propose a systemic methodology based on discourse and negotiation among participants to help in the resolution of complex issues in engineering design. Issues arise which affect the success of each process. There are a number of potential solutions for these issues which are subject to discussion based on the available evidence assembled from a variety of sources with a range of pedigrees. An evidence-based argumentation is used to assemble and balance the evidence which results in a success measure showing how well each solution meets the system's objectives. The uncertain arguments used by the participants and other imperfect evidences are combined using an extension of the mathematical theory of evidence. This process-based framework helps not only in capturing the reasoning behind design decisions, but also enables the decision-makers to assess the support for each solution. The complexity in this situation arises from the many interacting and conflicting requirements of an increasing range of stakeholders. There is never a 'right' answer, only a satisfactory resolution which this system helps to facilitate.

Keywords: Systems Approach, Process Modelling, Uncertainty, Argumentation, Evidence theory.

Introduction

There is an ever-increasing need to design engineering artefacts and systems capable of meeting stakeholders' requirements in complex, uncertain and dynamic situations. A complex system or problem contains many components and layers of subsystems with multiple, non-linear interconnections that are difficult to recognise, manage and predict (Maxwell et al., 2002). In addition, a complex system involves people, organisations, cultural and political issues and software agents capable of affecting whole or a part of a system. Characterisation of these systems and their components is generally incomplete, often vague, and riddled with significant uncertainties (Hall et al., 2004). An organisation's success in solving complex problems through design will depend largely on its ability to manage the complexity associated with these problems. This requires a methodology and process to reduce and manage the complexity associated with the system. Complex systems can exhibit behaviours which are properties of the whole system. These properties seem more intricate than the behaviour of the individual parts. An effective and efficient design could not usually be achieved without a proper understanding of the relationship between the whole and its parts as well as the emergent properties of the system. A *wicked* and messy problem (Conklin & Weil, 1997) like engineering design has many interlocking issues and consequences which may be unintended. The vast range of stakeholders involved in an engineering design project, e.g. public, client, construction team, designers, financers, managers, governmental agencies and regulating bodies; and their changing requirements from the system, escalates the complexity of the situations. Furthermore, the objectives change in response to the actions taken and each attempt for a solution changes the problem situation. In other words, the problem definition evolves as new possible solutions are considered or implemented. This is in contrast with *tame* problems that are understood sufficiently to be analysed by established methods. The problem statement is well-defined and the solution can be objectively evaluated as right or wrong. In practice, there is no specific boundary between tame and wicked problems in design, but the tame problems are surrounded by the wider wicked problem. In fact, a design team need to be well-equipped with a mixture of capabilities in solving both. As we go down from initial social context to the detailed technical solutions, the problem's face gradually changes from wicked to tame. For a typical design activity of a construction project, this transient change may happens during design stages as shown in Figure 1. Design stages are presented according to BS7000-4, Design management systems (BSI, 1996).

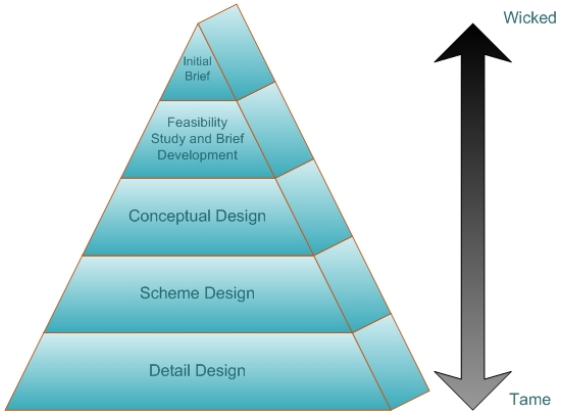


Figure 1 - Transient change of problem nature during design stages

The overall aim of the research described here is to address the need for improvement in managing engineering design and its decision-making process. A system-based approach is adopted to manage complexity, using a multi-level description of design activities. The proposed methodology also provides a framework for representing and capturing knowledge in order to offer a richer picture of design process by articulating process attributes, issues, alternatives and arguments leading to decisions. Communicating design intents between stakeholders and documenting the design process for review, verification, modification and reuse is another aim of this system.

Process Modelling: A Systemic Way to Approach Complexity

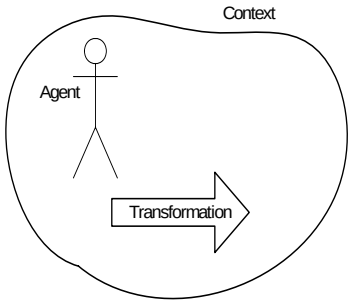
The socio-technical and multi-disciplinary nature of design in engineering systems does not lend itself easily to a pure scientific and technical way of thinking. The *systems* movement which began during 1940s is a set of attempts to explore the consequences of holistic rather than reductionistic thinking. System Thinking (Checkland, 1999) is introduced as a holistic paradigm to understand the situation by seeing the big picture and the connectivity between elements in the situation. It is a method of learning the way towards effective action by looking at connected wholes rather than separate parts. The UML approach to enterprise modelling (Marshall, 1999) and the Soft Systems Methodology (Checkland & Scholes, 1990) are other examples of systems approaches, but recent developments in process modelling (Blockley, 2000), (Davis, 2002) may offer more clarity and accord with the engineering practice. For instance, the implementation of a process-based approach is a core requirement of International Standards for Quality Management Systems (ISO 9001, 2000) and Total Quality Management. As such, it is essential to achieve a unified, simple and intuitive understanding of a process in order to implement the process approach.

The processes can be defined at different levels of definition to give a continuous spectrum of hierarchically structured models. This is in line with the characteristics of a complex system as described by Simon (1999). In this view, complex systems are usually hierarchical, and these hierarchically organised complex systems may be decomposed into sub-systems. Because of their hierarchical nature, complex systems can frequently be described in terms of a relatively simple set of symbols. This allows for a description of system at a range of complexity levels. Such multi-level frameworks must provide a coherent calculus that allows for the transfer of information or knowledge between the levels. For instance, within a complex system such as the rail network, evidence on performance must propagate upwards from the lowest levels to network management levels. On the other hand, government targets must be filtered down to the level of train operators and then to local network managers.

Blockley (1999, 2000) has identified a number of attributes for a process and an algorithm for building a process model based on its attributes. Although Checkland does not use the 'process' term, the notion of 'building purposeful activity models' in Soft Systems Methodology using CATWOE notation (Customers, Actors, Transformation, World-view, Owner and Environment) resembles a similar view.

We describe the process here as a set of activities which realises a transformation. The process elements can be described in terms of the simplest sentence structure, i.e. 'Subject + Verb + Adverbials', which is meaningful in the context of a language. The core element of every process is a transformation, which is a verbal definition of the action in a process. The transformation is carried out by or affects a number of human or software agents in a

specific context. In this definition, the important elements of a process can be categorized in terms of Agent, Transformation and Context (Marashi & Davis, 2004b). This is what we call the 'ACT' model (Figure 2).



Agent: Those who are involved, affected or concerned about transformation
Transformation: The conversion of a state or entity to another state or entity
Context: The situation in which the transformation is meaningful

Figure 2 - Elements of a process using the ACT model

Based on this general categorisation of process elements, a unified understanding of process attributes which enables a complete definition of a process is presented in Table 1. The transformation is an answer to the question 'what' to do as well as 'how' and 'why' to do it. "Selecting the site for windfarm power generation" and "Evaluating the noise level of windfarm" are two examples of process. The root process can be decomposed to a number of sub-processes. The contractual agreement or the scope of work of a project could be used as a guideline for start. In the absence of such information, a careful investigation of the activities and their nature should be done in order to identify the required sub-processes. Although the identification of sub-processes is somewhat subjective, it is constructive to think about each process in terms of three stages of Appreciating, Operating and Controlling (Table 1), as it is discussed in (Checkland, 1999). This leads naturally to the concept of a hierarchically structured set of sub-processes which realises the achievement of the defined objectives of that decision (see also Davis & Hall, 2003). In fact, process modelling can augment the notion of Checkland's purposeful activity systems by introducing a multi-level, connected set of processes which is more manageable and industry-oriented. The ACT model also generalises the CATWOE notation in Soft Systems Methodology to a form that is closer to the structure of natural language.

It is important to recognize that there is no 'right answer' to this identification and the process model is not a unique, one-off outcome for the whole life-cycle of the project. Several different hierarchies should be built by the design team in an iterative way until one emerges which is perceived to be robust enough and practical for the task in hand. It should be borne in mind that the engineering process creates systems for a purpose; that purpose is to satisfy the requirements of the systems' stakeholders. Thus, the characterisation of stakeholder requirements is a crucial sub-process.

At the heart of the methodology must be the recognition that creation and management of real engineered systems requires many disciplines and capabilities to work in harmony. This is one of the weaknesses of the existing engineering approach, which is based on disciplinary demarcations that inhibit interdisciplinary working.

At the lowest level of hierarchy, the achievement of each process may be viewed as addressing one or a number of issues. This could raise a debate which will be structured in an argumentation framework as we will see in the next section.

Evidential Discourse for Engineering

The need for argumentation and discourse is apparent in most complex decision-making situations. In general, a group of people reach a decision through debate and negotiations. Each stakeholder may have his own sets of preferences and view points, with arguments for or against a potential solution. Argumentation is primarily useful for tackling wicked and messy problems (Conklin & Weil, 1997).

Table 1- Attributes of a process based on the ACT model

Agent	Transformation	Context
Who?	Why? /What? /How?	Where? /When?

Roles (people): Customer Client Stakeholder Sponsor Owner Manager Worker	Appreciating: Objectives Purpose Scope Issues Criteria Success Failure	World-view Time Place Description of Situation: Hierarchy Uncertainty
Roles (other agents): Function	Operating: Activity Input Realization Output Sub-activities	Environment: Resources Constraints Hazards Risks
Roles: Responsibility Authority Accountability Communication	Controlling: Performance Measurement Monitoring	Social/Cultural dimensions Roles Norms Values Political situations
Subject , Object	Verb	Adverbials

In contrast to tame problems, the definition, requirements and criteria for whether a solution has been reached are not well-defined. Ill-structured problems, like those encountered in design and management, lack the predetermined linear route through problem solution stages applicable for structured problems. This is the reason why solving messy problems is an argumentative process requiring logical as well as informal reasoning. The study of argumentation is deeply rooted in various disciplines such as philosophy, logic and linguistics, but it is important to note that argumentation study tries to deal with the verbal, contextual, situational and other pragmatic factors of communication process in areas where logic can not adequately address the situation..

Argumentation is defined as "*The action or operation of inferring a conclusion from propositions premised* "(OED, 2005). The study of argumentation dates back to Greek antiquity, around 2400 years ago. The development of informal logic and argumentation theory within philosophy has represented a backlash against formal logic. Despite immense power and wide application of formal logic, it is not a suitable choice for representing and characterising complex, natural and real world language and arguments. Toulmin developed an intermediate approach between formal proofs of logic and persuasive strength of rhetoric (Toulmin, 1958). He developed a "layout of argument" which has been used largely for the analysis, evaluation and construction of arguments, especially in jurisprudence context.

According to Toulmin (1958), the argumentation process starts with formulation of a problem in the form of a question. A list of possible solutions is taken into consideration in the next stage, setting aside the solutions that appear inadequate straight away. The possible solutions are then weighed up against each other. A choice has to be made between possible solutions in order to select "the best" one, though it might be difficult to arrive at a solution in some fields of argumentation which deal with soft aspects of human affairs. An open-ended, dialectical process of collaboratively defining and debating issues, having its roots in dialectic of Aristotle, is a powerful way for reaching a consensus and conclusion. This perspective motivated the development of Issue-Based Information Systems (Kunz & Rittel, 1970) as a framework for modelling argumentation. Having its background in planning and policy problems, IBIS addresses design problems by using argumentation structures to facilitate a discussion amongst the stakeholders about issues, which allows the problem to be explored and framed. IBIS tries to identify, structure and settle issues raised by problem-solving groups. Issues are brought up and disputed because different positions are possible. This framework has also been developed through the Compendium methodology (Selvin et al., 2001) and HERMES system for multiple criteria decision-making (Karacapilidis & Pappdias, 2001). Fletcher & Davis (2003) also proposed a framework for simulation and capture of dialectical argumentation in a complex situation. Argumentation structures based on IBIS and QOC (Question, Option, Criteria) have also attracted a lot of attention in developing design rationale systems (Shum & Hammond, 1994).

The argumentation structure in IBIS model consists of *Issues*, *Positions* and *Arguments*. The issues are brought up by the participants and are subject to debate. Issues normally have the form of a questions or a controversial statement, which is raised, argued, settled, dodged or substituted. Each issue consists of a set of positions or options which represent the possible answers, ideas or choices of action that could be taken in response to that issue.

Arguments are asserted by participants to support or rebut a position. The argumentation structure attached to each process is presented in Figure 3.

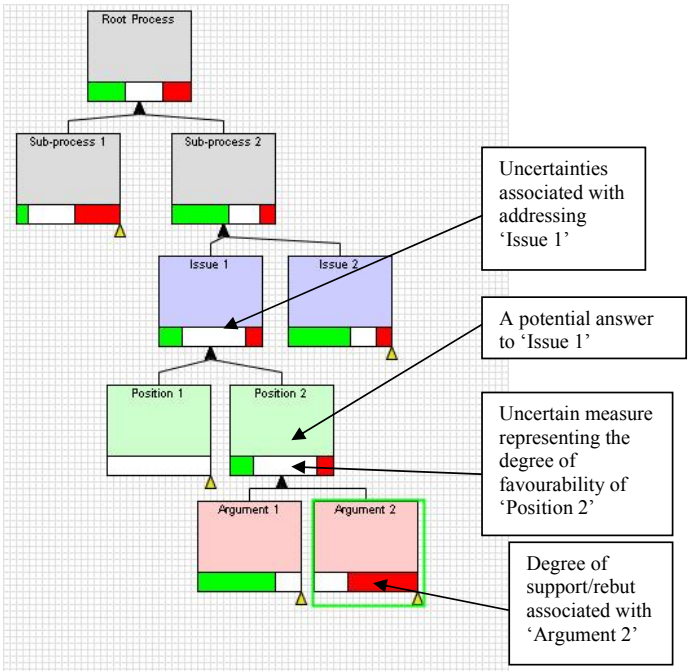


Figure 3 – Evidential Discourse for ENgineering (EDEN) framework (Marashi & Davis, 2004b)

This graphical representation of argumentation enables externalization and communication of discourse among participants and stakeholders. This integrated framework is called EDEN (Evidential Discourse for ENgineering) and its software implementation is under way in the Civil Engineering Systems Group at the University of Bristol, through extending its predecessor software tools Juniper and PeriMeta (Davis & Hall, 2003). From a linguistic point of view, this model allows for the representation of participle phrases (processes), interrogative statements (issues) and declarative statements (positions and arguments) which adds to the expressiveness of the process modelling system.

The arguments taken for or against a position are the basis for developing an uncertain success measure for the performance of that position or option. It shows the level of acceptance of that option in comparison with other options under the same issue. These measures provide a clearer understanding of which alternative solution is more prominent at the moment. In the same way, the uncertain measures attached to issues represent the successfulness of the debate in addressing that issue. These measures are combined and propagated up through the hierarchy using mathematical theories of uncertainty which is the subject of discussion in the next section. The advantage of this argumentation framework compared to previous works based on IBIS is that it provides the computational support for assessing the strength of arguments as well as a graphical representation of the connectivity of the argumentation elements. The use of visual language has been shown to be a powerful tool in the facilitation of dialogue and debate (Horn, 1998). Like any other form of knowledge representation and elicitation, the added expressive dimension of argumentation would however bring in its own overhead to the designers. This could be balanced by longer term benefits that the system can provide during design review and reuse. Argumentation is also a way of recognising, communicating and protecting stakeholder's social and technical interests and requirements (Goguen, 1994).Another difficulty arises with cultural and political dimensions of knowledge sharing, especially during the elicitation of tacit knowledge in an organisation (Turban & Aronson, 2001). An example of the application of this framework has been presented later in this paper.

Combining Uncertain Evidence

In many cases, there is insufficient knowledge to allow for the perfect transfer of information between levels and hence the incorporation of uncertainty management and propagation is vital for any multi-level system. Theories of evidence allow us to combine uncertain pieces of information issued from various sources dealing with the same subject. Unlike the Bayesian approach, these models do not require the additivity of beliefs, so ignorance and inconsistency in sources of evidence is also permitted.

The uncertain positions are combined using the generalized natural combination rule which has been derived from a generic method of aggregation of uncertain evidence, called Triangular-norm-based Combination Rule (Marashi & Davis, 2004a). This new combination method has emerged from an amalgamation of generalized fuzzy set operations with belief functions computations. This rule allows the aggregation of uncertain information with various levels of dependency between bodies of evidence.

Let Ω be a finite set of elements called the *frame of discernment*. It can be seen as a set of possibilities or hypotheses under consideration. By definition, a mass assignment could be built over the power set of Ω, such that $m: 2^{\Omega} \rightarrow [0,1]$ and

$$\forall X \subseteq \Omega \qquad \sum_{X \subseteq \Omega} m(X) = 1$$

$m(X)$ represents that part of belief that supports X as being the true hypothesis.

Now lets consider two pieces of evidence represented by mass assignments m_1 and m_2, supporting proposition $A \subseteq \Omega$. The aim of the TCR combination rule in Eq. (2) is to sum up the masses assigned to a subset A during the meet of two or more evidence, and to redistribute the conflicting mass $m(\varnothing)$ on a specified subset $P \subseteq \Omega$ according to weighting factors *w* in normalization step. The combined mass can be expressed as:

$$m(A) = \sum_{X \cap Y = A} m_{XY} + w(A, \mathbf{m}).m(\varnothing) \quad \forall A \subseteq \Omega \setminus \varnothing$$

$$m_{XY} = T^{XY}(m_1(X), m_2(Y)) \quad \forall (X,Y) \in S \subseteq F_1 \times F_2 \qquad (1)$$

$$m(\varnothing) = \sum_{X \cap Y = \varnothing} m_{XY} \qquad w(A, \mathbf{m}) = 0 \;\; A \not\subset P \qquad \sum_{A \subseteq P} w(A, \mathbf{m}) = 1, \forall A \subseteq P$$

where F_1 and F_2 are sets of focal elements for mass assignments m_1 and m_2, respectively. S is a subset of $F_1 \times F_2$ for which the masses are allocated using a t-norm. P is a subset of Ω on which the conflicting mass should be distributed and w are weighting factors associated to each subset of P. Function T^{XY} is a triangular norm. The selection of elements of S and values for m_{XY} should be in such a way that satisfies a set of constraints imposed by support pairs, i.e.

$$S_{n_1}(A) = \sum_{Y \in F_2} m_{AY} \qquad \forall A \in F_1$$

$$S_{n_2}(A) = \sum_{X \in F_1} m_{XA} \qquad \forall A \in F_2$$

$$\sum_{X \in F_1} \sum_{Y \in F_2} m_{XY} = 1$$

Now consider the following special case for a bipolar frame of discernment:

$$\Omega = \{A, \overline{A}\}, \; 2^{\Omega} = \{\varnothing, \{A\}, \{\overline{A}\}, \{A, \overline{A}\}\}$$

The following mass assignment for this representation introduces a support pair (Baldwin, 1986) as $A: [S_n, S_p]$ where S_n is the necessary support and S_p is the possible support for proposition A:

$$m(A) = S_n(A) \qquad m(\overline{A}) = 1 - S_p(A) \qquad m(\Omega) = S_p(A) - S_n(A)$$

This concept with its graphical illustration called the *Italian Flag* is used to represent the support pairs (see Figure 7). The green area on the left hand side represents evidence in favour of A, the red area in right is the evidence against A, and the white area is the amount of ignorance about A.
The generalized natural combination rule is derived using the following assumptions in Eq. (1) (Marashi & Davis, 2004a):

$$m_{AA} = T_\lambda^F(m_1(A), m_2(A))$$

$$m_{\overline{A}\overline{A}} = T_\lambda^F(m_1(\overline{A}), m_2(\overline{A}))$$

$$m_{A\overline{A}} = T_{\frac{1}{\lambda}}^F(m_1(A), m_2(\overline{A}))$$

$$m_{\overline{A}A} = T_{\frac{1}{\lambda}}^F(m_1(\overline{A}), m_2(A))$$

$$T_\lambda^F(x,y) = \begin{cases} \min(x,y) & \lambda = 0 \\ x.y & \lambda = 1 \\ \log_\lambda\left(1 + \frac{(\lambda^x - 1)(\lambda^y - 1)}{\lambda - 1}\right) & \lambda \in]0,1[\cup]1,\infty[\\ \max(0, x+y-1) & \lambda = \infty \end{cases}$$

where $T_\lambda^F(x,y)$ is the Frank's triangular-nom function (Frank, 1979).

The parameter λ can be used to model various dependency assumptions between bodies of evidence.

Climate Change Impacts on Electricity Supply Indusry: a Case of Application

There is growing evidence that the UK climate is changing over the coming decades due to a combination of natural and human causes. The Electricity Supply Industry (ESI) is an example of a complex utility which needs to tackle the climate change by:

- complying with the policies for reducing greenhouse gas emissions
- planning to adapt the industry to the unavoidable impacts of the climate change

The ESI is involved with a diverse range of stakeholders. Since privatisation, the gas and electricity industries have become fragmented and the central long-term planning of the pre-privatisation period has largely disappeared. The main players involved here are the generators, transmitters, distributors, suppliers which provide electricity to the customers under Ofgem (Office of Gas and Electricity Market) regulations in Britain. These relatively large groups of stakeholders place their own particular performance demand on the system. Effective management of the ESI must recognize this, and treat specific demands, such as those arising from climate change, in a holistic manner alongside other system requirements.
Energy supply, and particularly electricity generation, was responsible for a quarter of the UK's greenhouse gas emission in 2000. Consequently, ESI is under growing pressure to reduce the consumption of fossil fuels by increasing the use of renewable sources. The Energy White Paper (DTI, 2003) has set a 10% target for the fraction of the UK's electricity that should be supplied from renewable energy by 2010. A large part of this renewable energy will be provided by windfarms both on and offshore, as wind power will be the most competitive form of renewable energy in the medium term. Through the following brief example, we demonstrate how the EDEN methodology and tool can be used during the course of the feasibility studies and design of a new windfarm.
Figure 7 shows a snapshot of a process model for designing a windfarm. The root process "Designing the windfarm power generation" has been broken down into a number of sub-processes, namely "Developing the initial brief", "Performing the feasibility study" and "Performing the basic and detail design". Feasibility study of a power generation plant requires a wide range of technical, social, economical and environmental studies. "Selecting the site location" is an example of a process that not only needs to satisfy technical specifications of different design disciplines, but at the same time is involved with getting permission from local authorities, inquiring about public viewpoints and dealing with environmental campaigners. For this example, we only focus on two issues that have been raised which are the subject of discussion as shown. The first question here is to identify objectives/criteria, which has been answered subsequently through quantitative and descriptive explanations. The objectives/criteria can be translated to performance indicators using an appropriate value function. The value function can be numeric or linguistic depending on the nature of evidence. The importance of each criterion can be adjusted by setting the weighting factors attached to that performance indicator. Figure 5 shows an example of an S-shape value function

for assessing a site based on its distance from the dwellings. Two alternatives are under consideration for the location of this new windfarm. We assume that the local authority requires a minimum distance of 2 km from dwellings in order to minimise visual domination, noise and reflected light. The argument for Alt-1 which is located at 3 km from the city has been mapped to an Italian Flag using the value function in Figure 5. The evidential value of an argument is being assessed by measuring the value of its attributes against a set of criteria assigned to that argument. This is what we call the justified evidence for or against an option as it is based on the explicit criteria. The importance of the argument can also be adjusted through the importance factor assigned to its link. The combined interval value represents the uncertain support measure of the corresponding alternative. In a similar way, other objectives/criteria, e.g. wind speed, are translated into a value function. One of the important qualitative aspects of a windfarm is its visibility. This is one of those subjective issues that can not be easily assessed without incorporating public and experts viewpoints. A fuzzy mapping has been used to handle qualitative and linguistic assessment of the visual aspects. Opinions can be expressed in terms of a linguistic scale like very poor, poor, medium, good and very good, together with the confidence of the evaluator on her assessment. Figure 6 shows the value function corresponding to a 'good' assessment of visual aspect with medium confidence. Note that "Connection to the network is easy" has been added as an argument without referring to an explicit criteria. These types of arguments enable the participants to express their opinions and implicit judgments in an informal way. The participants are allowed to express their subjective opinions without a reference to a performance indicator. This feature of the methodology enables the inclusion of expert subjective opinions into the decision-making process.
The graphical representation of the arguments and its effect on supporting or rebutting a potential option helps the design team to easily communicate the reasons behind their decisions. The arguments should not be necessarily independent of each other, as the uncertainty calculi is able to deal with various levels of dependency between bodies of evidence. Each option should be assessed against the same set of performance indicators related to that issue, based on the value of the corresponding attribute of that option which could also be uncertain.

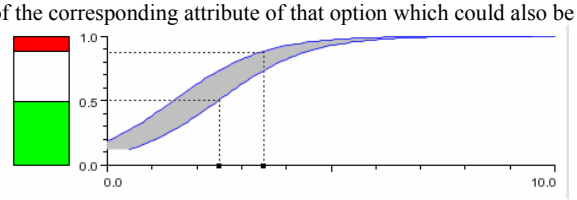


Figure 4 - Value function for site distance from the dwellings; support pair for x=3±0.5 km

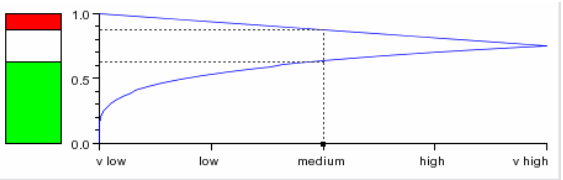


Figure 5 - Value function for 'good' visual aspect with 'medium' confidence

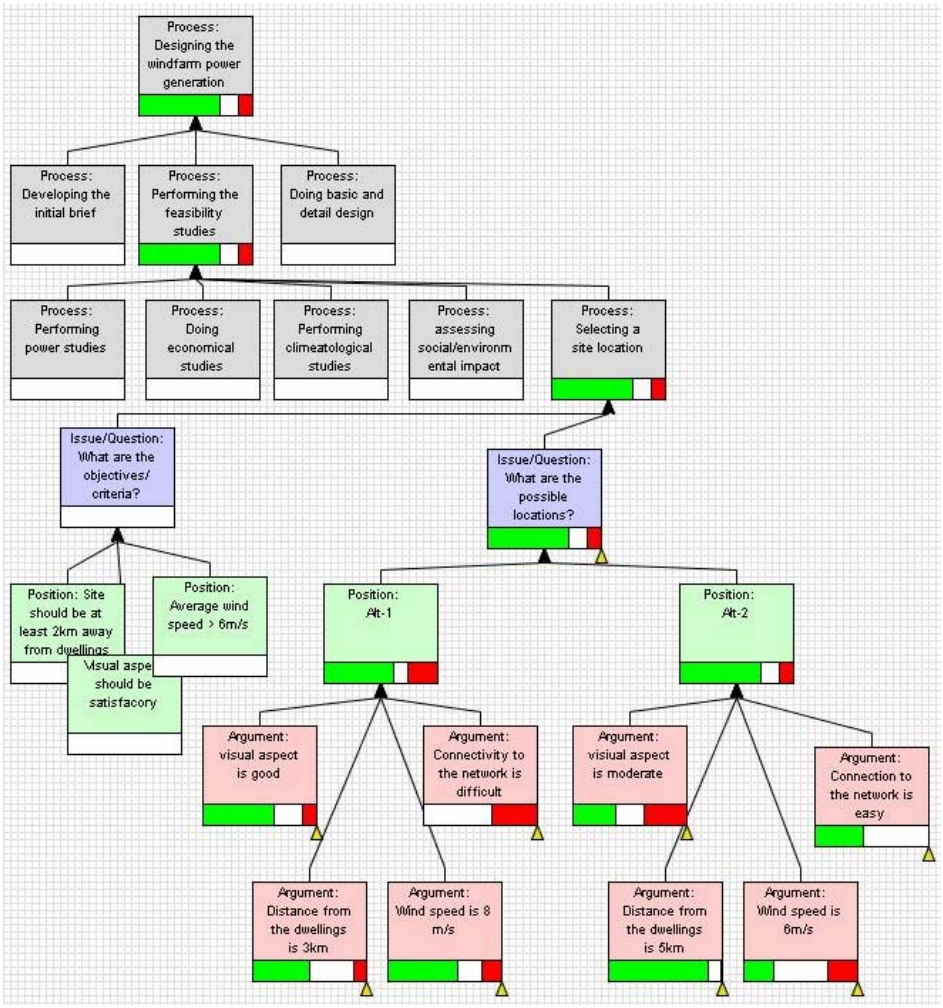


Figure 6 - A Snapshot of argumentation during a design decision-making process

Conclusion

An integrated framework is presented based on process decomposition and argumentation to help designers tackle complex, messy and ill-structured problems. The methodological steps are as follows:

- Understand the problem situation
- Decompose the complex system to its sub-systems and sub-processes
- Identify process objectives and criteria
- Resolve issues based on evidence through discourse and negotiation
- Decide on appropriate action and solution
- Continually review and update the above steps

A systemic view of the activities in the design process recognises the need for handling uncertainty, which is being addressed through a novel use of evidential and argumentation theories. Design rationale can be represented among of stakeholders in a more intuitive way. Visualizing the trend of arguments, with appropriate calculi for assessing the strength of claims, results in a clearer picture of the design decisions.

Acknowledgement

The authors acknowledge the support of the EPSRC project entitled 'A Generic Process for Assessing Climate Change Impacts on the Electricity Supply Industry and Utilities', grant no. GR/S18922, in generation of the case study mentioned above.

References

Baldwin, J.F. (1986). Support Logic Programming. Int Journal of Intelligent Systems 1: 73-104.
Blockley, D. & Godfrey P. (2000). Doing it differently: Systems for Rethinking Construction, Thomas Telford, London, UK.
Blockley, D. (1999). Process modelling from reflective practice for engineering quality. Civil Engineering and Environmental Systems (16): 287-313.
British Standard Institute (1996). BS 7000-4. Design management systems, Guide to managing design in construction, BSI.
Checkland P. & Scholes, J. (1990). Soft Systems Methodology in Action, John Wiley, UK
Checkland, P. (1999). Systems Thinking, Systems Practice: Includes a 30-year retrospective, New Ed., John Wiley, UK.
Conklin, J. & Weil, W. (1997). Wicked problems: naming the pain in organizations. GDSS Inc. (http://www.3m.com/meetingnetwork/readingroom/gdss_wicked.html)
Davis, J.P. & Hall, J.W. (2003). A software supported process for assembling evidence and handling uncertainty in decision-making. Decision Support Systems (35) 3: 415-433.
Davis, J.P. (2002). Process Decomposition. Proc. Int. Conf. on Hydroinformatics, Cardiff.
Department of Trade and Industry (2003). Our Energy Future: Creating a Low Carbon Economy, Cm 5761, HM Stationery Office, London, UK.
Fletcher, A. & Davis J.P. (2003). Dialectical evidence assembly for discovery. Proceedings of Discovery Science. Lecture Notes in Computer Science 2843: 100-113.
Frank, M.J. (1979). On the simultaneous associativity of F(x, y) and x + y - F(x, y), Aequationes Mathematica, 19:194–226.
Goguen, J. (1994). Requirements Engineering as the reconciliation of social and technical issues, Req. Engineering: social and technical issues, 165-194.
Hall, J.W., Le Masurier, J.W., Baker-Langman, E.J., Davis, J.P. & Taylor, C.A. (2004). A decision-support methodology for performance-based asset management. Civil Engineering and Environmental Systems (21) 1:51-75.
Horn, R.E. (1998). Visual Language: Global Communication for the 21st Century. MacroVU Inc., USA ISBN: 189263709X
International Standard Organization (2000). BS EN ISO 9001. Quality Management Systems. BSI.
Karakapilidis N.& Papadias D. (2001). Computer supported argumentation and collaborative decision-making: the HERMES system. Information Systems 26: 259-277.
Kunz W. & Rittel H. (1970). Issues as elements of information systems. Working Paper No. 131, Inst. of Urban and Regional Development, University of California, USA.
Marashi, E. & Davis J.P. (2004a). A generic rule for combination of uncertain information in the framework of evidence theory, 10th Int. Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU'04), Perugia, Italy, pp. 1709-16
Marashi, E. & Davis, J.P. (2004b). A framework for supporting discourse in the design decision-making process, 4th International Conference on Decision-Making in Urban and Civil Engineering, Porto, Portugal
Marshall, C. (1999). Enterprise Modelling with UML, Addison, Wesley.
Maxwell, T.T., Ertas, A. & Tanik, M.M. (2002). Harnessing complexity in design. Journal of Integrated Design and Process Science (6) 3:63-74.
Oxford English Dictionary (2005). online, http://dictionary.oed.com/, viewed: 25/01/05
Selvin, A., et al. (2001). Compendium: Making Meetings into Knowledge Events, Knowledge Technologies, March 4-7, Austin TX, USA.
Shum, S.B. & Hammond, N. (1994). Argumentation-based design rationale: what use at what cost. Int. J. Human-Computer Studies (40) 4:603-652.
Simon, A. H. (1999). The Sciences of the Artificial, Third Edition, The MIT Press, Cambridge, MA. USA.
Toulmin, S.E. (1958). The Uses of Argument, Cambridge University Press, Cambridge.
Turban E., Aronson, J.E. (2001) Decision Support Systems and Intell. Systems, 6th Ed., Prentice Hall, NJ.

A Communication Tool Between Designers and Accidentologists for the Development of Safety Systems

Walid Ben Ahmed*, ***, Mounib Mekhilef*, Michel Bigand**, Yves Page***

*LGI – Laboratory of Industrial engineering, Ecole Centrale de Paris, 92295 Châtenay-Malabry), FRANCE, E-mail: {walid, mekhilef@lgi.ecp.fr}.

**Research Group in Industrial Engineering, Ecole Centrale de Lille, BP 48, 59651 Villeneuve d' Ascq cedex, FRANCE, E-mail: Michel. Bigand@ec-lille.fr

***LAB (PSA-Renault), Laboratory of Accident research, Of Biomechanics and studies of the human behaviour, 132, rue des Suisses 92000 Nanterre, FRANCE, E-mail: yves.page@lab-france.com

Abstract: Designers and accidentologists have to collaborate in order to develop new safety systems. Accidentologists recognize *Accident Scenario* as a powerful tool to provide designers with the required knowledge about accident. However, an accident scenario has to be presented in a way that both designers and accidentologists can understand and use. The fact that designers and accidentologists do not share the same viewpoints, neither the same models to analyze an accident, nor the same technical language makes their communication a complex task in a design process. To address this issue, we use the *systemic approach* (a complex system modelling approach) to develop a new methodology allowing constructing multi-view accident scenarios.

Keywords: safety system development, accident scenario, systemic approach, multi-view modelling, complex system modelling.

Introduction

Several approaches, methods and tools exist in the literature to support designers developing new systems and functions. Functional Analysis and Query Functional Deployment (QFD) for example allow a designer to structure his design. However, these methods suppose that the main functions (functions related to the requirements) exist. Therefore, these methods only allow the deployment of the main functions and the structuring of the design space. When one deals with new systems development, the primary need is a tool to build the design space. In other words, we need tools to define functions to be realized in technical solutions. According to our records, there is a lack of research in literature dealing with this issue.

Our research is carried out in the LAB (Laboratory of Accidentology, Biomechanics and Human Behaviour), which is a shared laboratory between the two main French car manufacturers, PSA (Peugeot-Citroën) and Renault. This research is intended to provide safety system designers with accidentlogy knowledge to allow them to understand accident behaviour and therefore to develop new road safety systems.

Developing safety system is a complex task due to the fact that several disciplines have to be combined to achieve it. Indeed, designers who are generally specialized in mechanics and electronics, collaborate with accidentologists who are specialized in mechanics, biomechanics, ergonomics, infrastructure and psychology. Hence, the main issue consists of making possible the communication between these different skills.

In the LAB, brainstorming sessions are one of the means used to allow the communication between accidentologists and designers. The aim of these sessions is to understand the accident mechanisms and to propose new road safety counter-measures that designers may use as an input to elaborate new safety systems. However, there are many issues that have to be addressed in order to carry out successful brainstorming session:

- Designers and accidentologists do not share the same viewpoints, neither the same models to analyze an accident, nor the same technical language. For instance, a psychologist focuses more on the driver's information processing aspects whereas a designer is more interested in the mechanical aspects;
- There are many different approaches and viewpoints that can be used to analyse a road accident in order to understand the failure mechanisms. Some of these approaches focus on the accident's causal aspect. Others focus on the accident's sequential aspect (Brenac, 1997; Brenac and Fleury, 1999), or on the human mechanisms of error production and of information processing (Fuller and Santos, 2002; Van Elslande and Alberton, 1997). Some studies in cognitive psychology analyze the driver's behaviour as a process of skill learning and automatization (Summala, 2000), or as a risk management process (Fuller, 2000). Thus, each of these approaches focuses on a

specific aspect of the accident. However, when considering the complexity of the accident, several approaches should be combined in order to handle this complexity;

- Another difficulty that designers and accidentologists are facing when they work together in brainstorming sessions is related to the nature and forms of the accident data colleted in the databases. Indeed, using the thousands of accidents characterized by hundreds of attributes is a hard, time-consuming and thereby inefficient task.

Hence, the aim of our paper is to elaborate a tool intended to represent accidentology knowledge in a way that designers and accidentologists can use. In other words, we aim at developing a tool that represents accidentology knowledge for each operator in his own viewpoint. This may make easier and more efficient the communication between the various skills involved in safety system development.
In the first section of this paper, we present an overview of the use of accident scenarios as a communication tool between designers and accidentologistes. In the second and third sections we present respectively the systemic approach and its use to integrate different viewpoints stemming from designers and accidentologists in design process.

Accident Scenarios: a Powerful Interface Between Designers and Accidentologists

A scenario is a prototypical behaviour of a group of subjects or objects (customers, accidents, users, etc.) with similarities. Scenario-based approaches are used in several fields (Leite et al., 2000). For instance, in economy and finance, scenarios are used to anticipate market behaviour and thereby to perform adequate plans to address economical issues. Scenarios are also used in risk analysis in project management, nuclear installation etc. (Scheringer et al., 2001). They allow risk anticipation and handling. They are also used in software engineering as a tool to understand the user behaviour in order to anticipate the different software use-case (Caroll, 1995,1998; Gandon and Dieng, 2001; Jarke et al., 1998).
Accidentologists assume that similar accident factors entail similar safety countermeasures (Brenac and Megherbi, 1996; Fleury et al., 1991; Van Elslande and Alberton, 1997). Based on this assumption, accidentologists in the LAB recognize *Accident Scenario* (AS) as a powerful tool to provide safety system developers with the required knowledge. In Figure 1, we present an accident scenario example. It is a synthetic description of 30 road accidents. It is one of 18 scenarios we elaborated using a sample of 750 road accidents.

> 4% of accidents in the database concern the following situation: *"The accident happened at a junction of two main roads. The weather was sunny and the road surface was dry. A driver came up to the roundabout at the junction. He did not know which direction he had to take, and was concentrating on the road signs. As he reached the roundabout, he glanced left quickly, and thinking that the road was clear, pulled out. He declared his speed to be about 20 km/h. The crash barrier that runs round the middle of the roundabout reduces the visibility of vehicles coming from the left."*

Figure 10 - Example of an accident scenario.

Obviously, using the scenario presented in Figure 10 in a brainstorming session for example is easier than using the 30 accidents summarized by this scenario. Indeed, each accident in the data base is characterized by 900 attributes and thereby the use of the detailed cases is time-consuming and inefficient. Hence, accident scenario provides accidentologists and designers with a synthetic description of a group of accident with an adequate granularity level. Thus, instead of using 750 detailed accident cases in discussing session between accidentologists and designers, we use only 18 scenarios summarizing the different accident cases.
To elaborate such scenarios, several researches were carried out in literature. In (Brenac and Megherbi, 1996; Fleury *et al.*, 1991; Van Elslande and Alberton, 1997), the authors propose an expert approach: expert clusters accidents manually according to their similarity. Next, he elaborates a synthetic description for each cluster. However, this approach has some drawbacks related to the fact that expertise is expensive and scenarios depend on the expert viewpoint and discipline. Moreover, different granularity levels and ways of representing accident scenarios exist. Indeed, several models may be used to present accident scenario. A *Driver-Vehicle-Environment (DVE) model* may be used to describe what happened to each of these three components (i.e. driver, vehicle and environment). *Information processing model* is another model that can be used to represent accident scenarios (Van Elslande and Alberton, 1997). It consists of describing the scenarios according to the following steps: *perception*, *diagnosis*, *prognosis*, *decision* and *action*. A *sequential model* that presents accident as a sequence of five steps (*normal driving step*, *failure step*, *emergency step* and *crash step*) may also be used (Brenac and Fleury, 1999).

Other studies propose data-mining techniques in order to elaborate accident scenarios. In (Chovan et al., 1994; Najm et al., 2001; Sohn and Lee, 2003; Sohn and Shin, 2001), authors propose classification techniques to elaborate accident configurations. (Page, 2002; Page et al., 2004) propose clustering techniques to perform accident scenarios. However, data-mining techniques suffer from some drawbacks: the interpretation of the statistical clusters is a hard task for experts.
We propose the combination of the expert and the data-mining approaches. Concretely, we propose to apply clustering techniques[9] to regroup similar accidents. In a second step, we perform a projection of the obtained cluster according to chosen viewpoints. Thus, we allow the interpretation of accident scenarios as well as their representation according to the viewpoints and models that accidentologists and designers may chose (DVE model, sequential model, information processing model, etc.).
The main issue is: *how to identify the different viewpoints and models that are relevant to analyze road accident in order to define new countermeasures*? To address this issue, we propose to use the systemic (also called cybernetic) approach (Ashby, 1965; Le Moigne, 1974; Von Foerster, 1995) in order to identify the relevant viewpoints and models.

A Systemic Approach for Viewpoints Integration

Behaviour in road accidents is complex. This is not due to the number of components involved in the accident occurrence, neither the number of variables interacting during the accident. Most of all, it is the non-linearity and the impossibility to predict the DVE system behaviour that entails this complexity. This unpredictability is notably due to the fact that human actions are strongly involved in accident causation, and that human behaviour is unpredictable. Furthermore, during the road accident, the DVE system performs some functions (i.e. perception, interpretation, anticipation, decision, action), which generate transformations (i.e. new situation, new interpretation, new purpose, new requirement, etc.), which in turn generate new functions and behaviours, etc. DVE behavior then be described through feedbacks and recursive loops. According to Miller's definition of a living system (Miller, 1995), the DVE is an open and living system as much as each component (i.e. driver, vehicle, infrastructure, traffic, etc.) is constantly interacting with its environment by means of information and matter-energy exchanges. Due to these feedbacks and recursive loops, it is impossible for designers and accidentologists to identify with exhaustiveness and certainty all the failures and dysfunction mechanisms occurring in a road accident.

Moreover, a same accident may be seen differently according to the analyst viewpoint. We assume that each expert in accidentology and each designer have an individual perception of the same phenomenon. Our assumption is based on constructivist foundations, which assume that knowledge depends on how the individual "constructs" meaning from her/his experience. A system, in a constructivist perspective, is recognized as a representation of reality seen by some people in a given context.

Our approach is then intended to identify and integrate the various viewpoints in accident scenarios construction and interpretation. For this purpose, we propose the *systemic approach* (Le Moigne, 1999) as a shared architecture between accidentologists and designers in order to understand and analyze accident scenarios.

The systemic approach assumes that to handle a complex behaviour, it is fundamental to make junction between the *ontological*, *functional*, *transformational* and *teleological* viewpoints (Le Moigne, 1999). We use these viewpoints to analyse accident behaviour:

- **The ontological viewpoint** (i.e. what is the system?): it allows a structure-oriented and contextual analysis of the system. In other words, it represents the sub-systems (the driver, infrastructure, traffic, ambient conditions, vehicle, etc.), their taxonomic groups, their contexts (the driver's professional status, family status, etc.), their structures, as well as the various interactions between these sub-systems and their components;
- **The functional viewpoint** (i.e. what does the system do?): it allows a function-oriented analysis of the system. It represents the global process of the DVE functioning during the road accident, which combines several procedures (perception, diagnostic, prognostic, decision and action) (Van Elslande et al., 1997);
- **The transformational (or evolutionary) viewpoint** (i.e. how does the system evolve? What does it become?): it allows a transformation-oriented analysis of the system. The DVE system behaviour can be described as an evolution that goes through several states. The transformational viewpoint integrates the accident's sequential and causal models developed by the INRETS and described in the next section (Brenac, 1997; Fleury et al., 2001);

[9] We used k-means algorithm (MacQueen, 1967).

- **The teleological (or intentional) viewpoint** (i.e. what is the goal or intention of the system?): it allows a goal-directed analysis of the accident. In other words, it assumes that each of the DVE system components or functions has to serve a purpose in an active context in order to ensure the safety of the DVE system.

In the next section, we show how to use the systemic viewpoints in order to provide accidentologists and designers with a multi-view analysis tool of accident scenarios.

A Multi-view Interpretation of Accident Scenario

Using the systemic viewpoints presented in the previous section, we developed a software that enables us to represent the same scenario according to different models specific to different fields, i.e. safety system design field and accidentology fields. Each scenario user has the possibility to represent the scenario according to his own model.

Our approach is described through the following steps:

1. Find and/or construct accident representation models according to each systemic viewpoint. For example, the *DVE model* is assigned to the ontological view. The *sequential model* is assigned to the transformational view. The *information processing model* is assigned to the functional view etc.
2. Each model is composed of one or more concepts. For example, "*Normal driving step*", "*Failure step*", "*Emergency step*" and "*Crash step*" are the concepts composing the *sequential model*. "*Perception*", "*Diagnosis*", "*Prognosis*", "*Decision*" and "*Action*" are the concepts composing the *information processing model* etc.
3. Each concept is characterized by one or more attributes. Each attribute may characterize many concepts in different models. For example, the attribute "*steering angle*" characterizes, at the same time, the concept "*Driver/Vehicle interaction*" in the *DVE model*, the concept "*Emergency*" in the *sequential model* and the concept "*Action*" in the *information processing model*. In a sense, the attributes classification according to the model concepts can be perceived as the construction of *metadata* since it is a "*data about data*". Figure 11 shows how we use XML to represent these metadata and how an attribute (e.g. "*steering angle*") is assigned to various concepts.

```
<?xml version="1.0"?>
<Accident_Metadata>
<Viewpoint>
  <ViewpointName> Ontological_View </ ViewpointName >
 <Model>
  <ModelName> DEV_Model </ModelName>
  ...
  <Concept>
      <ConceptName> Driver/Vehicle interaction </ ConceptName >
      <Attributs>
          Steering angle
      </Attributs>
  </Concept>
  ...
  </Model>
  </Viewpoint>
<Viewpoint>
  <ViewpointName> Functional_View </ ViewpointName >
 <Model>
  <ModelName> Information_Processing_Model </ModelName>
  ...
  <Concept>
      <ConceptName> Action </ ConceptName >
      <Attributs>
          Steering angle
      </Attributs>
  </Concept>
  ...
  </Model>
  </Viewpoint>
...
</ Accident_Metadata >
```

Figure 11- An XML representation of the metadata: each attribute is assigned to several concepts according to the various models.

4. Since the accident clusters are characterized by attributes and since these attributes are classified according to the different concepts in the different models, we can perform a multi-view projection of a scenario accordingly (see Figure 12).

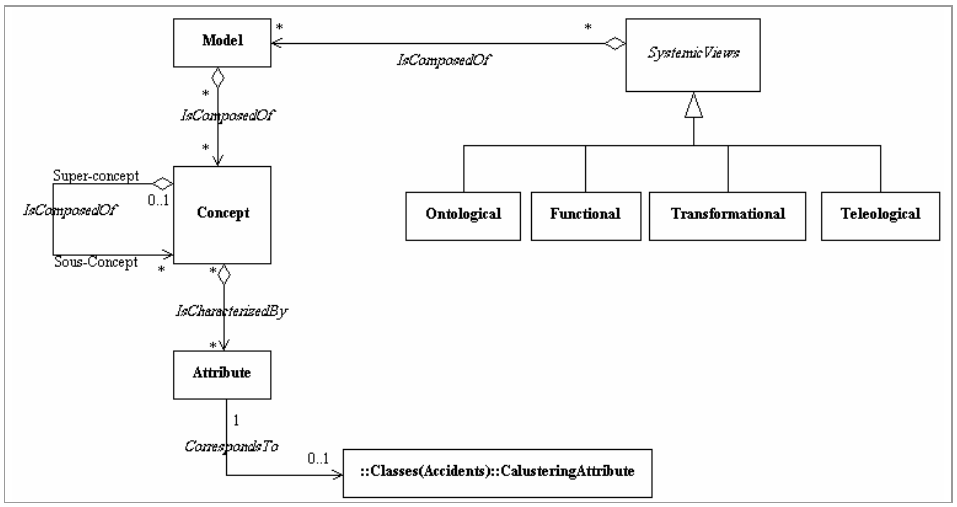


Figure 12 - The link between ASMEC and the clustering results: attributes in ASMEC correspond to attributes used in the clustering task.

Figure 13 shows the beginning of a table describing an accident cluster. Accidentologists and designers have to analyze each table using statistical features. Using our approach, we allow them to represent the same table (i.e. cluster) according to the different models (see Figure 14).

Clustering Attributes	Attribute modality	% of the modality in the study sample	% of the modality in the cluster
Crash_position	Offroad	26,64	96,72
Crash_Type	Rollover	21,76	78,69
Obstacle	Obstacle=ground	18,97	68,85
Number_Vehicles	Signle_Vehicle	29,15	72,13
Accid_situation	Control_Probl	32,50	73,77
Critic_task	Guidance_infrastr	15,62	44,26
Initial_event	External perturbation	5,72	22,95
Infrastructure_Typ	Straight line	24,83	49,18
Accid_Type	Pilotability	55,51	80,33
atmosphere conditions	Clear/Normal	55,79	80,33
Surface	Dry road surface	62,62	85,25
Accid. Position	Secondary road	47,98	70,49
Failure_Type	Action	9,07	22,95
Manœuvre	Lane_change_manoeuvre	6,14	18,03
Failure	failed_task	33,61	52,46
mask	No_Mask	65,13	81,97
critini	Perte_contrôle_tr_8#	17,85	32,79
failure_mecanism	Panic	5,72	14,75

Figure 13 - An example of an accident cluster.

CVE Model	Driver	Vehicle	Environment	Driv/Veh	Driv/Env	Env/Veh
	failed_task Lane_change_manoeuv external_perturbation typacc=pilotabilite driver_distraction Drug failure_mec=Panic failure_typ=Action	rollover Obstacl=ground	atmosph=clear/normal Dry_road_surf Secondary_road No_Mask driver_distraction straight_line	Control_Probl Dry_road_surf loss_lateral_control	Guidance_infrastr Dry_road_surf Lane_change_manoeuv	Obstacl=ground Dry_road_surf Single_Vehicle

Sequential Model	Permanent State	Normal Driving step	Failure step	Emergency step	Crash step
		atmosph=clear/normal Dry_road_surf Secondary_road straight_line	Control_Probl Guidance_infrastr No_Mask failed_task Single_Vehicle Lane_change_manoeuv driver_distraction	failure_mec=Panic	Obstacl=ground Dry_road_surf rollover Obstacl=ground Offroad

Inform. process. Model	Perception	Diagnostic	Prognostic	Decision	Action	Global
	No_Mask			failure_mec=Panic	fondef=Action failed_task loss_lateral_control	driver_distraction Drug/font>

Task Model	Navigation	Guidage latéral	Guidance longit.	Control latéral	longit. Control
	No_Mask	Guidance_infrastr straight_line	Guidance_infrastr	loss_lateral_control rollover	

Figure 14 - A multi-view projection of clusters.

Conclusion

Developing new safety systems requires the collaboration of designers and accidentologists. Brainstorming sessions are one of the means used in the LAB PSA Peugeot-Citroën and Renault to support the required collaboration. However, the various participants do not share the same viewpoint for accident analysis and understanding. Indeed, several models are used to analyze accident and this depends not only on the study objective, but also on the analyst specialty. A psychologist, for example, focuses more on the driver's information processing aspects whereas a designer is more interested in the mechanical aspects. This makes their communication hard and inefficient leading to a complex problem. Accident scenarios are one of the efficient tools allowing the required communication. However, even we use clustering techniques, the scenarios elaboration is time-consuming for experts. Moreover, these scenarios depend on the viewpoint of the expert performing them. Besides, they may be represented and interpreted according to several accident models that the various participants may used.

Using the *systemic (not systematic) approach*, we propose a multi-view architecture, which guides the user to identify the relevant models that may be used in accident analysis. It classifies the different models according to four viewpoints (ontological, functional, transformational and teleological). Then, we use an attribute-based approach to implement our approach. Concretely, we classify the attributes that characterize an accident according to the different concepts composing each identified relevant accident model. This allows us to represent automatically each accident scenario according to a specific model that users (accidentologists and/or designers) choose.

References

(Ashby, 1965) W. R. Ashby. An introduction to cybernetics, ed. Hall, C., London, 1965.

(Brenac, 1997) T. Brenac. *L'analyse séquentielle de l'accident de la route: comment la mettre en pratique dans les diagnostics de sécurité routière, Outil et méthode,* Rapport de recherche n°3, INRETS, 1997.

(Brenac and Fleury, 1999) T. Brenac and D. Fleury. Le concept de scénario type d'accident de la circulation et ses applications. *Recherche Transport Sécurité*, vol. 63, p. 63-77, 1999.

(Brenac and Megherbi, 1996) T. Brenac and B. Megherbi. Diagnostic de sécurité routière sur une ville : intérêt de l'analyse fine de procédures d'accidents tirées aléatoirement. *Recherche Transport Sécurité*, vol. 52, p. 59-71, 1996.

(Caroll, 1995) J. M. Caroll. Scenario-Based Design: Envisioning Work and Technology in System Development, John Wiley and Sons, New York, 1995.

(Caroll, 1998) J. M. Caroll. Scenario-Based Design. in *Helander M., Landauer T.K.,Prabhu P., Handbook of Human-Computer Interaction. 2nd edition, Ch. 17*, p., North-Holland, Amsterdam, 1998.

(Chovan *et al.*, 1994) J. D. Chovan, L. Tijerina, J. H. Everson, J. A. Pierowicz and D. L. Hendricks. *Examination of Intersection, Left Turn Across Path Crashes and Potential IVHS Countermeasures.,* Rapport N° : DOT HS 808 154, National Highway Traffic Safety Administration, 1994.

(Fleury *et al.*, 1991) D. Fleury, C. Fline and J. F. Peytavin. Diagnostic local de sécurité, outils et méthodes, Editions SETRA, Collection Etudes de sécurité, Bagneux, 1991.

(Fuller, 2000) R. Fuller. The Task-Capability Interface Model Of The Driving Process. *RTS, Recherche Transports Sécurité*, vol. 66, Tome 1, p. 35-45, 2000.

(Fuller and Santos, 2002) R. Fuller and J. A. Santos. Human Factors For High-way Engineers, Elsevier, Pergamon, 2002.

(Gandon and Dieng, 2001) F. Gandon and R. Dieng. Ontologie pour un système multi-agents dédié à une mémoire d'entreprise. *Ingénierie des Connaissances, IC'2001*, Grenoble, France, p 1-21, 2001.

(Jarke *et al.*, 1998) M. Jarke, T. Bui and J. M. Caroll. Scenario management: an interdisciplinary approach. *Requirements Engineering*, vol. 3(4), p. 155–173, 1998.

(Le Moigne, 1974) J. L. Le Moigne. La théorie du système général, P. U. F., Paris, 1974.

(Le Moigne, 1999) J.-L. Le Moigne. La modélisation des systèmes complexes, Dunod, 1999.

(Leite *et al.*, 2000) J. Leite, J. Doorn and K. GN. A Scenario Construction Process. *Requirements Engineering*, vol. 5, p. 38-61, 2000.

(Miller, 1995) J. G. Miller. Living Systems, University Press of Colorado, 1995.

(Najm *et al.*, 2001) W. G. Najm, J. D. Smith and D. L. Smith. *Analysis of Crossing Path Crashes,* Rapport N° DOT HS 809 423, National Highway Traffic Safety Administration, 2001.

(Page, 2002) Y. Page. *Elaboration de scénarios types d'accident pour le développement des systèmes de sécurité active embarqués dans les véhicules,* Rapport interne, LAB PSA Peugeot-Citroën Renault, 2002.

(Page *et al.*, 2004) Y. Page, R. Driscoll, J.-Y. Le Coz and T. Hermitte. Combination of statistical and case-by-case approach for accident situations classification. *FISITA*, Spain, 2004.

(Scheringer *et al.*, 2001) M. Scheringer, T. Vögl, J. Von Grote, B. Capaul, R. Schubert and K. Hungerbühler. Scenario-Based Risk Assessment of Multi-Use Chemicals: Application to Solvents. *Risk Analysis*, vol. 21(3), p. 481-497, 2001.

(Sohn and Lee, 2003) S. Y. Sohn and S. H. Lee. Data Fusion, Ensemble and Clustering to Improve the Classification Accuracy for the Severity of Road Traffic Accident in Korea. *Safety Science*, vol. 41(1), p. 1-14, 2003.

(Sohn and Shin, 2001) S. Y. Sohn and H. W. Shin. Pattern Recognition for Road Traffic Accident Severity in Korea. *Ergonomics*, vol. 44(1), p. 107-117, 2001.

(Summala, 2000) H. Summala. Automatization, automation, and modeling of driver's behavior. *RTS, Recherche Trans-ports Sécurité*, vol. 66, Tome 1, p. 35-45, 2000.

(Van Elslande and Alberton, 1997) P. Van Elslande and L. Alberton. *Scénarios-types de production de l'erreur humaine dans l'accident de la route, problématique et analyse qualitative,* Rapport de recherche N°218, INRETS, 1997.

(Von Foerster, 1995) H. Von Foerster. The Cybernetics of Cybernetics (2nd edition), Future Systems Inc., Minneapolis, 1995.

A Barrier-based approach to integrating human factors analysis into the analysis and design of complex systems

B. Schupp, P. Wright., M. Harrison*

Department of Computing Science, University of York, York, YO1 5DD.

*School of Informatics, University of Newcastle, Newcastle.

Abstract: Perrow pointed out complex systems can be characterised in terms of tightly coupled, non-linear interactions between diverse subsystems. As complexity increases risk analysis becomes difficult. Humans are an integral part of defence in depth, but can also be a source of error. In sociotechnical systems safety analysis is therefore further compounded by the difficulty of reliably specifying likely human performance with the system. Though extensive analysis and adaptation of the system later in design (for e.g. through prototyping and simulating interactive components) may lead to risk reduction at a late stage of system development, this can be expensive. A more cost-effective solution is to attempt to analyse systems safety earlier in the development process. What is required is an architectural approach to safety analysis. Our approach to safety architecture uses the Hazard Barrier Target (HBT) model to achieve risk reduction. The method facilitates analysts in identifying system elements or interactions which are hazardous to a certain target in the system its environment, and helps them reason about appropriate barriers to mitigate the risk caused by these hazards.

Adapting Interface Representations for Mobile Support in Interactive Safety Critical Contexts

Fabio Paternò, Carmen Santoro, David Touzet

ISTI-CNR Pisa, Italy.
http://giove.isti.cnr.it

Abstract: Mobile technology is penetrating many areas of human life. However, little attention has been paid so far to its use and impact in interactive safety critical contexts. We present a method that aims to provide designers with a better understanding of the introduction of mobile devices in such contexts and help them to identify and derive interfaces that support users in their activities. The method is a novel combination of a systematic analyses of potential deviations in task performance and information representations based on distributed cognition. The results of the conceptual design can drive model-based transformations able to identify and implement suitable interface representations. The originality of the contribution is in combining the results of a distributed task performance analysis with a transformation-based approach for generating user interfaces able to take into account such results.

Keywords: user interface, safety-critical systems, mobile devices

Introduction

In recent years there has been an increasing availability and use of a wide range of interactive devices, in particular mobile devices. This type of technology is penetrating many areas of human life. In interactive safety critical systems the introduction of new technology is often slow because people need to carefully understand their implications in terms of potential hazards. Only recently these issues have started to be addressed (Buisson and Yannick, 2001).

In this paper we present a novel method that aims to help designers in understanding such implications and finding the suitable representations that can be provided through mobile devices in order to improve usability while preserving safety.

Our method (analysis of distributed task performance) is based on the integration of systematic analysis of deviations and analysis of information representation based on the Distributed Cognition approach (Hitchins, 1995), (Hollan, Hutchins, Kirsh, 2000) (Fields *et al.*, 1998). In deviations analysis there is a systematic analysis of potential effects in case of *deviations* from the task plan. In order to help with such analysis, a number of deviation types (indicated by *guidewords*) have been identified. The approach is supported by task models, which are suitable to providing an overall view of the possible activities but may not be able to capture all the possible contextual aspects. This information can be provided through the support of Distributed Cognition analysis. It focuses on how knowledge is distributed across individuals, tools and artefacts in the considered context/environment. One basic point is that the breakdown in task performance is the consequence of inadequate access to the distributed representation of information resources supporting task performance. One limitation of this approach is the difficulty of translating its results into specific design criteria. The integration with the analysis of tasks and their performance can create the basis for addressing this issue. Once a better understanding of the tasks to accomplish and their information needs has been achieved, the third main element of our method comes into play: we apply a model-based approach (Paternò, 1999) (Wilson *et al.*, 1993) to the design of interfaces for a variety of devices, including mobile devices, taking into account the tasks to accomplish and the information regarding the context of use. In this process the idea is that the final representations provided should be suitable for the activities to support but can radically differ depending on the interaction resources available on the device at hand.

In the paper we describe and discuss the proposed method and show its application in a real safety-critical context, an Air Traffic Control Centre, for which we analyse the possible use of mobile interactive devices for supporting a specific role and some activities.

The Case Study

In order to show a potential application of the approach described and understand its feasibility, we considered a

real application domain, the Air Traffic Control, and we extracted a case study in the working environment of Rome-Ciampino control centre in Italy. In this control room, there is a number of en-route and approach working positions in charge of controlling respectively cruising flights and airplanes taking off/landing to the nearby major Fiumicino airport. In addition, there are other stakeholders (a chief controller, a technician supervisor, a flow controller), together with three or more supervisors having the responsibilities for making decisions about closing/opening sectors (usually in a vertical manner), depending on data about the estimated traffic load and airport capacity.
Air Traffic Management is based on the concept of airspace division into a number of sectors. The number of flights that are planned to cross a specific sector is called traffic demand. The maximum number that may be in a certain sector simultaneously (namely the number of flights that the sector itself is able to handle for each hour), is called traffic capacity and can be calculated using a number of parameters (types of flights, air complexity, etc.). If the number of flights that intend to cross a specific sector (traffic demand) is higher than the number of flights that the sector itself is able to handle (traffic capacity), a controller (flow controller) requires the emission of a flow or a so-called regulation. When a flight is subject to a regulation, a time slot in which a flight should take off is assigned, so as to distribute the traffic within a broader interval of time (while still maintaining the requests within the capacity of the sector).
More specifically, when the controllers in charge of the flow position recognise a situation in which the traffic demand exceeds the sector capacity, they coordinate with the supervisors and chief controller, and decide the actions to be taken (e.g.: some flights might be redirected through a different path with the same length, or a regulation might be triggered). In the current system, controllers use graphs visualised on paper-based documents such as that displayed in Figure 1, to analyse the variation in the traffic demand during (a part of) the day. As can be seen from the picture, the threshold line for the capacity of the concerned sector is 40 flights, which means that the sector is supposed to manage a maximum number of 40 flights per hour. However, there are some peak hours during the day in which the expected traffic might exceed such a limit: indeed in the time interval 11:00-15:00 it might occur more than once that the traffic demand is higher than the capacity; so appropriate action (e.g.: a regulation) should be triggered by the controllers.

In a situation of critical meteorological conditions, there is a controller in the centre, who *manages information about the upcoming traffic* in order to trigger de-combining of two sectors. Indeed s/he has the responsibility of making decision about closing/opening sectors (usually dynamically divided in a vertical manner), depending on data about the estimated traffic size and the airport capacity, and also personnel resources available on site.

The ATC supervisor can be regarded as the only role without any “dedicated” position within the control room. The need to have permanent access to real time traffic information may imply a high level of mobility in the control room.

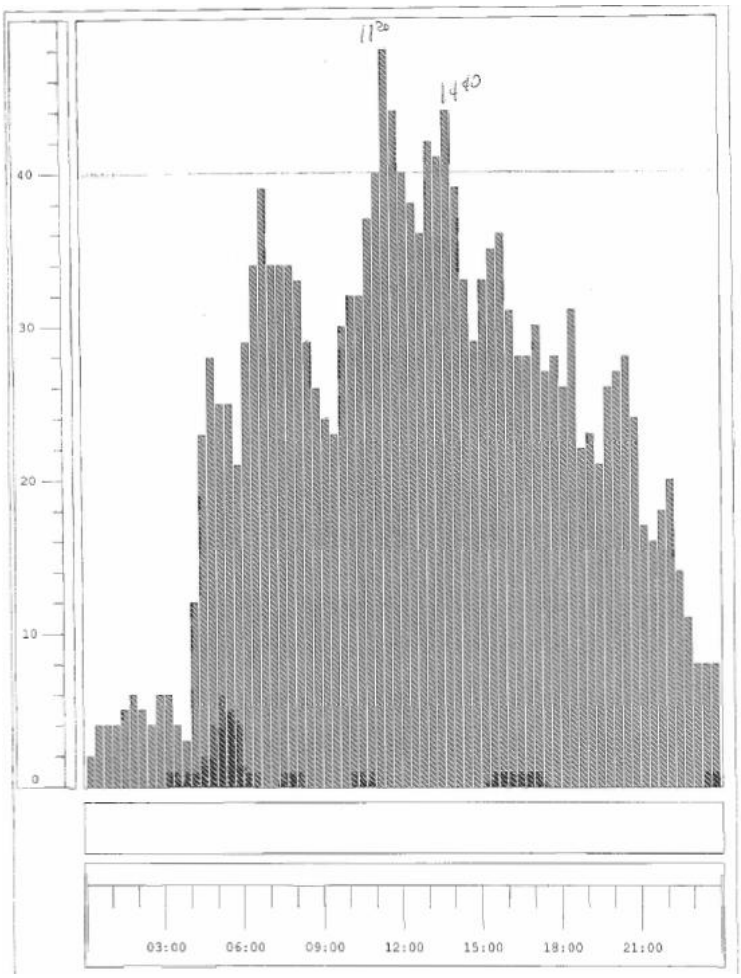


Figure 1 - A Working Tool for the Flow Controller in the Current Environment: Foreseen Traffic Displayed with Paper Bar Charts

In order to de-combine two air space sectors, the supervisor has to identify overloaded air traffic sectors, and the level of criticality of the upcoming air traffic; at the same time, he has to evaluate the on-site workload allocation of the controllers, and also to identify the personnel available to assume control of a new sector. The complex information supporting the supervisor's task is available from several sources distributed in the task space: flight information system, air traffic monitoring system, radar, flight progress strips, meteorological information, etc.

In this scenario the ATC controllers have to decide whether to open a new sector or not by checking the critical threshold of the upcoming air traffic level. In order to perform their activities, they use an integrated set of tools (computer displays, telephones, paper-based documentation). In addition, they need real time access to information about the current and estimated up-coming air traffic levels, as the decision to open a new sector will be further based on the information manipulated by this task.

The Methodological Approach

The basic idea is to evaluate a current design in order to analyse and identify possible areas for improvement, in terms of some criteria which should also depend on the specific application considered. For instance, if we consider a safety-critical application, a possible analysis should perform risk assessment in the current design, namely understanding and quantifying the level of risk involved in the considered design in order to evaluate if it is acceptable/tolerable or new actions should be provided to manage hazardous situations that might occur. As soon as such areas of improvements have been identified, the next step is to identify and specify a new arrangement/distribution of activities, roles, artefacts and devices in order to compensate for the identified shortcomings. Indeed, such specification should describe how the activities are supposed to be carried out in the new system, also specifying how they evolve over time, the context in which they are supposed to be performed, the roles that are expected to act, and the artefacts and/or devices available in the new setting.

In the analysis of potential improvements it is also possible to consider the impact of new technology, such as mobile devices. Once a new model of how activities should be organized has been identified in such a way as to address the issues of the previous solution, it can be captured in a specification that can be used to derive a new user interface able to take into account the new requirements. This can be obtained with the support of model-based design and development tools able to consider the logical descriptions of the activities to perform and suggest and generate user interfaces suitable for their support, even considering the possible interaction platforms.

Such process can then be iterated because, once a prototype of the new user interface is produced, it might in turn be subject to evaluation (according to the analysis of distributed task performance) and, as a result of such analysis, further changes to the specification might be included, so as to restart the process.

Analysis of Deviations

In this section we provide a more detailed description of the method proposed. One starting point for this research has been the deviation analysis (Paternò and Santoro, 2002). In executing the analysis, several aspects need to be carefully identified: role, task, representations, and deviations.

1. Analysis of the task and related properties;
2. Analysis of the representations associated with the task
3. Analysing deviations from the task plan

The results of such analysis may be stored in a table with the following information:

- *Task*: the activity currently analysed, together with some properties relevant to our analysis;
- *Representation distribution*: the resources supporting task performance and their distribution;
- *Guideword*: the type of interaction failure considered;
- *Explanation*: how an interaction failure has been interpreted for that task and the given deviation;
- *Causes*: the potential causes for the interaction failure considered and which configuration of resources might have generated the problem;
- *Consequences*: the possible effects of the interaction failure in the system;
- *Recommendation*: suggestions for an alternative (if any) distribution of resources able to better cope with the considered interaction failure.

The deviations analysis can be applied to all possible activities, even if the detailed consideration of causes, consequences and recommendations makes particularly sense in the case of safety critical systems. By way of example, we can consider a simple activity such as printing and a deviation type such as None. This means that the printing has not effect and we have to understand why this happens. This could be for several reasons: no input information (the user did not correctly select the file to print) or the activity is not performed (the printer is broken) or the activity is performed but does not generate the result (the printer is out of paper).
If potential safety-critical issues have been identified then our analysis aims at identifying better representations (or distributions of them) that could be more suitable for carrying out the considered tasks and prevent occurrences which can have safety-critical effects. The evaluation has to consider if a different allocation of resources may be envisaged, which implies different representations of information and could involve considering different devices, that may result in a significant improvement for the overall system's safety and usability.

Analysis of Information Representation

In this part of the method the focus is on the analysis of the information representations in order to understand whether they are optimal for the distributed task performance. To this end, a number of important attributes have been identified:

Externalisation: to what extent the representation is explicitly provided in the real world or is based on implicit, internal representations..

Accessibility: whether it can be problematic for the user to access the information because it is difficult to see or to hear it or for any other reason.

Access modality: the type of access that a user has to a representation. Different types of access (sequential/concurrent) could be exploited for externally available representations.

Mobility: whether the access to the information requires the user to move or the user can move while accessing information.

Sharing: refers to the extent the perception of a representation is: i) Local to individuals; ii) Shared (e.g. by the members of a team); iii) Globally available to all. This property might be connected with the type of supporting platform (for example if controllers annotate a strip on their PDAs, this information will be available locally to them).

Persistence: whether transient or permanent access to information is allowed.

Flexibility of modifying the representation, ability to flexibly update and modify the representation, for example allowing a person to annotate an external representation (i.e. strips).

Operations and actions supported such as:

Comparability: with other objects / representations available in the user's context;
Combinability: allowing users to combine information from different sources;
Ease of production: allowing reconfiguring and multiple views of information;

If this type of analysis is applied to our case study we can notice that several information objects supporting task performance may be identified: normal and critical threshold of upcoming air traffic level, additional parameters such as estimated numbers of aircraft, together with time intervals, planned trajectories, etc. In particular, such thresholds of upcoming air traffic level are visualised on paper in the form of a bar chart, such as that visualised in Figure 1. As for the properties associated to the different resources, various properties have been identified:

Externalisation: they are available in both graphical and numerical representation forms.

Accessibility and Access Modality: the representation forms and media (in this case, large computer screens) allow users concurrent, easy access to a variety of information. If accessing the same information with a PDA, it is expected that its physical constraints (i.e. screen size) will make sequential the access to information, therefore increasing the time and effort needed to visualise the same items. On the other hand, a PDA would allow permanent access to the required information, even if user changes his position across the control room.

Mobility: The user is expected to move about the control room to access the needed information.

Sharing: similarly, using a small screen device is likely to change the observability of information, from being easily *shared* with other members of the team, to *locally* available to the user of the device.

Persistence: critical information (e.g.: threshold of the upcoming air traffic level) is graphically represented, thereby allowing non-transient access.

Flexibility of cognitive tracing and interactivity. In the considered case, the parameters of interest are changing *autonomously* according to the real-time situation of the air traffic flow. Controllers have no or minimal permission to effect changes, to annotate or update an external electronic representation. A standard working position is equipped with no input device (i.e., a keyboard), as only direct manipulation of the objects already available on the screen is allowed.

Operations and actions supported

Comparability. the graphical representation of information employed (i.e., clustered columns) provides users with the possibility to directly *compare* various values of the monitored variables (i.e., by rapidly perceiving differences between the height of two columns).

Combinability - possibility to combine and reconfigure or re-represent the information of interest: well supported in the current ATC work settings. For instance, the information contained in an electronic flight strip can be displayed in two different formats; the values of the upcoming traffic may be represented graphically as well as numerically, etc. For the hypothetical situation of using a PDA, the question is how to

effectively display the relevant information in the perimeter of a very small screen space, while maintaining a high level of interactivity. For instance, a solution would be to reduce the amount of graphics, rely mainly on the numerical representation of information, and using additional codes (e.g. sounds) in order to facilitate user's rapid discrimination of critical information.

Analysing Deviations from the Task Plan

It is possible to apply the deviation analysis to the case study to identify potential safety-critical issues. For instance, if we consider the possibility that the representation of interest is not available, there are some possible causes that might be identified. For instance, *the information is not visible* and the possible causes are that there are some difficulties in perceiving the relevant information due to some usability issues (the object represented is too small, ambiguous shape, wrong choice of colour, etc.; but also supervisor' s distraction / interruption by other activities, etc.). In other cases, *the representation might not be persistent*, due to rapid change of information values, which does not give the user the necessary time to internalise the perceived information and to integrate it with the other information supporting his decision making.

As for possible consequences, if there is no information available in the task performance space (not visible, not persistent) then various types of task failure can occur (e.g. stop task performance, delay, etc.). Then, as a result of this analysis, a possible r*ecommendation* might be to rely on multiple ways of representing the same information (e.g., visual and auditory): access to concurrent representation of the same information could be especially important for users 'on the move', who allocate their attention to several competing tasks. Design should facilitate a rapid perception of the relevant information, and support an accurate interpretation of its significance (e.g., estimation of the air traffic flow - under/ over a critical level, or its approximate value). For instance, discrimination of the critical information could be facilitated by suitable use of colour, use of multimedia facilities such as animation, blinking images, the use of sound, etc.

Redesigning the Work Model

As the first step of the analysis has highlighted the need for the controller to have information available while moving round in the control room, then, a new specification of the activities that should be carried out is to be described, in order to solve the problem that has been identified. In the new, envisioned system, the activities should be carried out so as to allow controllers constant access to the information that is needed to perform their activities. The new specification should identify the new context in which the activities are carried out, and the new arrangements of resources/devices.

More specifically, the envisioned system calls for providing the controller with a mobile device to display the critical information. The controller needs to access such information in real time, so it should be always available, and, to this end, the possibility that such information should be visualised on a PDA might be envisaged in the new system. Then, in this case, due to the limited capabilities of the handheld device, only a selected subset of the data normally displayed by the current system tools should be visualised on such devices. In addition, due to the new context of information displayed on the PDA (the user is supposed to be mobile), specific presentation techniques able to cope with the rather noisy environment of the control room should be foreseen and the eventuality that the controller not watch the device constantly be adequately controlled for.

In addition, special attention should be paid on how to render some specific types of representation (like those used in the ATC case study considered, eg bar charts and line charts) in order to understand the best way to render such data on the small screens available on handheld devices such as PDAs.

In our new approach, once the activities have been specified, together with information about the new arrangements of tasks and resources, such specification represents the input from which a new design of the user interface can be derived. This information should be specified so as to address the characteristics of the problem in a top down manner, from the abstract level and then refining to more concrete levels. Indeed, the process allowing to derive the user interface from a high level description of how the activities are supposed to be carried out in the new environment includes three basic steps, that will be described in more details in the following subsection.

Linking Design and Development

The method for generating the user interface is based on three main transformations. Such transformations have been implemented in a tool, TERESA, which is a semi-automatic environment supporting a number of

transformations useful for designers to build logical descriptions and exploit the information that they contain to consequently generate the user interface for various types of platform.
The main abstraction levels considered are: the task model, where the logical activities to support are identified and the abstract user interface, a logical description of the user interface. They are used to obtain a user interface implementation able to support effectively the tasks identified. Then, there is a concrete level, which is useful to link the abstractions and the implementation. The main transformations considered are:
- *From Task Model-related Information to the Abstract User Interface*. The task model specification, along with information regarding groups of tasks that are enabled over the same period of time, are the input for the transformation generating the associated abstract user interface, which will be described in terms of both its static structure (the presentation part) and dynamic behaviour (the dialogue part). The structure of the presentation is defined by logical interaction objects (interactors) characterized in terms of the basic tasks that they support, and their composition operators. Such operators aim to structure logically the presentation according to the communication effects desired. They are grouping, which indicates a set of interface elements logically connected to each other; relation, highlighting a one-to-many relation among some elements, one element has some effects on a set of elements; ordering, which indicates that some kind of ordering among a set of elements can be highlighted, and hierarchy, which is used when different levels of importance can be defined among a set of elements.
- *From the Abstract User Interface to the Concrete User Interface*. This transformation starts with the abstract user interface; it is possible to move into the related concrete user interface for the selected specific interaction platform. The difference between these two levels is that the abstract description is modality and platform independent whereas the concrete description is platform dependent. Thus, for example at the abstract level the designer can specify that there is a need for a selection object whereas at the concrete level the specific interaction technique is indicated (for example, a radio-button or a list or a vocal selection). It is worth pointing out that the concrete level is still a logical description and is independent from the specific implementation language or device which is going to be used. Indeed, the platform is a characterization of a group of devices that share similar interaction resources (such as the desktop, the vocal device, the PDA and so on). A number of parameters related to the customization of the concrete user interface are made available to the designer in order to obtain the concrete interface, with different levels of intervention required from the designer, ranging from completely automatic solutions to other cases in which the designer might modify all the possible details in the design process. The tool can provide suggestions according to predefined design criteria, but developers can modify them. In addition, depending on the type of platform considered there are different ways to implement design choices at the user interface level.
- *From the Concrete User Interface to the Final User Interface*. This last step generates the interface code according to the type of platform selected from the concrete user interface. Before generating the final code, it is possible to specify within the tool the value of additional parameters allowing the designers to still diversify between the various devices belonging to the same platform. For instance, if we select to generate the final code for a mobile platform, depending on the features of the current device, a different final user interface will be produced by the tool e.g. in XHTML, XHTML Mobile Profile, SVG and VoiceXML.

Rendering Interactive Graphical Representations
In order to better support the results of a distributed task performance analysis, a new version of the TERESA environment (Mori, Paternò, Santoro, 2004) has been designed. The main new features consist in the possibility of generating interactive graphical representations (implemented in SVG) that adapt to the feature of the devices considered, including mobile devices, following different strategies. This means that a novel transformation from the abstract user interface level and the implementation has been designed for this purpose.
The abstract user interface is generated from the task model. This transformation is based on the information associated with each user task and the temporal and semantic relation among them. Thus, abstract structured objects compose elementary objects according to some logical information. For instance, at the abstract level structured tables are considered as structured objects composed of lists of basic elements (numerical, textual elements…), so they are modelled as a set of ordered lists characterized by the type of data that they contain, and associated each other by means of some abstract relations that model the semantic link existing between them. The various ways in which such relations occur at the abstract level (we call them abstract operators) are translated, at the concrete level, into appropriate techniques able to convey the related meaning also at this level. For example, concrete chart encoding like pie chart, bar charts, etc are all examples of concretely translating abstract combinations of list of ordered elements (it is an example of application of what we called the *objectOrdering* abstract operator).

Concrete user interfaces are basically computed based on the structure of abstract user interfaces together with

additional information useful for selecting the most appropriate concrete rendering technique, depending on the platform that will be used to render the user interface. Indeed, moving from the abstract to the concrete level first requires the designer to select a target device type among those supported by the environment.

These concrete models associate each structured object with a specific concrete chart encoding (bar chart, line chart…) of data. These techniques have to be consistent with the previously declared abstract types. Among the relevant representations that are possible (for example bar charts, line charts, and scatter plots), if some may not be rendered correctly for any reason (such as insufficient colour or interactivity support) on the target platform then they are not considered. According to the selected chart type, designers should be able to specify a number of additional information in order to customise the presentation, so as to fine-tune better the layout of the chart that will be generated. For instance, with charts for desktop environments, the designer can decide whether numerical values have to be displayed on the chart in addition of their graphical encoding.

In our case, and with the specificity of the case study considered, the constraints inherent to small screen enabled devices are addressed by the generation of interactive structured graphics embedding dynamic exploration functionalities. By exploiting some information visualization techniques, the interactive exploration facilities aim to enable users to efficiently access information encoded by structured graphics on small displays.

As interactive exploration facilities depend on the abstract structure of the considered object, a technique that at the abstract level allows the designer to specify the relative importance of basic elements (we called it *objectHierarchy* abstract operator) will be associated, at the concrete level, to an interaction technique able to highlight differently the information depending on the available capabilities of the device (for instance through the use of techniques such as semantic zooming). In addition, an *objectHierarchy* operator at the abstract level can be also rendered through the use of fisheye view exploration mechanism, able to highlight the most important information to be presented depending on the Degree Of Interest dimension.

Analysis of the Resulting Representations in the Example

The application of our approach to the case study can find a more efficient manner of showing data needed to the ATC controller in a PDA-enabled new system, as a consequence of ensuring a greater level of safety in situations such as that highlighted by the deviation-based analysis previously performed, in which the controller might be temporarily unaware (because, e.g. is distant) of some critical information currently visualised on some tools.
In the new, envisaged system, the mobile device enables the controllers to move around the control centre bringing with them the device so as to get a full, continuous awareness of the expected situation. In addition, thanks to the fisheye view-equipped graphs the controllers have on their PDAs, it is possible for them to more properly focus on the current area of interest, which are the intervals of time when the threshold limit are likely to be overcome. Moreover, there is the possibility for the controller to have additional information on specific time intervals, by tapping-and-holding the pen stylus on some specific bars, so as to have visualised more precise information on the concerned values. For instance, in Figure 2, the controllers have currently focused their interest on the period of time between 12:40 and 13, and a tooltip is displayed in order to more precisely show that the number of flights that are expected for that period of time is 42, which is beyond the supposed capacity of the sector.

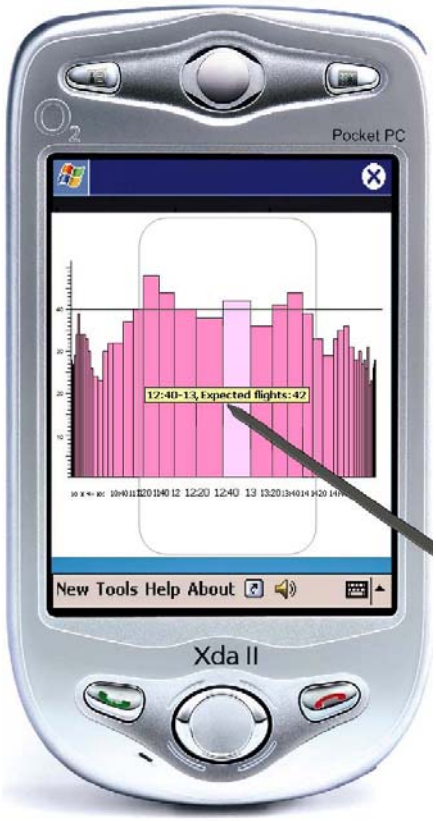


Figure 2 - The Fisheye Viewed Bar-Chart with Flight Information for PDA.

The resulting presentation is able to address some issues highlighted by the deviation-based analysis performed in the first stages of our approach. First of all, the introduction of a mobile device in the new environment allows the controllers to get anytime the necessary information for performing their work, so compensating to the lack highlighted when the controller is far from the stationary workstations and then the concerned information is out of reach. In addition, the use of effective information visualisation techniques on handheld devices has made the rendering of such critical information on a small screen of a PDA effective, so maintaining a high level of usability even for users which are mobile. As it is possible to note from Figure 2, the precision requested for selecting the different areas of the graphs does not pose strict constraints to users on the go, as it is a fairly easy task selecting a bas on the bar chart even for a mobile controller. In addition, the most critical information is visualised in various redundant ways: it is not only displayed on the graph, but also emphasised by the fisheye view, and further displayed in a dedicated tool-tip activated on the window.

Conclusions

In this paper we have presented a method composed of two main phases: a distributed task performance analysis, which aims to identify potential safety-critical issues through the analysis of deviations from the task plan and the information necessary for its accomplishment; and a transformation-based tool able to take the result of an envisioned conceptual model and obtain interfaces effective for the activities to support. One key advantage of this method is the possibility to support design in safety-critical contexts when the introduction of new mobile technology is considered. This result is obtained because the analysis is able to consider the context and how it can affect the user interaction and the tool is able to generate interfaces that are able to adapt to the feature of the devices considered.

In this way the environment supports the work of multi-disciplinary groups where the result of the conceptual design can be used to actually support the development phase.

Future work will be dedicated to extending the multi-modal aspects of the interfaces generated by the tool.

References

Buisson, M., Yannick, J. (2001). *Design Issues in Distributed Interaction Supporting Tools: Mobile Devices in an ATC Working Position*, Proceedings of Mobile HCI 2001.

Fields, R.E., Wright, P.C., Marti, P. & Palmonari, M. (1998). *Air Traffic Control as a Distributed Cognitive System: a Study of External Representation*. Proceedings of the 9th European Conference on Cognitive Ergonomics - ECCE-9, EACE Press.

Fields, R., Paternò, F., Santoro, C., Tahmassebi, S.(1999). *A Method for the Comparison of Design Options for Allocating Communication Media in a Cooperative and Safety-Critical Context*. ACM Transactions in Computer-Human Interaction Vol.6, N.4, pp.370-398, ACM Press, December 1999.

Hollan, J., Hutchins, E. & Kirsch, D. (2000). *Distributed Cognition: Toward a New Foundation for Human-Computer Interaction Research*, in ACM Transactions on Computer-Human Interaction, 7 (2), p. 174-196.

Hutchins, E. (1995) How a Cockpit Remembers Its Speeds. Cognitive Science, 19, pp. 265-288.

Mori, G., Paternò, F., Santoro, C. (2004). *Design and Development of Multi-Device User Interfaces through Multiple Logical Descriptions*, IEEE Transactions on Software Engineering, August 2004, Vol.30, N.8, pp.507-520, IEEE Press.

Paternò, F, Santoro, C. (2002). *Preventing User Errors by Systematic Analysis of Deviations from the System Task Model*. International Journal Human-Computer Studies, Elsevier Science, Vol.56, N.2, pp. 225-245.

Paternò, F. (1999). Model-based Design and Evaluation of Interactive Applications, Springer Verlag, ISBN 1-85233-155-0, 1999.

Wilson, S., Johnson, P., Kelly, C., Cunningham, J. and Markopoulos, P. (1993). *Beyond Hacking: A Model-based Approach to User Interface Design*. Proceedings of HCI'93. pp.40-48, Cambridge University Press, 1993.

Viewpoints and Views in Engineering Change Management

René Keller, Claudia M. Eckert, P. John Clarkson

Engineering Design Centre, University of Cambridge
Trumpington Street, Cambridge, CB3 9BB, United Kingdom
http://www-edc.eng.cam.ac.uk
Email: rk313@cam.ac.uk, cme26@cam.ac.uk, pjc10@cam.ac.uk

Abstract: Visualising connectivity and change propagation in complex products is difficult, but nevertheless is a key for successful design. Several stakeholders, such as designers, managers and customers have different viewpoints on the designed artefact and require different information. Multiple views provide a means to visualise complex information and are also a way to fulfil the demands of different user groups. In this paper we introduce the concepts of multiple viewpoints and multiple views in engineering design and show how multiple views are integrated into a software tool for predicting change propagation.

Keywords: Change Management, Complex Products, Visualisation.

Introduction

Change is an essential part of all design projects. Most products are designed by modifying others. Changes to the existing state of the design can occur at any stage in the design process. When one part is changed, other parts can be affected through the links that exist between them. Component parts are linked by different types of relationships, such as mechanical, spatial, thermal or electrical links, which often correspond to the fields to expertise contributing to the design of the product. Knock-on changes to other components can be unwanted and very costly. Predicting such changes accurately can thus be the key to risk-assessment.

In one example, an engine company missed an electrical link between adjacent components when a metal pipe was replaced with a cheaper plastic one. They found that the engine did not work, because one part was no longer earthed. Indirect links between non-adjacent components can be even more problematic to spot. When a helicopter is customised for example, additional systems are often mounted to the outside of the craft. If their weight exceeds the strain margin of the fuselage, the fuselage has to be reinforced, which leads to many other costly changes.

Current change prediction methods depend primarily on the experience of engineers, however in complex products, designers are likely to overlook connections and miss potential change paths. The CPM (Change Prediction Method) tool is a software tool that aids in the analysis of change propagation. A number of case studies (Jarratt et al. 2004) including a gas-turbine company and a diesel engine manufacturer showed its industrial applicability. As the industrial success of such a change propagation tool highly depends on finding a way to present all the desired information visually so that the user (in this case the designer) is not overwhelmed by the amount of information, the primary focus is on the development of appropriate human-computer interfaces.

In this paper we introduce two concepts. One is the existence of multiple viewpoints in the design process of complex products, due to different stakeholders. The visualisation needs of a project manager are different from that of a designer responsible for the design of a special component and potential customers again demand very different views on the product. As a consequence of these different viewpoints and due to the complexity of the design artefact, we propose the use of different views to visualise design information in effectively. The CPM tool will serve as an example of how such a strategy of multiple views can be integrated into the design process.

Complexity in Design

Designing a product is complex in many ways. Earl et al. (2004) identified four layers in which complexity in design can occur. First, the product itself is complex as it might have many components that are highly interrelated and linked in various ways. Second, the process of designing the product can consist of many interlinked tasks with probabilistic outcomes that can cause costly iteration. Third, the organisation that designs the product can be considered complex, as it consists of a large number of multidisciplinary teams that are involved in the design. Fourth, the relation of the product to its environment can be complex. In this paper we will focus on two facets of

Field Code Changed

designing a complex product, the existence of multiple viewpoints and the need for multiple views for visualising a complex product.

Multiple Viewpoints: In a company designing even simple products, many people are involved in the design process. Each designer comes form a different object world (Bucciarelli 1996) and has a different background and task focus and sees the product in a different way. Design managers also operate on a different level of abstraction from designers engaged in detailed design. As described in Eckert et al. (2004), even chief designers of a helicopter manufacturer admitted that they only understood roughly half of a helicopter in any degree of detail (see Figure 1 right). Other designers have an even more biased view on the product. They know only about their own task and those of the people they directly interact with. For example, a mechanical engineer might know a lot about stress engineering, but only have a vague understanding of avionics.

In the famous caricature by Saul Steinberg, the "*View of the World From 9th Avenue*" (see Figure 1 left) a similar concept is shown. In this drawing, Manhattan is shown in very high detail, including single streets and buildings, while the rest of the USA (everything beyond the Hudson River) is reduced to landmarks like large cities (Chicago) or geographical objects (Rocky Mountains). The view of the world from any other city would look quite different. The idea of multiple viewpoints is best represented in information visualisation as fisheye views (Furnas 1986). The concept behind this theory is that everyone is mainly interested in the part of the environment directly surrounding them. The further away something is, the less attention and interest is spent. When visualising information with fisheye views, the user is able to set a viewpoint. The screen space is then assigned to the objects based on a "Degree of Interest", assigning less space to less interesting ones.

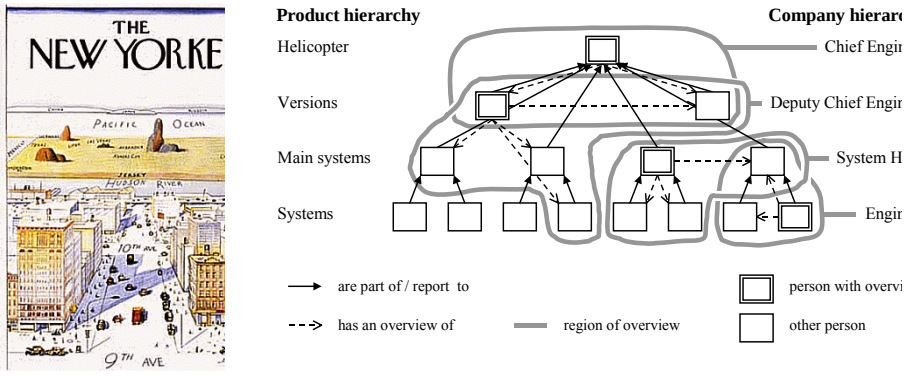


Figure 1 - Two concepts of different viewpoints: "View of the World From 9th Avenue" (left); Overview over a helicopter (right) (Eckert et al. 2004)

Multiple Views: Multiple views are widely and successfully used for the visualisation of complex information (Unwin 1999), in software engineering (Meyers and Reiss 1992) and even in an engineering design context (Packham and Denham 2003). In the case of complex products, we argue that there are two reasons for using multiple views:

- The amount of information in a complex product is too large to be displayed in one single graph. The information has to be broken down into smaller chunks that can be visualised and analysed much easier. Different graphs can show different information, revealing structure that cannot be shown in one diagram. Some representations are good for some purposes and not for others. A network diagram for instance is a very capable representation for most relational data. However, very dense graphs cause the problem of edge-

crossings. For representing very dense graphs, the compact form of a Design Structure Matrix (DSM) is a much better display (Ghoniem et al. 2004). DSMs on the other hand do not show the structure of the network in an intuitive way, especially when indirect connections between components have to be assessed.

- Different people involved in the design process have different viewpoints and demand different views on the product data. Potential customers demand different information than the designer responsible for the design of the Cylinder Head of an engine. For example in one case, a designer demanded the capability to 'fade' out all but one linkage type in a product model, as he wanted to see only the linkage type that had the biggest impact on his design. This concept is best described as overview.

Tailored displays that are able to adapt the viewpoint of the particular user could be highly beneficial. However, hardly any tools exist in current design practice that offers such functionality.

CPM Tool

The CPM tool is a software tool developed at the Engineering Design Centre in Cambridge (Clarkson et al. 2004). The core representation is the Design Structure Matrix (Browning 2001). The DSM interfaces have been refined using feedback from designers in two leading UK engineering companies. This *User Centered Design* approach (Brown 1996) of interviewing potential users to gather functional requirements promotes the development of optimised interfaces and visualisation techniques for the CPM tool. The interviews showed that designers are overwhelmed by the amount of information provided by a DSM. For efficient decision-making, they required a balance between detailed information and a global overview. The software tool supports risk assessment by drawing designers' attention to components that are highly connected to other components and where changing any of these components would result in major rework on other, not necessarily directly connected components. We argue that proper visual representations and interfaces are the key for industrial acceptance of this software tool (for the impact of human-computer interfaces on design see Ligetti et al. (2003).

Currently the CPM tool supports the design change process in two different ways. On the one hand it supports abstract product-model building. Systematically populating the corresponding product model in a multidisciplinary team increases awareness of the participants. It helps the individual designers as well as team leaders to understand how the components in their field of responsibility are connected to other parts of the products and where possible interfaces with other teams exist. With the software tool, information such as different linkage types and direct change impact and likelihood can be captured. The model building was carried out successfully in two UK companies, a diesel engine manufacturer and a gas turbine company.

The second benefit of the CPM tool is that it provides a platform to analyse change propagation data, based on combined component connections. For that purpose, algorithms for calculating combined risk from direct impact and likelihood values were developed and integrated into the tool (Clarkson et al. 2004). This allows designers to quickly assess the probability of change propagating from one component to other components as well as the overall risk associated with a component change. The visualisation techniques described in this paper are designed to support this second case.

Visualising Change Propagation

The CPM tool incorporates a number of interactive linked views for visualising different facets of change propagation data. These include:

- Matrix-based visualisations such as the Combined Risk Plot (see Figure 2, left) for assessing direct linkages and combined change-risks as well as building product linkage models;
- Network displays for showing both direct and indirect links between components (see Figure 2, right);
- Tree-based diagrams for showing propagation paths resulting in a change from one component;
- Mechanisms to support sensitivity analyses of the product models, design freezes and component hierarchies.

For a more detailed description of the different displays offered by the CPM tool and scenarios that are supported, see Jarratt et al. (2004) and Keller et al. (2005).

These visualisation techniques support the designer in different stages of the design process, ranging from model building to the analysis of the data stored in such a model. Currently, the CPM tool is mainly used in industry for building product models.

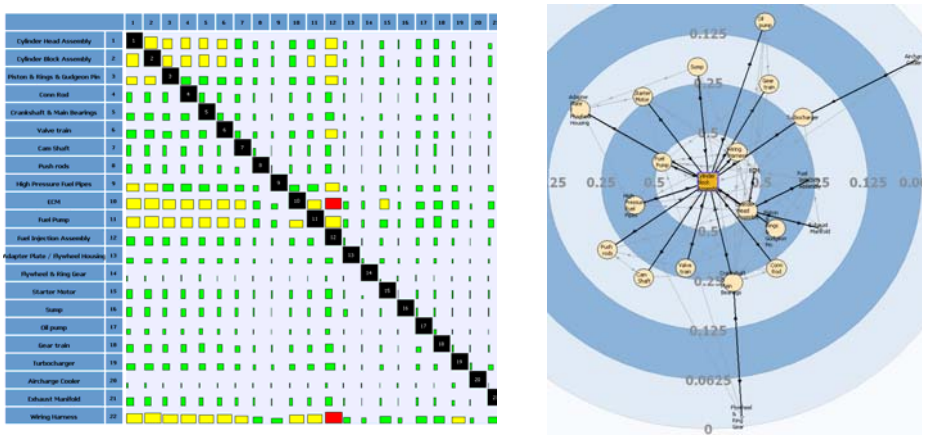


Figure 2 – A Combined Risk plot shows combined change risks (left), a change network visualises change propagation paths (right).

Usability Testing

In order to ensure that the CPM tool is accepted in industry, the usability of the interfaces that are part of the CPM tool must be verified. In order to improve the usability of the CPM tool, we follow two approaches: One is a *User Centered Design* approach (Brown 1996) with close cooperation with our industrial partners. This involves interviews with potential users to gather requirements, group meetings where the current state of the software is presented and sessions where the tool is used in "real world" scenarios.

The second approach is to do in-house testing of the visualisations. This includes controlled experiments that compare user-performance using different representations for a certain task. We especially focus on the differences between the two main representations incorporated into the software: DSMs and network-based displays. A comparison between these two representations used for model building revealed that the differences between both representations are only marginal; participants in the study assessed more links with DSMs but needed more time. However, we found that certain users have strong preferences. One experienced designer who participated in the study for instance mentioned: "*Lets face it, a DSM is not a representation designers like using*".

Future work will reveal whether one visual representation is best for analysing complex change propagation data. In a similar study, Ghoniem et al. (2004) discovered that matrix-based techniques are more suitable for showing relational data than networks, especially if the networks are very dense. We expect similar results for product networks, which tend to be very dense.

These findings will be incorporated into the software tool so that the best possible visual representation is available for designers analysing change propagation data of complex products.

Conclusion

In this paper we introduced the concept of multiple viewpoints in the design of complex products. These are common in large and multidisciplinary design teams. Additionally, designers are not interested in parts of the product that have little or no impact on their area of responsibility. They demand a view on the product that is tailor-made and does not show an overwhelming amount of unnecessary information.

The complexity of the underlying product is another reason why traditional means to visualise complex products are not sufficient. We introduced multiple and fisheye views as ways to tackle this problem and showed how the CPM tool incorporates such a strategy for visualising change propagation data. Finally, we showed how we ensure that the software is usable even for displaying complex products.

References

Brown, J. (1996). Methodologies for the Creation of Interactive Software. Technical Report CS-TR-96/1. Wellington, New Zealand, Department of Computer Science, Victoria University of Wellington.

Browning, T. R. (2001). "Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and new Directions." IEEE Transactions on Engineering Management **48**(3): 292-306.

Bucciarelli, L. L. (1996). Designing Engineers. Cambridge, MIT Press.

Clarkson, P. J., C. Simons and C. M. Eckert (2004). "Predicting Change Propagation in Complex Design." ASME Journal of Mechanical Design **126**(5): 765-797.

Earl, C., J. Johnson and C. M. Eckert (2004). Complexity. Design Process Improvement. P. J. Clarkson and C. M. Eckert. London, Springer Verlag.

Eckert, C. M., P. J. Clarkson and W. Zanker (2004). "Change and customisation in complex engineering domains." Research in Engineering Design **15**(1): 1-21.

Furnas, G. W. (1986). Generalized Fisheye Views. Proceedings of CHI '86, Boston, Massachusetts, USA.

Ghoniem, M., J.-D. Fekete and P. Castagliola (2004). A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations. Proceedings of InfoVis 2004, Austin, Texas, USA.

Jarratt, T., C. M. Eckert and P. J. Clarkson (2004). Development of a Product Model to Support Engineering Change Management. Proceedings of the TCME 2004, Lausanne, Switzerland.

Jarratt, T., R. Keller, S. Nair, C. M. Eckert and P. J. Clarkson (2004). Visualization Techniques for Product Change and Product Modelling in Complex Design. Diagrammatic Representation and Inference. A. F. Blackwell, K. Marriott and A. Shimojima**:** 388-391.

Keller, R., T. Eger, C. M. Eckert and P. J. Clarkson (2005, accepted for publication). Visualising Change Propagation. ICED '05. Melbourne, Australia.

Ligetti, C., T. W. Simpson, M. Frecker, R. R. Barton and G. Stump (2003). "Assessing the Impact of Graphical Interfaces on Design Efficiency and Effectiveness." ASME Journal of Computing and Information Science in Engineering **3**(2): 144-154.

Meyers, S. and S. P. Reiss (1992). "An Empirical Study of Multiple-View Software Development." ACM SIGSOFT Software Engineering Notes **17**(5): 47-57.

Packham, I. S. J. and S. L. Denham (2003). Visualisation Methods for Supporting the Exploration of High Dimensional Problem Spaces in Engineering Design. International Conference on Coordinated Multiple Views in Exploratory Visualisation, London, UK, IEEE Computer Society.

Unwin, A. R. (1999). "Requirements for Interactive Graphics Software for Exploratory Data Analysis." Computational Statistics **14**: 7-22.

Applying Task Analysis to Facilitate the Design of Context-Aware Technologies

Yun-Maw Cheng and Chris Johnson*

Institute of Information Science, Academia Sinica, Taiwan
kevinc@iis.sinica.edu.tw, http://www.iis.sinica.edu.tw/~kevinc

* Department of Computing Science, University of Glasgow, UK
johnson@dcs.gla.ac.uk, http://www.dcs.gla.ac.uk/~johnson

Abstract: Current research in design of Context-Aware applications appears very technology focused, in particular software and sensor development and deployment rather than utilizing Human-Computer Interaction (HCI) principles such as task analysis to assist design of the applications. Developers specify what context-aware behavior to implement and determine what context information is needed based on intuition and introspection when they design applications of this kind. As a result, users of context-aware applications may have to repair the inappropriate predictions which the applications make about based upon likely user tasks, and immediate environments. This paper describes the approach of utilizing Hierarchical Task Analysis (HTA) to analyze the interaction between users and context-aware applications. The purpose of this research was to address the issue that there is not enough knowledge about where to discover and how to exploit context information. This is arguably the biggest obstacle when designing this type of applications. Case studies on analyzing existing context-aware applications are presented. The intention is to demonstrate the effectiveness and indicate the limitations of using HTA to better understand context-aware interaction. It is important to stress that we intend to exploit existing task analysis techniques instead of creating a new approach to validate existing context-aware applications. HTA provides an easy entry level for developers who have little knowledge about task analysis to inspect the existing context-aware applications and understand what previous developers of the applications consider context information and how they transform the information as input to their applications. This paper also describes a hybrid task analysis approach for modeling context and facilitating design of applications of this type. A case study is presented to validate this approach.

Keywords: Context-Aware Application, Hierarchical Task Analysis (HTA), Scenario-Based Design, Entity-Relationship Modeling

Introduction

Context-Aware Computing is currently a hot topic in multidisciplinary research fields. The trend is that computing devices and applications serve their users beyond the traditional desktop into diverse environments. A number of researchers have made claims about the benefits of context-aware applications. The applications can exploit not only explicit input from their users but also implicit input both from the users and their immediate surroundings to provide information tailored to the users' tasks (Chen and Kotz, 2001, Dey, 2001, Schmidt, 2000a). According to Schmidt, an implicit input is a user action that the user does not intend to perform to interact with an application. However, the application recognizes its meaning and considers as an input (Schmidt, 2000a). Implicit inputs may come from sensors which sense factors about user's activities, surroundings, location, etc (Masui and Siio, 2001, Selker and Burleson, 2000). The effect is to reduce the explicit input efforts as well as attention performed by the users when using applications of this kind. However, this increases the complexity associated with the design of context-aware applications. Developers must know what changes from users or environments are related to the tasks that the users perform to achieve goals with help from the applications. The changes may be considered as context information if the connection between user and application tasks and the changes can be found. However, the current state of this field is that there is no systematic approach to identify context information of applications uses in different domains. This results in that context information is defined by intuition and handled in improvised manner. In order to validate the claimed benefits, first of all, we must understand what context is considered and how it is exploited in the existing applications. However, it is a non-trivial task due to the considerable disagreement over the definition of "context-awareness" in human-computer interaction.

When designing an application, the first step is to understand what the application should be doing. Task analysis can help developers understand what needs to be accomplished by the application and break down the major task, the purpose of the application, into the simplest component part. Each component consists of clear goals and tasks.

In order to carry out the application, developers need to know what information is needed for each task. The information may come from the user input or the state of current runtime environment. Traditional human-computer interaction requires explicit user input whereas context-aware applications can adapt both explicit and implicit input that is regarded as context information. In the case of designing a context-aware application, developers need to know what user tasks are necessary to operate the application and also need to figure out which part of user input can be transferred to the application task in order to increase the level of context-awareness of the application.

According to Lieberman et al, it is important to identify user, user and application task, and application models of a context-aware application in order to validate how it simplifies the interaction scenario. User model holds information about the user's current and past state and preferences that related to the current tasks. The user and application task model captures actions that are performed by a user to complete a task with help from an application. Application model describes the capabilities of the application itself (Lieberman and Selker, 2000). This paper reviews the existing context-aware applications that have emerged to help users in different domains. The hypothesis is that by utilizing HTA and our proposed hybrid approach in the light of Scenario-Based design and Entity-Relationship Modeling techniques for analyzing the interaction between users and context-aware applications, we can see what and how user's tasks could be reduced and better understand what context-awareness entails.

Describing Context-Aware Applications Subject to HTA

Hierarchical Task Analysis (HTA) focuses on the way a task is decomposed into subtasks and the order and conditions where these are executed. They are represented as a hierarchy of tasks, subtasks and plans. The output of HTA can be represented textually and diagrammatically (Dix et al., 1998, Shepherd, 1989). This technique has the potential to represent goal structures and cognitive decision-making activities. It is similar to GOMS in terms of its nature of decomposition, sequence representation, and task operation selection. However, GOMS is designed to express the internal cognitive processes when a person performs tasks. Thus, application actions as well as feedback are not stated in a GOMS task model. This raises difficulties for developers to capture application tasks in application-supported scenarios. In this research, we mainly focus on observable users' actions rather than their internal cognitive state. Therefore, this technique is not suitable for our approach.

Some researchers argue for the advantage of HTA is at an early stage in the design process (Carey et al., 1989). It can provide brief picture of user tasks and basic functional specification of the proposed application. It can also be exploited as a rough-and-ready tool at this stage. The nature of decomposition enables developers to concentrate on parts of the overall task without losing the picture of overall task activities. In addition, the top down structure ensures completeness and is easy to comprehend (Carey et al., 1989, Shepherd, 1989). Regarding the task-design mapping, HTA provides a clear description of all task functions for mapping on to the new interactive application. It is also ideal for the identification and mapping of information input and output requirements in design of applications (Carey et al., 1989). We can take advantage of this easy yet hands-on task analysis technique to identify the input and output of a context-aware application in order to understand how the developers of previous projects consider context information and how they transform the information as input to their s. The textual representation of HTA in this paper follows the format listed in (Dix et al., 1998). The following sections focus on the existing context-aware applications in different domains. In particular, we are interested in the analysis of user, user-application, and application tasks in each application scenario. These are highlighted after each task and sub-task. We look at the comparison between context-aware and non-context-aware approach based on the same scenario to illustrate differences in the resulting effect. This emphasizes the benefits of context-aware applications.

Office Utilities: The early demonstration of context-aware applications focused on tailoring, disseminating, and presenting information to users based on the current locations in the office domain. The intention is to improve not only the interaction between office workers but also the effectiveness of using computers, printers, and electronic equipment that helps with tedious repetitive tasks. The following describes call-forwarding service in this domain.

The call-forwarding application scenario is that the users can be tracked within a building and phone calls are forwarded to the nearest phone to them. The first demonstrations were based on the Active Badge application (Want et al., 1992). Users are required to wear badges and move around the building. Their location information can be obtained and updated to the database by the application. The database contains information about the users' current or most recent location, whether or not they are in their working places or offices. It also contains status message, and the nearest phone extension. When the receptionist receives a phone call for a particular user, she can

use the database to look up the recipient's location information and forward the phone call to his/her last known location. The following scenario is based on the call-forwarding application at AT&T Laboratories Cambridge and assumes that the intended recipient exists (Cambridge, 1992a) (Cambridge, 1992b). This is not the only HTA we can produce. It is simply an example. The HTA of call-forwarding is as follows:

Non-Application Support:

```
Receptionist:

0. Receptionist forwards calls to intended recipient

       1. pick up the incoming call (User task)
       2. converse with the caller (User task)
       3. identify the intended recipient (User task)
       4. check the recipient's status (User task)
            4.1 check the recipient's in/out status from in-out board (User task)
            4.2 check the recipient's schedule (User task)
       5. appoint the next call with the caller for the recipient (User task)
       6. forward the message to the recipient (User task)
       7. check the recipient's extension number (User task)
            7.1 Get the phone list (User task)
            7.2 Look up the phone list to obtain the recipient's extension number (User task)
       8. forward the call to the phone in the office or somewhere close to (User task)

Plan 0: 1 - 4 in that order
        if the recipient is not available
        then 5 - 6
        else 7 - 8

Plan 1: do 4.1 - 4.2 in that order

Plan 2: do 7.1 - 7.2 in that order

Recipient:

0. update the current status (in/out, in a meeting, and etc.)
       1. go to reception desk/office (User task)
       2. inform the receptionist about in/out status and activity status (User task)

Plan 0: do 1 - 2 in that order


0. receive calls from others

       1. answer the nearest phone (User task)

Plan 0: do 1
```

We refined the HTA and include Active Badge in the analysis to see what happen when users use this technology.

Application Support:

```
Receptionist:

0. receptionist forwards calls to intended recipient
       1. pick up the incoming call (User task)
       2. converse with the caller (User task)
       3. identify the intended recipient (User task)
       4. check the recipient's status from application database (User-application task)
       5. appoint the next call with the caller for the recipient
       (User task)
       6. update the message to the recipient correspondent database entry (User-application
       task)
       7. forward the call to the phone close to the recipient
       (Use the result from task 4) (User-application task)

Plan 0: do 1 - 4 in that order
        if the recipient is not available
        then 5 - 6
        else 7

Recipient:

0. update current status
       1. operate the Active Badge (User-application task)

Plan 0: do 1
```

```
0. Receive calls from others
       1. answer the nearest phone (User task)

Plan 0: do 1
```

The HTA reveals that the application reduces the number of user tasks required for the receptionist and the recipient to perform call-forwarding. The application considers its users' location and their current status as context information. The application can obtain the context information about the user's identity, location, and timestamp of last seen and update the database automatically. That means the application considers the recipients' movement as an implicit input. This makes the interaction flows of completing the task smoother than doing the task without application support. It reduces the physical activities required to explicitly update information about the recipients' current situation. Updating one's current location requires that the telephone recipients go to inform the receptionist. Also, receptionist need not iterate to check recipients' current status and extension number using paper based list. Instead, the application can help integrate the information about the last known location, status, and closest phone extension to the recipients.

It is argued that the users want to have more control over the subsequent interaction tasks, depending on their current situation such as they do not want to take unexpected calls or receive instant messages when they are in a meeting (Adams, 2002, DeVaul and Dunn, 2001, Schigeoka, 2002). User's preference should be taken into account to make the application meet their users' social need. This illustrates key point about HTA gives no user perception of preference in each step.

Tour Guides: When a person visits a city or an exhibition, she can go to an information centre or counter to get a paper-based map and use it to guide herself. However, visitors might get lost if they cannot find the link between the physical place and the map. Other situation such as they might want to have personalized visiting routes. Context-aware applications in this domain tend to provide their users with information about their current location and suggest routes based on user's preferences (Chan, 2001b, Davies et al., 2001, Long et al., 1996, MacColl et al., 2002, Oppermann and Specht, 1999, Spasojevic and Kindberg, 2001, Youll et al., 2000). The user's preference may be obtained from a history of where previous users have been or the user's interests (Galani and Chalmers, 2002).

Many indoor exhibitions, for instance, museums provide their visitor not only with paper-based guides but also tape recoded guides. Both mediums provide predefined visiting routes and lack flexibility to adjust itself to suit their users' needs based on their current situation. For example, a visitor may feel bored with her current route or attracted by a particular exhibit. She may want to have another choice of visiting path. The paper-based and audio guide cannot support the dynamic nature of visitors' interests. The Hippie application was developed to avoid this limitation. It is an context-aware guide application in an indoor environment (Broadbent and Marti, 1997, Oppermann and Specht, 2000). The visitor carries a PDA and wears earphones while walking within the museum. Each exhibit is equipped with an infrared transmitter, which is used as a link to the corresponding digital information stored in the application. In the original prototype, information about the exhibits in the museum was cached on a PDA. The current development of Hippie has incorporated wireless LAN to provide dynamic information to the users.

Non-Application Support:

```
0. visit a museum using paper-based guide
       1. obtain a paper-based guide from the counter (User task)
       2. choose a categorized visiting path (User task)
       3. follow the categorized visiting path (User task)
       4. walk abound the museum (User task)
       5. stop at the interested exhibit (User task)
       6. look up information about the exhibit on the guide (User task)
       7. read the description of the exhibit displayed around the exhibit (User task)
Plan 0: do 1
        if categorized visiting route provided on the guide
        then do 2 - 7
        else do 4 - 7

0. visit a museum using audio guide
       1. obtain a audio guide from the counter (User task)
       2. choose a categorized visiting path (User-application task)
       3. follow the categorized visiting path (User task)
       4. walk abound the museum (User task)
```

```
        5. stop at the interested exhibit (User task)
        6. press the number displayed on the exhibit on the audio guide keypad (User-application
        task)
        7. hear the description of the exhibit displayed around the exhibit (User task)

Plan 0: do 1
        if categorized visiting route provided on the audio guide
        then do 2 – 7
        else do 4 - 7
```

Application Support:

```
0. visit a museum using context-sensitive mobile computing system (Hippie)
        1. obtain the device from the counter (User task)
        2. choose a preferred visiting path organized by the system
        (User-application task)
        3. follow the visiting path (User task)
        4. stop at the exhibit interest you (User task)
        5. information about the exhibit is presented through the earphone (Application task)
        6. want to discover different topic (User task)
        7. change current visiting path to another (Application task)
        8. follow the visiting suggested by the system (User-application task)

Plan 0: do 1 – 4 in that order
        if the visitor is attracted by something else
        then do 5 – 8 – 3 – 4
```

The Hippie development team claimed that the application utilizes the user's location/presence and preference as context information to simplify the user's visiting task. From the HTA, the task 2 and 3 in the non-application support are conditional while they are unconditional tasks in the application support. This emphasizes that a personalized visiting path is an essential function for a context-aware guide application. To personalize a visiting route for the visitor, the application asks the visitor what kind of tour they would prefer and then guides the visitor based on her preference. At this stage, the application needs to gather context about its user's preference explicitly from the visitor. During the visit, the application shows the visitor her current location and the path to the next planed exhibit on the PDA. This reduces the effort that the visitor needs to check the paper-based guide. An audio guide is arguably better than a paper-based guide because the visitor can visually focus on the physical environment and audibly receive the direction guide to the next exhibit. The HTA revels that the user's task of finding a description of an exhibit in the non-application support section can be transformed to an application task by adapting to user's location information. The task 5 to 7 in the non-application support section requires explicit interaction between the visitor and a paper-based guide or an audio guide. The visitor has to match the label on the exhibit with either the label on the paper-based guide or press the corresponding label (i.e. number) on the audio guide keypad to read or hear the description. However, the application support can detect the visitor's location and provide the information about exhibits automatically. As for task 6 to 8 in the application support, it is an exclusive for the context-aware guide application. For example, the visitor may be interested in a specific exhibit and want to know more about it by selecting the detail information option on the PDA's screen. The application can sense the implicit changes about the visitor's interest from the interaction between the visitor and the PDA. It may then suggest a new route to visit the rest of the exhibits in the museum.

Social Enhancement: In this application domain, we focus on the application that can recognize and adapt themselves to their user's current social situation while providing services. Current development of mobile phone is not designed for context-aware. Users must set an appropriate operation mode for their social setting. However, users often forget to setup their mobile phone to meet the current situation. Research on context-aware mobile phone focuses on the user's current situation, for example, location, activity, and co-location of the user and her mobile phone (i.e. in the pocket, in the user's hand, on the desk, etc.) and utilizes the information to enhance the quality of usage in terms of social aspects (DeVaul and Dunn, 2001, Lijungstrand, 2001, Schmidt et al., 2000, Tuulari, 2000). For example, a mobile phone detects that its user is in a meeting and does not want to receive any call except emergency ones. The mobile phone can then adjust itself to the "meeting" mode and apply the appropriate call filter during the meeting. Inspired by instant messaging (IM) services, Schmidt et al implemented their concept of "context-call" over the Wireless Application Protocol (WAP) (Schmidt, 2000b). In this case, the user or the mobile phone itself can publish the current situation and contact method to the central server. Callers contact the user by making a context-call in the same way as using IM services to see the status of a recipient and decide to make a call, leave a message, or call the user later.

The following scenario is based on the context-call development in TecO (Schmidt, 2000b). The scenario shows that a person is in the middle of a meeting at the customer site. One of her colleagues is calling her about going for a drink later.

Non-Application Support:

```
The person in a meeting:

0. change the mobile phone status to "meeting" mode
        1. press appropriate key set on the mobile phone
        (User-application task)
        2. check/answer the phone (User-application task)

Plan 0: do 1
        if the incoming call goes through the filter
        do 2
```

Application Support:

```
The person in a meeting:

0. change the mobile phone status to "meeting" mode
        1. check/answer the phone (User-application task)

Plan 0: if the incoming call goes through the filter
        do 1
```

The HTA shows the task 1 in non-application support is transferred from explicit user-application task to application task. The user's activity of walking into the meeting room is considered as implicit input for the system. The application support approach allows the user focus on his current tasks in the meeting with her customer and do not have to explicitly adjust her mobile phone to "meeting" mode. The user need not worry about whether the mobile phone has been set to an appropriate mode.

Games: A number of recent research implementations have built context-aware games to expand the arena from virtual space to mixtures of virtual and physical space (Bjork et al., 2001, Falk, 2001, and Headon, 2001). The aim is to evaluate how traditional game design can benefit from mobile computing, wireless communication, and sensor technologies. They want to investigate how to maintain and encourage social interaction in play. We look at "Pirates!", a context-aware multi-player game, and apply HTA to illustrate the differences between context-aware support and traditional game playing. This game exploits context information about its player's location, other players' location, and the location of game objects, such as treasures. The game scenario is that each player represents the captain of their ship. They have to walk around the physical game arena to obtain treasure and earn points. They may, however, be engaged in a battle with other ships nearby. Playing this game, the player carries a handheld device equipped with a wireless connection and a sensor receiver while they are moving around the physical game environment.

Non-Application Support:

```
0. play the Pirates! game
        1. move the game character around using game pad or keyboard (User-application task)
        2. search for treasures (User-application task)
        3. attack other ships (User-application task)
Plan 0: do 1 - 2 in that order
        if encounter other ships
        then do 3
```

Application Support:

```
0. play the Pirates! game using context-sensitive mobile computing device
        1. move around the game character by physically walking around the physical game arena
        (User task)
        2. search for treasures (User-application task)
        3. attack other ships (User-application task)

Plan 0: do 1 - 2 in that order
        if encounter other ships
        then do 3
```

From the HTA, we see the task 1 in the non-application support can be transferred from explicit user task to implicit user task. Namely, the application regards the player's movement is an implicit input. The game character,

the ship, moves while the players walk around instead of pressing the buttons on a game pad. The benefit of application support is that the player can immerse into the game. The immersive experience in the game play would increase the level of excitement when the player playing the game (Headon, 2001, Schneider, 2001). Augmented Reality (AR), which tackles the research issue of interaction between human, physical, and virtual entities, is rather suitable to describe the interaction between the player, game application, and physical and digital game arena. Many researchers in this field tend to exploit the context-aware game applications as social interaction test-bed to discover more about how the players react to each other on particular game tasks (Dennis, 2001, Pering, 2001, Schneider, 2001).

We learned a lot about HTA. It does not capture human factor and social issues very well. Call-forwarding applications with active tracking sensor mechanisms allow users concentrate on performing relevant tasks to deal with their current situations without the disturbance from the application tasks. However, users may lose control of their privacy. In the case of smart mobile phone interaction, the real issue is not work saved for users but annoyance to their colleagues. In the context-aware game scenario, the HTA cannot address the issue of enjoyment.

A Hybrid Approach for Modeling Context Information

This section introduces a systematic approach for finding innovative uses for future technologies. It is to extract user tasks from situations that are elicited from a scenario. As noted by Carrol, scenarios are stories about people and their activities (Carroll, 2000a and Carroll, 2000b). Each scenario has a setting that explicitly describes the starting state of the current situation and implicitly depicts the characters that take part in the situation in the scenario. Each scenario has actors who perform tasks to achieve goals in different situations in a scenario. Each task can be regarded as what needs to be done in the situation. We analyze the user tasks in terms of the answers to the questions, “Who should be responsible for the situation?”, and “What should be known to act on the situation?”. HTA is utilized to picture and describe what happens in a scenario and presented in user, user-application, and application tasks performed in a scenario. In order to figure out the transformation between user and computer application tasks, we also adapt the Entity-Relationship Modeling to identify the relationships between entities, actors, and actions described in the HTA. In addition, the user would feel easier to stay in role and resolve any potential hesitation if the adapted scenario can reflect situation based on the user’s previous experiences with realistic reasons for performing the tasks. The closer that the scenario represents reality, the more chance the useful context is discovered.

A Case Study

To illustrate how the approach can be applied, we introduce a case study, Virtual Notelet (Cheng and Johnson, 2001). Using this application, the user carries a wireless-enabled mobile device and walks around in the office environment. When the user stands in front of an office door, the device displays the occupant’s status in the office and virtual notes virtually attached on the office door. The user can adjust her status in her office and leave virtual notes to others.

Interaction at office doors happens frequently in office environments. Office doors do not simply act as physical barriers to particular rooms. They also play a significant role in communicating information about the location and availability of the occupant. In a wider sense, they can also be thought of as a medium of communication for information from the occupant to her colleagues and vice versa. The doors to communal and shared locations play a similar role. It is often possible to tell if a room has been booked by looking for notes attached to the door. Similarly, if a meeting is being conducted then the same approach can be utilized to indicate whether it is socially acceptable to interrupt that meeting or not. People in the office domain may stop by an office door in order to find out if a colleague is in her office. In this situation, the visitor encounters a problem about whether to enter or not to do so whereas the occupant faces intrusion if she is engaged in something. How does the visitor put an anchor to the uncompleted activity in order to resume the interaction with the occupant if the occupant is not in the office or unavailable for the visitor? From our preliminary observation, some staff in our department use annotations to indicate their current status in the offices. Some of them provide their visitors with Post-It notes, which are attached on the door, so they can leave messages on the door. Some authors argued that people’s actions at a door are determined by the status of the occupant in the office (Selker, 2000). Their observation shows most visitors perform the action “knock and wait”, “check status”, or leave notes”. The actions such as “walk in” and “knock and walk in” are rarely happen. This reflects the importance of the annotation on an office door.

There are two situations we are interested. Firstly, a person stands in front of an office door and checks the status of the occupant. Secondly, an office occupant stands in front of her office door, and manipulates the annotations to

indicate her in/out status, check the messages left by the visitors, and opens the door in order to enter her office or close the door to leave her office. The following describes three different situations when using the Virtual Notelet. The user task at office door is described using HTA to show a high level view of the interaction. The label, Role, is a light weight user model to indicate the type of user in the application scenario. We also utilize the entity-relationship modeling to figure out the entities, actions, and actors involve in the interaction n the scenario. The relationship between the entities, actions, and actors is important when we try to transform user tasks to computer tasks.

Situation 1: Approaching a colleague's office and want to know her status in the office when standing in front of the door.

Role: visitor

```
0. in order to meet the colleague in her office
        1. walk by the office door
        2. check the context board attached on the door
        3. leave a note (using  Post-It notes)
                3.1 write message on the note
                3.2 detach the note from the pile of Post-It notes
                3.3 attach the note on the door
        4. knock the door
        5. wait few seconds
        6. open the door
        7. walk in

Plan 0: do 1-2-4-5-6-7 in that order
         When the occupant is away from the office or busy in the office do 3

Plan 3: do 3.1-3.2-3.3 in that order
```

Object Visitor **human actor**

Actions:

```
        V1-1: walk to the office
        V1-2: check occupant's status showing on the door
        V1-3: leave notes
        V1-4: knock, open the door, and walk in the office
```

Object Post-It note **simple**

Attributes:

Affordances: hold/fold/attach/detach/draw or write

Events:

```
E1-1: occupant is free in the office
E1-2: occupant is busy in the office
E1-3: occupant is not in the office
```

Relations: object-object

```
Location (Post-It notes, office door)
Location (context board, office door)
```

Relations: action-object

```
patient (V1-2, context board and notes)
        - Visitor "sees" the context board and notes attached on the door
instrument (V1-3, Post-It notes)
        - Visitor writes down messages on the note and attaches it on the door using its
        self-attaching area on the back
patient (V1-4, door)
        - Visitor knocks and opens the door
```

Relations: action-event

```
before (V1-2, V1-3)
        - Visitor must check the office occupant's status before deciding whether to
        leave a note
triggers (E1-1, V1-4)
        - "Occupant's status is free" triggers the visitor to knock, open the door, and
        walk in the office
triggers (E1-2, V1-3)
        - "Occupant's status is busy" triggers the visitor to decide to leave a note
triggers (E1-3, V1-3)
        - "Occupant's status is away" triggers the visitor to decide to leave a note
```

As mentioned previously, the task analysis is not an end in itself. For instance, the events listed in the previous paragraph are very unlikely to provide a complete description of the changes that must be considered by the

system. In contrast, the HTA represents an initial stepping stone between the informal scenario and the more detailed information required to move towards a prototype implementation. Both the scenario and the task analysis are refined by the insights that are provided once users can access the system. Considering building a context-aware application to help the user perform these actions we should determine what actions the application needs to perform and what input it expects. From the task analysis listed above, there are two human actor actions, V1-2 and V1-3, we are interested in. In more detail, the application should display the occupant's status in the office and provide a Post-It note like function so that a visitor can write messages and post it on the office door. From the application point of view, when its user stands at an office door it must first identify his/her role in the ongoing interaction. This can be done by requiring the user to "login" so the application knows whether she is a visitor or an occupant in the office. The login process can be implicitly adapting sensing technology or explicitly asking the user to type in her ID and password. Once the user's role is obtained, the application can perform subsequent actions. For instance, the application displays the occupant's status in the office. The acquisition of the information is described in situation 2. As shown in the action-event relations section, the occupant's status in the office determines the visitor's subsequent action, "leave a note or knock the door". The occupant's status can be regarded as an input to the application to activate its Post-It like function to the user. The user can write messages on the virtual note and virtually attach it on the office door.

Object Virtual Notelet **non-human actor**
Actions:
VN1-1: identify the user's role
VN1-2: display the occupant's status in the office
VN1-3: activate note editor
VN1-4: associate the virtual note with the physical office door

Object Virtual Post-It note **simple**
Attributes:
Affordances: virtually attach/detach/draw or write

Relations: object-object
Location (virtual note, computing device (i.e. PDA))

The following indicates the input to each action performed by the Virtual Notelet. Reversely, the input is interpreted and presented as context information related to the application actions. It is important to note that these are not the label listed in the entity-relationship model.

Input to Virtual Notelet
user ID and password • VN1-1
occupant's status in the office (i.e. in/out) • VN1-2
occupant's "bust" or "away" status • VN1-3
user fires "attach" command • VN1-4

Context in Virtual Notelet
[user] • VN1-1
[office occupant's status] • VN1-2, VN1-3
[virtual note manipulation command] • VN1-4

The following describes the interaction at occupant's office door when coming back to the office.

Situation 2: Office occupant is approaching to her office door from outside of the office and she wants to adjust her in/out status and check the notes left by others when she stands in front of the door.

Role: occupant

0. enter her office
1. walk by her office door
2. adjust the in/out status on the context board on the door
3. check the notes attached by visitors on the door
4. remove notes
5. open the door
6. walk in

Plan 0: do 1-2-3-5-6 in that order
if any note attached on the door then do 4

Object Office occupant **human actor**

Actions:
V2-1: walk to the office
V2-2: adjust in/out status showing on the door
V2-3: check and remove notes from the door
V2-4: open the door and walk in the office

Object Post-It note **simple**
Attributes:
Affordances: hold/fold/attach/detach/draw or write

Object context board **simple**
Attributes:
Affordances: adjustable indicator

Events:
E2-1: notes left by the occupant are attached on the door
E2-2: notes left by other visitors are attached on the door

Relations: object-object
Location (Post-It notes, office door)
Location (context board, office door)

Relations: action-object
patient (V2-2, context board)
- occupant adjust the in/out status displayed on a context board
patient (V2-3, Post-It notes)
- occupant "see" and remove the notes attached on the door
patient (V2-4, door)
- occupant open the door and walk in

Relations: action-event
before (V2-2 or V2-3, V2-4)
- occupant adjust her in/out/busy/free status and check notes attached on the door before she enter the office

triggers (E2-1 or E2-2, V2-3)
- notes left and attached on the door by the occupant or others trigger the occupant perform the action "remove the notes"

When the user arrives at her office door the application must identify the relationship between the user and the office as described in situation 1. If the user's role is identified as the occupant of the office she can adjust her in/out status manually on the computing device or implicitly updated by the application if it embodies a more sophisticated user model (i.e. meeting schedule, location, and etc.). If a virtual note has been left by others or the occupant herself, the application displays the notes on the user's computing device.

Object Virtual Notelet **non-human actor**

Action:
VN2-1: identify the user's role
VN2-2: activate the virtual context board
VN2-3: display virtual notes and provide the user with note manipulation function
VN2-4: modify the relations between the virtual note and the physical door

Object virtual context board **simple**
Attributes:
Affordances: virtually adjustable indicator

Object Virtual Post-It note **simple**
Attributes:
Affordances: virtually attach/detach/draw or write

Relations: object-object
Location (virtual note, computing device (i.e. PDA))
Location (context board, computing device (i.e. PDA))

Input to Virtual Notelet
user ID and password • VN2-1
occupant stands at the door • VN2-2
notes attached on the door • VN2-3
user fires "remove" command • VN4

Context in Virtual Notelet

```
[user] • VN2-1
[occupant's location] • VN2-2
[virtual note attached on the door] • VN2-3, VN2-4
```

As mentioned, "Input" indicates the input to each action performed by the Virtual Notelet. Reversely, the input is interpreted and presented as context information related to the application actions. Situation 3 describes the occupant's interaction at the door when leaving the office.

Situation 3: Office occupant is approaching to her office door, opening, walking out, and locking the door. She adjusts the in/out status and may leave notes to state further information when she stands at the door.

Role: occupant

```
0. leaving her office
        1. open the door
        2. close the door
        3. adjust the in/out status on the context board attached on the door
        4. check and remove notes
        5. leave a note for extra message
                5.1 write message on the note
                5.2 detach the note from the pile of Post-It notes
                5.3 attach the note on the door
        6. lock the door and leave

Plan 0: do 1-2-3-6 in that order
         if any note attached on the door then do 4.
         if further message is needed then do 5

Plan 5: do 5.1-5.2-5.3 in that order
```

Object Office occupant **human actor**

Actions:

```
V3-1: walk to the office door
V3-2: open the door, walk out, and close the door
V3-3: check and remove notes from the door
V3-4: adjust in/out status showing on the door
V3-5: lock the door
```

Object Post-It note **simple**

Attributes:

Affordances: hold/fold/attach/detach/draw or write

Object context board **simple**

Attributes:

Affordances: adjustable indicator

Events:

```
E3-1: occupant is free in the office
E3-2: occupant is busy in the office
```

Relations: object-object

```
Location (Post-It notes, office door)
Location (context board, office door)
```

Relations: action-object

```
patient (V3-2 and V3-5, door)
        - occupant open, close, and lock the door
patient (V3-3, Post-It notes)
        - occupant check and remove the notes
patient (V3-4, context board)
        - occupant adjust her in/out status displayed on the context board
```

Relations: action-event

```
trigger (E3-2, V3-3)
```

When the user stands at the office door, the application must identify the relationship between the user and the office as described in situation 1 and 2. If the user's role is identified as the occupant of the office she can adjust her in/out status manually on the handheld or it can be implicitly updated by the application if it embodies a more sophisticated user model (i.e. meeting schedule, current location, and etc.). If a virtual note has been left by others or the occupant herself, the application displays the notes on the user's handheld device. From the user task

analysis listed above, we see that the office occupant's status and notes that attached on the office door interest both a visitor and an office occupant when they stand at the door and influence their subsequent tasks.

Object Virtual Notelet **non-human actor**
Action:
VN3-1: identify the user's role
VN3-2: activate the virtual context board
VN3-3: display virtual notes
VN3-4: modify the relations (remove) between the virtual note and the physical door and provide the user with note manipulation function

Object virtual context board **simple**
Attributes:
Affordances: virtually adjustable indicator

Object Virtual Post-It note **simple**
Attributes:
Affordances: virtually attach/detach/draw or write

Relations: object-object
Location (virtual note, computing device (i.e. PDA))
Location (context board, computing device (i.e. PDA))

Input to Virtual Notelet
user ID and password • VN3-1
occupant stands at the door • VN3-2
notes attached on the door • VN3-3
user fires "remove" and "create" virtual note command • VN3-4

Context in Virtual Notelet
[user] • VN3-1
[occupant's location] • VN3-2
[virtual note attached on the door] • VN3-3
[occupant's status in the office] • VN3-4

To sum up, the context information supported by the initial prototype of the Virtual Notelet application will include "the role of the user", "the user's location", "office occupant's status", and "virtual note on the office door". The user's location triggers the information presentation about the occupant's status in the office and notes left by the occupant or others virtually attached on the door. The context, occupant's status in the office, determines the visitor's tasks at the office door such as leaves a note or "knocks and walks in".

Conclusions

Most existing context-aware applications were designed in an improvised way. The aim of this paper is to utilize task analysis techniques as tools for thought for designers when they design applications of this kind. It describes the approach of exploiting HTA to identify tasks in existing context-aware applications in different domain in attempt to better understanding of context-awareness. We are interested in user tasks in a scenario. In particular, we focus on the actors, goals, and settings of a scenario. We concentrate on the way users perform tasks to accomplish goals. The point is that task analysis can help us to move from a scenario to a more concrete design. User testing can then be used to observe limitations with the task analysis that will only be apparent when real people actually start to use the application. We also utilize Entity-Relationship Modeling to identify the relationship among actors, objects, and actions when users perform their tasks. The aim is to figure out which part of user tasks can be transformed to applications if the level of context-awareness is increased on the application side. The hybrid task analysis technique is applied to the design scenario of Virtual Notelet in order to identify actors, goals, and actions that include both with and without application support when users perform tasks in the application scenario. Our finding is that HTA and the hybrid approach do not reduce the complexity nature of context-aware applications but provide a blueprint of modeling context information for application uses in diverse domains. It is important to stress that this approach should encourage developers with more knowledge about underlying technologies of context-aware applications to concentrate more on observation of user activities that are performed to achieve goals in specific application domains. The hope is that more novel and meaningful context-aware applications will be discovered and developed.

References

Adams, D. (2002). Programming Jabber, 1st edition. O'Reilly, Sebastopol, CA, USA.

Bjork, S., Falk, J., Hansson, R. and Ljungstrand, P. (2001). Pirates! - Using the Physical World as a Game Board., Interact 2001, IFIP TC.13 Conference on Human-Computer Interaction, Tokyo, Japan, 2001.

Broadbent, J. and Marti, P. (1997). Location Aware Mobile Interactive Guides: usability issues, Fourth International Conference on Hypermedia and Interactivity in Museums (ICHIM97), pp.88-98. Paris, 1997.

Cambridge, AT&T Laboratory (1992). Active Badge on ``Beyond 2000'', http://www.uk.research.att.com/pub/videos/qsif-200/beyond-qsif-200.mpg, 1992

Cambridge, AT&T Laboratory (1992). The Active Badge System, http://www.uk.research.att.com/pub/videos/qsif-200/badge-qsif-200.mpg, 1992

Carey, M. S., Stammers, R. B. and Astley, J. A. (1989). In Task Analysis for Human-Computer Interaction (Ed, Diaper, D.) Ellis Horwood, pp. 56-74.

Carroll, J. M. (2000a). Five reasons for scenario-based design, Interacting with Computers, **13,** pp. 43-60.

Carroll, J. M. (2000b). Making Use Scenario-Based Design of Human-Computer Interactions, The MIT Press, Cambridge, Massachusetts.

Chan, W. (2001). Project Voyager: Building an Internet Presence for People, Places, and Things, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology

Chen, G. and Kotz, D. (2001). A Survey of Context-Aware Mobile Computing Research, Dept. of Computer Science, Dartmouth College

Cheng, Y-M. and Johnson, C. (2001). The Reality Gap: Pragmatic Boundaries of Context Awareness, Blandford, A., Vanderdonckt, J. and Gray, P., *IHM-HCI 2001*, pp.412-427. Springer, Lille, 2001.

Davies, N., Cheverst, K., Mitchell, K. and Efrat, A. (2001). Using and Determining Location in a Context-Sensitive Tour Guide, IEEE Computer, **34,** 35-41.

Dennis, B. (2001). On Integrating First Person Shooter Games and Ubiquitous Environments, UbiComp2001, Atlanta, U.S., 2001.

DeVaul, R. W. and Dunn, S. (2001). The Context Aware Cell Phone Project, http://www.media.mit.edu/wearables/mithril/phone.html

Dey, A. K. (2001). Understanding and Using Context, Personal and Ubiquitous Computing, **5,** pp. 4-11

Dix, A., Finlay, J., Abowd, G. and Beale, R. (1998). Human-Computer Interaction, Prentice Hall Europe, 2nd edition.

Falk, J. (2001). The Nexus: Computer Game Interfaces for the Real World, UbiComp2001, Atlanta, U.S., 2001.

Galani, A. and Chalmers, M. (2002). Can You See Me? Exploring Co-Visiting Between Physical and Virtual Visitors, Museums and the Web 2002, Archives & Museum Informatics, Boston, U.S., 2002.

Headon, R. and Curwen, R. (2001). Ubiquitous Game Control, UBICOMP Workshop on Designing Ubiquitous Computing Games, Atlanta, GA, USA, 2001.

Kindberg, T. and Barton, J. (2001). A Web-based nomadic computing system, Computer Networks (Amsterdam, Netherlands), **35,** pp. 443-456.

Lieberman, H. and Selker, T. (2000). Out of Context: Computer systems that adapt to, and learn from, context, IBM SYSTEM JOURNAL, **39,** pp.617-632.

Lijungstrand, P. (2001). Context Awareness and Mobile Phones, Personal and Ubiquitous Computing, **5,** 58-61.

Long, S., Kooper, R., Abowd, G. D. and Atkeson, C. G. (1996). Rapid prototyping of mobile contextaware applications: The cyberguide case study, conference on Human Factors in Computing Systems ,CHI'96, pp.97-107. 1996.

MacColl, I., Brown, B., Benford, S. and Chalmers, M. (2002). Shared Visiting in EQUATOR City, Collaborative Virtual Environments 2002, Bunn, 2002.

Masui, T. and Siio, I. (2001). Real-World Graphical User Interfaces, Thomas, P. and Gellersen, H. W., *HUC 2000*, pp.72-84. Springer-Verlag, Bristol, U.K., 2001

Oppermann, R. and Specht, M. (1999). A nomadic Information System for Adaptive Exhibition Guidance, ICHIM99, International Cultural Heritage Meeting, pp.103-109. Washington, D.C., U.S., 1999.

Oppermann, R. and Specht, M. (2000). A Context-Sensitive Nomadic Exhibition Guide, Thomas, P. and Gellersen, H. W., *HUC2K*, pp.127-142. Spring-Verlag, Bristol, U.K., 2000.

Pering, T. P., C. (2001). Mercantile: Social Interaction Using a Mobile Computing Platform, UbiComp2001, Atlanta, U.S., 2001.

Schigeoka, I. (2002). Instant Messaging in Java, 1st edition, Manning Publications Co.

Schmidt, A. (2000a). Implicit Human Computer Interaction Through Context, Personal Technologies, **4,** pp. 191-199.

Schmidt, A., Takaluoma, A. and Mntyjrvi, J. (2000b). Context-Aware Telephony over WAP, pp.225-229. Springer-Verlag, London, Ltd., 2000.

Schneider, J. and Kortuem, G. (2001). How to Host a Pervasive Game - Supporting Face-to-Face Interactions in Live-Action Roleplaying, UbiComp 2001, Atlanta, GA, USA, 2001.

Selker, T. and Burleson, W. (2000). Context-aware design and interaction in computer systems, IBM SYSTEM JOURNAL, **39,** pp. 880-891.

Shepherd, A. (1989). In Task Analysis for Human-Computer Interaction(Ed, Daiper, D.) Ellis Horwood, pp. 15-55.

Spasojevic, M. and Kindberg, T. (2001). A Study of an Augmented Museum Experience, HP Laboratories, 19.07.2001

Tuulari, E. (2000). Context aware hand-held devices, MSc Thesis, VTT ELECTRONICS, NETWORKING RESEARCH, TECHNICAL RESEARCH CENTRE OF FINLAND, OULU

Want, R., Hopper, A., Falcao, V. and Gibbons, J. (1992). The Active Badge Location System, ACM Transactions on Information Systems, **10,** pp. 91-102.

Youll, J., Morris, J., Krikorian, R. and Maes, P. (2000). Impulse: Location-based Agent Assistance, Fourth International Conference on Autonomous Agents, Barcelona, Spain, 2000.