Beyond Belief: Representing Knowledge Requirements For The Operation Of Safety-Critical Interfaces

Chris Johnson

Glasgow Accident Analysis Group,
Department of Computer Science, University of Glasgow, Glasgow,
Scotland, G12 8QQ.

<http://www.dcs.gla.ac.uk/~johnson>
E-mail: johnson@dc.s.gla.ac.uk

ABSTRACT: Human intervention has played an important role in many major accidents. The staff of the London Underground continued to allow trains to deposit passengers in Kings' Cross after the fire had started. The crew of the Herald of Free Enterprise left their bow doors open. One reason why human 'error' is such a significant factor in major disasters is that companies frequently fail to consider the detailed requirements that must be satisfied by operator training. Accident reports often criticise the lack of structure and content in the courses that are provided for safety-critical applications. The following pages address this problem. A mathematical notation is used to represent knowledge requirements for off-shore oil production. This provides an explicit representation of the factual and procedural information that must be included in training courses. This approach also helps to document the standards by which an operator's performance will be assessed. The application of a formal notation is appropriate because a range of organisations, including the UK Ministry of Defence and NASA, are advocating these languages for large-scale development projects. Logic-based formalisms, therefore, provide a common language for systems engineering and human-factors development. This helps to ensure that training issues are not relegated to an after-thought in the design and implementation of safety-critical systems.

KEYWORDS: Safety-critical interfaces; formal methods, training; human factors; logic.

1. INTRODUCTION

Recent accidents in the transportation, power generation and chemical industries have illustrated the importance of ensuring that operators understand the rules and procedures which govern the operation of their applications. For example, the Piper Alpha enquiry found that:

"The lack of an exact format or content for the induction training; the brevity of time devoted to it; the almost cursory assessment of whether an individual required to attend the training...all point to the failure to ensure that all were properly informed on matters critical to their safety in an emergency." (Cullen, 1990)

Such findings are surprising because national and international legislation emphasises the importance of adequate training. For example, the Australian Petroleum (Submerged Lands) Act states that:

"...an employer contravenes (the act) if the employer fails to take all reasonable steps to provide to employees in appropriate language, the information, instruction, training and supervision necessary to enable them to perform their work in a way that is safe and without risk to their health".

If companies are to obey this legislation then they must provide the regulatory authorities with evidence which shows that they deliver appropriate training and supervision. Unfortunately, many companies document their training in terms of high-level objectives, such as "the operator shall be aware of the general operating procedures for this system" (Fennel, 1988). This makes it difficult for companies to defend their practices when failures do occur. Accident reports into Bhopal (Morehouse and Subramaniam, 1986), Kegworth (AAIB, 1990) and Three Mile Island (USNRC, 1980) all criticise the training practices of the companies involved.

This paper argues that formal notations can be used to represent training objectives for safety-critical interfaces. This is appropriate because commercial and regulatory bodies have recommended the use of these notations in the development of safety-critical systems (MOD, 1991). Formal methods have also been used to support the design of human-computer interfaces (Harrison and Thimbleby, 1989). For instance, Dix (1991) has exploited a mathematical notation to investigate timing problems in multi-user systems. Elsewhere, we have shown that probabilistic logics can be used to analyse the stochastic processes that characterise interaction with safety-critical systems (Johnson, 1993, 1995 1995b). An important benefit of these techniques is that they provide a bridge between current practice in both human computer interaction, through task analysis (P. Johnson, 1992), and software engineering (Fidge, Kearney and Staples, 1993). Unfortunately, formal notations have not been used to represent training requirements. This makes it difficult for trainers to reason about the rules and regulations that are intended to guide an operator's interaction with a safety-critical system.

Figure 1: Plan of the Piper Alpha Control Room

Competency requirements for an off-shore oil production facility are used to illustrate the remainder of this paper. These systems have posed a significant challenge for both systems designers and human factors specialists (Wardell,

1989). Oil production facilities are complex applications. This is illustrated by the plan of the Piper Alpha control room, shown in Figure 1.

Operators must monitor the extraction of oil from geological structures under the sea-bed. They must also control the extraction and purification of any gas that is recovered with the oil. They must monitor repair activities and maintenance schedules. They must also watch for problems that threaten the safety of the rig. For instance, gas leaks pose a considerable risk of fire. If gas is detected then control room personnel must investigate the cause and identify potential solutions. These tasks involve almost continuous interaction with computerised sensing and control systems.

It is important to emphasise that this paper focuses upon the operating rules and procedures that form a critical component of any operator's training with safety-critical computer systems. Brevity prevents a detailed discussion of the physical training that also play an important role during induction courses and competency tests. This topic is addressed in a companion paper (Johnson, 1996).

2. Representing Training Requirements

Companies that operate oil production facilities are faced with a difficult problem. On the one hand, regulatory authorities, such as the Health and Safety Executive, are urging them to document their training procedures. On the other hand, there are few techniques that can be used to develop such documentation. Natural language reports are unwieldy and quickly become intractable if they are used to represent many different training requirements for many different operators (Johnson, McCarthy and Wright, 1995). An additional problem is that training reports frequently describe the procedures that are used to instruct operators. They do not describe the objectives for that training. This makes it difficult to identify the criteria that are used to gauge the success or failure of that training. User models provide a solution to these problems. In particular, they provide structures that can be used to represent the cognitive objectives for an operator's training (Green and Benyon, 1995). For example, Duke, Barnard, Duce and May (1995) use the following clause to describe what should happen when an operator is trained to delete warnings from a safety-critical application. This approach exploits Barnard's Interacting Cognitive Subsystems model. A representation of the message is formed at the propositional level. This representation is then used to form the goal of deleting the message at an implicational level. Finally, the articulatory subsystem translates the goal into a process

that activates the musculature to move the hand and complete the necessary action. The training would have failed if the operator could not recognise the message, could not form the correct goal, or failed to produce the correct articulatory response:

$$\text{per(delete_message)} \Rightarrow \text{mesg@prop} \wedge \text{: prop-obj::obj-lim::lim-hand:} \propto \text{config} \quad (1)$$

This states that if the user can delete a message then they form a representation of the message at the proposition level, *mesg@prop*, and they form a goal to remove the object, *prop-obj*, this is translated through processes that activate their musculature, *obj-lim::lim-hand*, to complete the necessary action.

Practical problems prevent trainers from using this approach to represent training requirements. The level of detail in (1) would place heavy burdens upon any company that had to document training requirements for many different operators. The developers of this approach are aware of these issues and are currently developing tools to support the application of their user models (May, 1995).

3. Epistemics for Training Requirements

In anticipation of current research in user modelling, it is important that trainers have some alternative means of representing training requirements. Epistemology, or the study of knowledge, has a history stretching back to the Ancient Greeks (Fagin, Halpern, Moses and Vardi, 1995). This work has led to a number of epistemic logics that can be used to reason about the consistency and completeness of knowledge in areas such as Artificial Intelligence, Philosophy and Distributed Systems (Barwise and Perry, 1983).

Before showing how epistemics support the analysis of training requirements, it is first important to demonstrate the ways in which logic supports systems engineering. For example, the Piper Alpha accident was probably caused by a condensate injection pump being operated without a pressure safety valve. A systems requirement is, therefore, that a pump is started if it is protected by a safety valve and there is no permit to perform maintenance on the pump. This latter requirement is important because operators must not start a pump that is being repaired by their colleagues:

$$\text{safe_start(injection_pump)} \Leftarrow \text{working(ps_valve, injection_pump)} \wedge \text{not work_permit(injection_pump)}. \quad (2)$$

It is safe to start an injection pump if a pressurised safety valve is fitted and there is not a permit to perform maintenance work on that component.

This systems engineering constraint provides a focus for the development of training requirements. For example, the following clause uses the epistemic *Kn* operator (read as 'knows') to state that the user should only attempt to start a pump if they know that the safety valve is working and that there is no permit to work on that pump. This illustrates the close relationship between interface development and operator training. The human-machine interface must enable its users to start up the injection pumping equipment, *input(user, start_injection_pump)*. Training must be provided so that operators understand the conditions which must be satisfied if they are to issue this input. The following clauses exploit a first order variant of epistemic logic, the interested reader is directed to (Fagin, Halpern, Moses and Vardi, 1995) for more information on the formal under-pinnings of this notation. In contrast, the remainder of this paper focuses on the application of the logic:

$$\text{input(user, start_injection_pump)} \Leftrightarrow \text{Kn(user, working(ps_valve, injection_pump))} \wedge \text{Kn(user, not work_permit(injection_pump))} \quad (3)$$

A user provides input to start an injection pump if and only if they know that the safety valve for that pump is in working order and they know that there is no permit to work for that component.

Such clauses provide a precise and concise means of representing the competency requirements that must be satisfied by the users of complex systems. They represent the minimum standards that operators must satisfy in order to be certified for particular applications. For example, the previous requirement might be used in an examination by posing questions such as:

What checks should be made before you issue input to start up the condensate injection pumps?

An important benefit of clauses, such as (3), is that they strip away the clutter of implementation details that frequently obscure critical properties of safety-critical systems. Trainers and operators can identify high-level training objectives without being forced to consider low level device handling. For instance, permits to work can be revoked if maintenance has to be suspended for a short time. Under these circumstances, operators may start the equipment if they know that a suspension permit has been issued. Such high-level objectives can easily be lost amongst the many different output events and input sequences that must be considered during the later stages of training:

$$\begin{aligned} \text{input}(\text{user}, \text{start_injection_pump}) \Leftrightarrow \\ \text{Kn}(\text{user}, \text{working}(\text{ps_valve}, \text{injection_pump})) \wedge \\ (\text{Kn}(\text{user}, \text{not_work_permit}(\text{injection_pump})) \vee \\ \text{Kn}(\text{user}, \text{suspension}(\text{injection_pump}))) \end{aligned} \quad (4)$$

The user provides input to start an injection pump if and only if they know that the safety valve for that pump is in working order and they know that there is no permit to work for that component or they know that there is a suspension permit on the pump.

Section 6 will describe how presentation details and input requirements can gradually be introduced into clauses, such as (4), to represent competency requirements during the later stages of training. For now it is sufficient to realise that previous clauses refer to 'generic' users. They do not demonstrate that epistemic logic can be used to identify training requirements for particular operators performing particular tasks.

4. Training For Particular Tasks

Many oil production operators will never need to start up the injection pumping equipment. Toolpushers, divers and rig electricians are not expected to perform such tasks during their routine duties. Fortunately, the allocation of training requirements can be represented by changing the *user* marker in previous clauses to describe particular roles in the operation of the system. For example, the lead production operator, *lpo*, is responsible for starting up the condensate pumping equipment:

$$\begin{aligned} \text{input}(\text{lpo}, \text{start_injection_pump}) \Leftrightarrow \\ \text{Kn}(\text{lpo}, \text{working}(\text{ps_valve}, \text{injection_pump})) \wedge \\ \text{Kn}(\text{lpo}, \text{not_work_permit}(\text{injection_pump})) \end{aligned} \quad (5)$$

The lead production operator provides input to start an injection pump if and only if they know that the safety valve for that pump is in working order and they know that there is no permit to work for that component.

Additional clauses can be introduced to represent the training requirements that relate to other users. For example, it might be specified that the dive superintendent, *dsi*, should issue orders to start diving operations if they know that the diesel fire pumps have been switched from manual to automatic. This is a critical training requirement because fire-fighting pumps are fed with sea-water. If the pumps are activated while divers are working on the rig then they may be sucked towards the pump inlets. As before, it is important that trainers have some means of explicitly representing such requirements. Competent dive superintendents must check that the pumps are on manual if they allow divers to enter the water:

$$\begin{aligned} \text{input}(\text{dsi}, \text{start_diving}) \Leftrightarrow \\ \text{Kn}(\text{dsi}, \text{manual}(\text{diesel_fire_pumps})). \end{aligned} \quad (6)$$

The diving superintendent issues input to start diving operations if and only if they know that the diesel fire pumping systems have been turned from manual to automatic.

The previous clause will only be successful if operators can access the information that they have been trained to consider. For example, it would be little use training the dive superintendent to check on the state of the *diesel_fire_pumps* if they could not find out about the state of these components using the displays presented by the on-board information systems.

5. Training and Systems Engineering

It is important to emphasise that epistemic clauses, such as (5) and (6), do not guarantee the safety of an application. For instance, it is entirely possible for the diving superintendent to believe that the diesel pumps are on manual when they are, in fact on automatic. Similarly, the lead production operator might believe that the safety valve is operational when it is broken:

$\text{potential_accident} \Leftarrow$
 $\text{Kn}(lpo, \text{working}(ps_valve, \text{injection_pump})) \wedge$
 $\text{not working}(ps_valve, \text{injection_pump})$ (7)

There is the potential for an accident if the lead production operator knows that the safety valve is working and it is, in fact, the case that the valve is not working.

This clause represents a situation in which the lead production operator holds an erroneous belief about the state of their application. Systems engineers, trainers and interface designers must co-ordinate their activities to ensure the operator only believes that the pressure safety valve is operating if and only if that component is working:

$\text{Kn}(lpo, \text{working}(ps_valve, \text{injection_pump})) \Leftrightarrow$
 $\text{working}(ps_valve, \text{injection_pump})$ (8)

The lead production operator knows that the safety valve for that pump is in working order if and only if that component is, in fact, working.

Clause (8) represents a design objective that must be satisfied by the underlying systems engineering. Sensors must be deployed to accurately gauge the state of the valve. If these systems reported that the safety valve was working when it was broken then the operator might start the condensate pump. In such circumstances, the user's training might actually threaten the safety of the system. The operator simply follows the criteria laid down in clause (3). If, however, the systems engineers believed that the monitoring equipment might provide erroneous values then (3) might be altered so that operators were trained to perform additional checks.

6. Training and Interface Design

Norman (1990) argues that operator training must be supported by the development of 'appropriate' displays. The converse is also true. An interface is of little benefit if operators cannot be trained to exploit the information that it presents. The following clause demonstrates that epistemic logic can be used to represent the link between display objects and the information that operators have been trained to obtain before starting the injection pumps:

$\text{Kn}(lpo, \text{working}(ps_valve, \text{injection_pump})) \Leftrightarrow$
 $\text{present}(lpo, ps_valve_ok_icon)$ (9)

The lead production operator knows that the safety valve is in working order if and only if there is an icon to confirm this fact.

Clause (9) provides an extremely abstract description of an interactive system. This is an advantage because it does not force trainers to consider the low level details of particular presentation devices. Unfortunately, the logic provides little idea of the images that might be used to represent the valve icon. Logic can, however, be used to represent the structure and content of complex displays. For example, the following clauses describe the way in which the *condensate_display* might include the image of the *ps_valve_ok_icon*. This in turn includes the image of a feed adjuster. Finally, the feed adjuster is composed of the graphical primitives that are presented to the user:

$\text{part}(\text{condensate_display}, ps_valve_ok_icon)$ (10)

$\text{part}(ps_valve_ok_icon, \text{feed_adjuster})$ (11)

$\text{line}(\text{feed_adjuster}, 0.1, 0.2, 0.6, 0.2)$ (12)

The first clause states that the *ps_valve_ok_icon* is part of the *condensate_display*. The second clause states that a feed flow adjuster is, in turn, part of the *ps_valve_ok_icon*. Finally, the third clause states that the image of the feed flow icon contains a line from Cartesian co-ordinates (0.1,0.2) to (0.6, 0.2).

Figure 2 presents the structure of the image that is described by the previous clauses. The resulting graphical structures can be directly executed using the Prelog prototyping system. A detailed discussion of this approach is presented in (Johnson, 1995a).

The significance of the formulae (9,10,11,12) is that they explicitly represent the information which interface designers intend to communicate through particular images. The *ps_valve_ok_icon* tells the user that the injection valve is working, $\text{Kn}(lpo, \text{working}(ps_valve, \text{injection_pump}))$. These clauses provide a focus for evaluation. Lead production operators can be asked to describe the information that is represented by the valve icon. If the users fail to recognise that the valve is working then the image must be redesigned or operators must be trained to recognise the meaning of the existing image.

Figure 2: Image for the Condensate Display

7. Representing Training Violations

Previous sections have used epistemic logic to represent knowledge requirements for a safety-critical interface. The application of a formal notation does not, however, guarantee that operators will remember their training. For example, the lead production operator might attempt to start the pump with insufficient information about current maintenance activities. Under such circumstances, they would not know whether a permit to work had been issued, *not Kn(lpo, work_permit(injection_pump))*:

$$\begin{aligned} \text{training_failure} \Leftarrow & \\ & \text{input}(lpo, \text{start_injection_pump}) \wedge \\ & \text{Kn}(lpo, \text{working}(ps_valve, \text{injection_pump})) \wedge \\ & \text{not Kn}(lpo, \text{work_permit}(\text{injection_pump})) \end{aligned} \quad (13)$$

A training failure occurs if the lead production operator provides input to start an injection pump and they know that the safety valve for that pump is in working order but they do not know that there is no permit to work for that component.

It is important to emphasise the difference between clause (3) and clause (13). In the former case the lead production operator definitely knows that there is no permit to work, *Kn(lpo, not work_permit(injection_pump))*. In clause

(13), the operator does not know whether there is a permit, *not Kn(lpo, work_permit(injection_pump))*. These different situations can, in turn, be compared to one in which the user deliberately starts the pump even though they know that there is maintenance being performed on that component:

$$\begin{aligned} \text{deliberate_training_violation} \Leftarrow & \\ & \text{input}(lpo, \text{start_injection_pump}) \wedge \\ & \text{Kn}(lpo, \text{working}(ps_valve, \text{injection_pump})) \wedge \\ & \text{Kn}(lpo, \text{work_permit}(\text{injection_pump})) \end{aligned} \quad (14)$$

A lead production operator deliberately violates their training if they start an injection pump and they know that the safety valve for that pump is in working order and they know that there is no permit to work for that component.

Clauses (13) and (14) are violations of the training requirements represented in (3). Regulatory authorities would expect a formal review if the lead production operator attempted to start an injection pump that was currently being worked on. If the enquiry could demonstrate that the operator was simply negligent, as represented in (13), then the sanctions would be less severe than the deliberate neglect described in (14).

8. Conclusion And Further Work

This paper has demonstrated that epistemic logic can be used to represent training requirements in terms of the knowledge that operators must possess in order to safely perform particular tasks. This approach provides a more precise and concise means of representing competency criteria than the high level objectives that are frequently criticised in accident reports. An important benefit of our approach is that it can be directly integrated with current techniques for software engineering. This is vital if operator requirements are to be explicitly considered during the detailed development of safety-critical systems.

Our approach is limited in the sense that it focuses on knowledge requirements for safety-critical interfaces. We have not attempted to represent the physical skills that form an important component of many training programmes. Elsewhere we have described how logic can be used to reason about the physical layout of complex working environments (Johnson, 1996).

Future work will build closer links with task analysis and user modelling. For instance, epistemic logics can

be used to express the propositional and implicational information that is represented in the ICS model. Alternatively, the logic notation can be used to formalise the structured notations in Green and Benyon's (1995) ERMIA user modelling techniques.

Finally, we are concerned to improve the usability of the approach described in this paper (Gray and Johnson, 1995). Mathematical requirements provide non-formalists with a poor impression of what it would be like to interact with a safety-critical system. We are addressing this problem by extending the Prelog prototyping. This enables designers to directly execute the logic clauses that are presented in this paper. At any point during interaction, designers can ask the user to justify their reasons for a particular action. These can then be compared to the epistemic, *Kn*, clauses to determine whether the operator has taken into account all of the information requirements that were identified during their training.

Acknowledgements

Thanks go to the members of the Glasgow Accident Analysis Group and the Glasgow Interactive Systems Group (GIST) and to the Formal Methods and Theory Group in Glasgow University. This work is supported by the UK Engineering and Physical Sciences Research Council, grants GR/JO7686, GR/K69148 and GR/K55040.

References

- Air Accidents Investigations Branch, Department of Transport. *Report On The Accident To Boeing 737-400 G-OBME Near Kegworth, Leicestershire on 8th January 1989*, number 4/90, Her Majesty's Stationery Office, London, United Kingdom, 1990.
- J. Barwise and J. Perry. *Situations And Attitudes*. Bradford Books, Cambridge, United States of America 1983.
- Cullen, *Proceedings Of The Public Enquiry Into The Piper Alpha Disaster*. The Department of Energy, Her Majesty's Stationery Office, London, United Kingdom, 1990.
- Dix, A. *Formal Methods For Interactive Systems*, Academic Press, London, United Kingdom, 1991.
- D. Duke, P. Barnard, D. Duce and J. May. *Systematic development of the human interface*. In APSEC'95-Second Asia-Pacific Software Engineering Conference 1995.
- R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning About Knowledge*. MIT Press, Boston, United States of America 1995.
- D. Fennel, *Investigation Into The Kings' Cross Underground Fire*. Department of Transport, Her Majesty's Stationery Office, London, United Kingdom, 1988.
- C. Fidge, P. Kearney and J. Staples, *Formal Verified Real-Time Software: An Integrated Development Strategy*, Technical Report, Software Verification Research Centre, Department of Computer Science, University of Queensland, Australia, June 1993.
- P.D. Gray and C.W. Johnson, *A Critical Analysis Of Interface Specification Notations*. In P. Palanque and R. Bastide, editors, *The Specification And Verification Of Interactive Systems*, 113-133, Springer Verlag, Berlin, 1995.
- T.R.G. Green and D. Benyon, *Displays As Data Structures: Entity-Relationship Models of Information Artefacts*. In K. Nordby, P. Helmersen, D. Gilmore and S. Arnesen (eds.), *Interact'95*, 55-60, Chapman Hall, London, 1995.
- M.D. Harrison and H.W. Thimbleby, (eds.), *Formal Methods In Human Computer Interaction*, Cambridge University Press, Cambridge, United Kingdom, 1989.
- C.W. Johnson, *A Probabilistic Logic For The Development of Safety-Critical Interactive Systems*. *International Journal of Man-Machine Systems*, 39(2):333-351, 1993.
- C.W. Johnson, *The Formal Analysis Of Human-Computer Interaction During Accident Investigations*. In G. Cockton, S.W. Draper and G.R.S. Weir, editors, *People And Computers IX*, 285-300. Cambridge University Press, Cambridge, 1994.
- C.W. Johnson, *The Application of Petri Nets to Represent and Reason about Human Factors Problems During Accident Analyses*. In P. Palanque and R. Bastide, editors, *The Specification And Verification Of Interactive Systems*, 93-112, Springer Verlag, Berlin, 1995.

- C.W. Johnson, *Using Z to Support the Design of Interactive, Safety-Critical Systems* Software Engineering Journal, (10)2:49-60, 1995a.
- C.W. Johnson, *Decision Theory and Safety-Critical Interfaces*. In K. Nordby, P. Helmersen, D. Gilmore and S. Arnesen (eds.), Interact'95, 121-127, Chapman Hall, London, 1995b.
- C.W. Johnson, Impact of Working Environment upon Human-Machine Dialogues: A Formal Logic for the Integrated Specification of Physical and Cognitive Ergonomic Constraints on User Interface Design. *Ergonomics* (39)3:512-530, 1996.
- C.W. Johnson, J.C. McCarthy and P.C. Wright, *Using A Formal Language To Support Natural Language In Accident Reports*. *Ergonomics*, (38)6:1265-1283, 1995.
- P. Johnson, *Human Computer Interaction: Psychology, Task Analysis and Software Engineering*, McGraw-Hill, London, United Kingdom, 1992.
- J. May, Dual-task configuration in ICS: An ART* enterprise application. Technical report, MRC Applied Psychology Unit, Cambridge, United Kingdom. Reference EC Amodeus-2 technical report UM/WP34. 1995.
- Ministry Of Defence, Requirements for the Procurement of Safety Critical Software, MOD DEF-STAN 00-55, London, United Kingdom, 1991.
- W. Morehouse and M.A. Subamaniam, The Bhopal Tragedy. Technical Report. Council for International and Public Affairs, New York, United States of America, 1986.
- D. Norman, *The 'Problem' With Automation : Inappropriate Feedback And Interaction Not Over-automation*. In D.E. Broadbent and J. Reason and A. Baddeley (eds.), *Human Factors In Hazardous Situations*, 137-145, Clarendon Press, Oxford, United Kingdom, 1990.
- J. Reason, *Human Error*, Cambridge University Press, Cambridge, United Kingdom, 1990.
- United States Nuclear Regulatory Commission, Three Mile Island: A Report To The Commissioners and the Public. Technical report NUREG/CR-1250-V (The Rogovin Report), Washington D.C., United States of America, 1980.
- R.W. Wardell, *An Ergonomic Perspective On Safety In The Oilfield*. In D.L. Pettigrew, editor, *Proceedings of the 33rd Annual Meeting Of The Human Factors Society*, pages 999-1003. United States of America, 1989.