The Formal Analysis Of Human-Computer Interaction During Accident Investigations

Chris Johnson

The H.C.I. Group, The Department of Computer Science,
The University of York, Heslington, YO1 5DD, U.K.

Tel: +44 (0904) 433376
Fax: +44 (0904) 432767

EMail: johnson@minster.york.ac.uk

Many safety-critical applications rely upon complex interaction between computer systems and their users. When accidents occur, regulatory bodies are called upon to investigate the causes of user 'error' and system 'failure'. Reports are drawn up so that the designers and operators of future systems will not repeat previous 'mistakes'. These documents present the work of specialists who are drawn from many different technical disciplines: human factors; forensic investigation; engineering reconstruction; computer simulation; etc. The findings of these different experts are often separated into different sections. This creates a number of problems. Important evidence can be hidden within numerous appendices. The interaction between systems and users can be obscured by tortuous cross referencing schemes. There are occasional temporal ambiguities and inconsistencies between the different analyses. This paper presents ways in which formal methods can be exploited to address these problems. Mathematical notations provide means of representing and reasoning about the circumstances that lead to accidents in human machine systems. Executable logics can also be used to simulate event sequences. These simulations might be shown to other analysts. They can be used to encourage agreement on the course of events prior to more detailed investigations.

Keywords: ***safety-critical systems; temporal logics; formal methods; simulation.***

1 Introduction

Accident reports are intended to ensure that the faults of previous systems are not propagated into future applications. For example, the Presidential investigation into the Three Mile Island accident led the United States' Nuclear Regulatory Commission (NRC) to adopt a policy of minimal intervention [27]. Whenever possible operators should not be required to intervene in order to preserve the safety of their system. It was stipulated that the high-head charging pumps, part of the emergency cooling equipment, must be run for at least twenty minutes after reactor scrams. In 1979, changes in the generation process employed by the North Anna reactor led to dangerous temperature profiles following a scram [7]. The operators were faced with a difficult choice. If they obeyed NRC regulations then the safety of the plant would be threatened. If they disobeyed the regulations then the plant could be saved but they would break the NRC conditions of operation. Plant management chose to disregard the regulations; a pump was taken off the coolant circuit and the emergency was resolved. This incident underlines the importance of accident reports within the design and management of safety-critical systems. Regulatory authorities use them to direct the operation of interactive systems. It also illustrates the need to consider both human and system requirements during accident analysis. An initial response to the operator 'error' at Three Mile Island led the NRC to rely upon systems engineering. Users were forced to intervene when systems engineering failed to guarantee the safety of the North Anna reactor.

Accident reports present the findings of specialists who are drawn from many different technical disciplines: human factors; forensic investigation; engineering reconstruction; computer simulation; etc. These different strands of analysis are, typically, contained in different chapters. For instance, the U.K. Department of Transport's Air Accident Investigations Branch (AAIB) report into a near collision at Gatwick airport places meteorological and engineering details within the body of the document whilst the human factors analysis is placed in an appendix [1]. Reactor design and operator training are considered in different chapters of the Watt Committee's report into the Chernobyl accident [37]. This separation is useful in the sense that each section of a document reflects a particular perspective or focus. It also creates a number of problems. Many reports rely upon tortuous cross-referencing to draw together different strands of analysis. Others leave the reader wondering how the different findings of the experts can form a coherent picture of the events leading to an accident [19]. Further problems can be created by inconsistencies within the text of accident reports. The Kegworth investigation concluded that on-board systems failed to prevent pilots from shutting down a healthy engine [2]. In some places the engines are referred to by their numbers; 1 and 2. Elsewhere they are referred to as the left and right engines. Some of these inconsistencies arise from the verbatim reporting of witnesses when they describe the ways in which they interacted with their systems. Others have no such justification. This creates considerable confusion; the reader of the report is drawn into continual cross referencing to ensure that they understand the natural language description [17].

This paper argues that formal methods can be applied to address some of the limitations of conventional accident reports. Mathematical specifications are increasingly being recruited to support the engineering of complex applications. For example, both

NASA [25] and the European Space Agency [11] are applying formal notations to represent and reason about on-board communications systems. Formal analysis is a necessary part of many design projects commissioned by the United Kingdom's Ministry of Defence [22]. These techniques have also been applied to support interface development. Thimbleby has shown how finite state machines might be used to analyse interactive dialogues with a range of commercial applications [33]. Wright, Fields and Harrison shown how Hoare's Communicating Sequential Processes (CSP) notation can be used to describe interactive properties of safety-critical systems [38]. Previous work has built upon this research by showing that executable logics can be used to specify and prototype human-computer interfaces [16, 13, 14]. Formal notations provide a precise and concise means of representing traces of interaction. Executable subsets of mathematical specifications can then be used to derive partial implementations. These prototypes can be shown to system operators. They provide a medium of communication between the various members of concurrent design teams. Many of these benefits can also be extended to the analysis of accidents involving safety-critical systems. Logic can be used to represent the interaction that leads to a failure. These notations provide means of stripping out irrelevant detail which might otherwise obscure critical events during the course of an accident. Executable subsects of logic formalisms can also be used to derive simulations from abstract descriptions. This provides a means of ensuring that mathematical descriptions accurately capture the complex interactions that lead to accidents.

Much of the work described in this paper is in its early stages of development. Many questions remain unanswered. We do not address ways in which formal notations might be used to develop cognitive models of system operators. This is a weakness in our work; expertise and experience play an important role in the response to system failures. Further research intends to address this limitation. Our intention here is to demonstrate that formal techniques can be pragmatically applied to real-world problems.

1.1 The Outline Of This Paper

Section 2 briefly describes the events leading to the Chernobyl accident. This is used as an example for the remainder of the paper. Section 3 goes on to show how logic can be used to represent some of the factors which contributed to the failure in this human-machine system. Section 4 builds upon this and uses the same notation to specify requirements that must be met if the accident is not to be repeated in similar systems. Section 5 exploits the logic formalism to represent and reason about techniques that might be used to satisfy these requirements. It is argued that the application of formal methods does **not** remove the need for skilled analysis. It is perfectly possible to specify techniques that would threaten the safety of an interactive system. Section 6 describes means of identifying these potential threats. Executable subsets of the logic notation can be used to directly derive simulations of the events leading to an accident. The Prelog prototyping tool has been developed so that prototype displays can also be used to determine whether changes to human-machine interfaces might reduce the likelihood of future accidents. Section 7 presents the conclusions that can be drawn from this research. Areas for further work are also discussed.

2 An Example

At 01.23 hours on the 26th April 1986, Unit 4 of the Chernobyl Nuclear Power Station experienced a rapid, uncontrolled power release. This destroyed the reactor core and breached the containment buildings. Radioactive material was released into the environment for almost ten days. The events leading to this emission started with a test that was intended to determine whether the reactor could continue to supply power once it had begun to be run down. If successful, the pumps which supplied emergency coolant to the reactor could be powered by that reactor until diesel generators were started. Similar tests had failed on two previous occasions because the power level had fallen too rapidly. Operator intervention provided the immediate causes of the accident: they disabled the plant protection systems and made several errors in the way that they conducted the tests. The effects of this intervention were exacerbated by design flaws in the construction of RBMK reactors.

The events surrounding this accident are used to illustrate the remainder of this paper. This example is appropriate because it typifies many of the problems that frustrate investigations into human-machine failures. The causes were complex. They involved the interaction of operator 'error' and design problems. It has also been the subject of a number of international reports [35, 37]. These documents have been supplemented by analyses from a wide range of experts including medical authorities [23], meteorological researchers [26], human factors experts [29] and systems engineers [28]. A further reason for analysing the causes of this accident is that the TACIS '92 (Technical Assistance To The Commonwealth of Independent States) project is continuing to bring the human-machine systems in the CIS nuclear industry up to western standards. One of the aims of this initiative is to install computer based management information systems in RBMK reactors, similar to that at Chernobyl [8].

3 Informal Descriptions To Accident Analysis

Logic provides a precise and concise means of representing the informal description of the previous section. One of the contributory factors to the Chernobyl accident was an operator input error [29]. A 'hold power' order was not entered once the power reduction test had been initiated. This can be represented by the following clause. Intuitive readings for the logic will be given throughout the remainder of the paper:

$$dangerous_power_hold_omission :- \\ operator_input(initiate_power_reduction), \\ not(operator_input(power_hold)). \quad (1)$$

This states that a dangerous omission occurs if operators issue input to initiate a power reduction and they do not issue input to hold the power level.

The logic notation can also be used to represent system behaviour. In our example, the input omission led to a programmed power reduction below the target level of 700-1000 Megawatts (MW). The power fell to 30 MW. The RBMK reactors at Chernobyl were

known to be unstable below 700 MW:

$$dangerous_power_reduction :- \\ dangerous_power_hold_omission, mega_watt_power_level(30). \quad (2)$$

This states that a dangerous power reduction occurs if the operators initiates the power reduction and omits the power hold command and the power level is 30 MW.

The previous clause illustrates an important limitation of first order logic as a means of representing interactive systems. There is no notion of sequence. A *dangerous_power_reduction* would be true if the power level reached 30MW and then the operators issued input to initiate the power reduction. This does not accurately describe the course of events leading to the accident. The input was issued before the power fell below safe limits. Previous papers have described a number of different means of representing time within logic specifications of interactive systems [16, 12]. For instance, it is known that the operators of the Chernobyl plant had stabilised the power at 200 MW by 01.00 hours. This can be represented by the clause:

$$response_dangerous_power_reduction(0100) :- \\ operator_input(initiate_power_reduction, 2310), \\ mega_watt_power_level(30, 2349), \\ mega_watt_power_level(200, 0100). \quad (3)$$

This states that the operator responds to a dangerous power reduction if a command to initiate power reduction is issued at 2310 hours and the power level is 30 MW at 23.49 hours and by 01.00 hours the power is 200 MW. Additional parameters might be introduced to represent the date, month and year on which these propositions were true.

This approach addresses one of the major problems in existing accident reports. Temporal ambiguities often arise within the chapters produced by different teams of experts. For instance, the Watt report [37] into the Chernobyl report contains the following lines: 'The operator stabilised the power at 200 MW (thermal) by manual control at 01.00 on 26th April... By 01.23 the operator had apparently stabilised the reactor system at least temporarily. (p. 20)." The first sentence can be interpreted as meaning that the reactor stabilised at 01.00 hours. Doubt is created by the second sentence which appears several paragraphs later; manual intervention started at 01.00 hours and the reactor had not stabilised until 01.23 hours. This ambiguity illustrates the importance of identifying the actual sequence of events which leads towards an accident. The final recommendations of the Watt report embody the second interpretation; systems must present information about the state of the reactor during the delay between operator intervention and stabilisation. Other sections do not expand upon this point. It is not addressed in the chapter on reactor operation and training even though the delay would seem to be a significant stage during interaction with the system. The explicit parameterisation of clauses with time stamps, as in (3), can be used to avoid temporal

ambiguity. These clauses provide a partial ordering or time-line for the events leading to an accident. For instance, the previous quotation might be represented as follows:

$$\begin{aligned} &manual_response_to_power_reduction(0123) :- \\ &\quad operator_input(initiate_power_reduction, 2310), \\ &\quad mega_watt_power_level(30, 2349), \\ &\quad operator_input(manual_control_initiate, 0100), \\ &\quad mega_watt_power_level(200, 0123). \end{aligned} \tag{4}$$

This states that the operator responds to a dangerous power reduction if a command to initiate power reduction is issued at 2310 hours and the power level is 30 MW at 23.49 hours and at 01.00 hours operators issue input to initiate manual control of the reactor and by 01.23 hours the power is 200 MW.

This clause makes explicit the observation that manual control started at 01.00 hours and that the power had reached 200MW by 01.23 hours. This avoids the ambiguity that was present in the natural language description. Such clauses do not, however, provide a panacea for accident analysis. The Ukranian parliament has recent voted to cancel a 1991 resolution closing the remaining Chernobyl RBMK units. They will remain in operation until 2003 [4]. Safety requirements must be identified if previous faults are to be avoided in other systems.

4 Initial Analysis To Safety Requirements

Logic can be used to represent requirements that might be imposed upon interactive systems as a result of accident analysis. For instance, from (4) we know that the power level reached 30MW at 23.49 hours. The danger might have been averted if the operator had issued a command to hold the level at a time when the power was above the critical limit:

$$\begin{aligned} &halt_power_reduction(2330) :- \\ &\quad operator_input(initiate_power_reduction, 2310), \\ &\quad operator_input(power_hold, 2330), \\ &\quad mega_watt_power_level(700, 2330). \end{aligned} \tag{5}$$

This states that the operator responds to a dangerous power reduction if a command to initiate power reduction is issued at 23.10 hours and input is issued at 23.30 to hold the power level and the reactor was generating 700MW at 2330.

Such clauses cannot easily be used to guide the development and operation of future systems. They refer to specific events at particular moments during an accident. Operators might have responded to the power reduction by issuing the *power_hold* command at 23.31 hours, 23.32, 23.33 etc. Analysts would have to repeat clauses such as (5) to describe the effect of operator intervention at each of these points in time. This

problem can be avoided by the introduction of temporal logic operators. For example, the $\Diamond$ (read as 'eventually') operator can be defined for a formula w as follows: $\Diamond w \Leftrightarrow \exists t \geq 0, w|t = \ true$. In this definition $w|t$ is intended to represent the evaluation of w at time t. Manna and Pnueli present a more complete introduction to the syntax and semantics of this logic [21]. In contrast, the following clause shows how temporal logic operators can abstract away from the specific time points that were used to describe the accident scenario:

$$\begin{aligned} &eventual_halt_in_power_reduction :- \\ &\quad operator_input(initiate_power_reduction), \\ &\quad \Diamond(operator_input(power_hold), mega_watt_power_level(700)). \end{aligned} \tag{6}$$

This states that the operator responds to a dangerous power reduction if a command to initiate power reduction is issued and eventually input is issued to hold the power level and the reactor is generating 700MW.

The problem with this requirement is that the $\Diamond$ operator may not be sufficient to preserve the safety of the system. The requirement specifies that users must eventually stabilise the power at 700MW. It would be perfectly possible for the power level to dip below 700MW before the input is provided. Additional temporal operators can be exploited to avoid this limitation. The $\mathcal{U}$ (read as 'until') operator can be defined for formulae, w_1 and w_2, as follows: $w_1 \mathcal{U} w_2 \Leftrightarrow \ \forall t_1, \ 0 \leq t_1 < t_2, \ w_1|t_1 = \ true \ and \ w_2|t_2 = \ true$. Designers can specify that the power level should remain above the 700MW level until the operator issues the power hold command:

$$\begin{aligned} &user_response_to_dangerous_power_reduction :- \\ &\quad operator_input(initiate_power_reduction), \\ &\quad (mega_watt_power_level(Level), Level > 700) \\ &\quad \mathcal{U} \ operator_input(power_hold). \end{aligned} \tag{7}$$

This states that the operator responds to a dangerous power reduction if a command to initiate the reduction is issued and the power is greater than 700 MW until a command is issued to hold the power level.

Such requirements do not represent the final stage in accident analysis. These clauses must be used to guide the detailed design or re-design of similar applications if previous failures are not to be repeated. In other words, it must be possible to identify a range of techniques that might be employed to achieve safety requirements and break the chain of events leading towards failures.

5 Safety Requirements To Design Specifications

Systems engineering might be recruited to achieve the requirements specified by clauses such as (7). In the aftermath of the accident "arrangements were made to avoid the operation of the reactor below quarter power except during start-up and shutdown"

[37]. Automated systems can withdraw neutron absorbing control rods when reactor power falls below the 800MW threshold. This technique achieves the requirement of (7) by substituting *system_input* for *operator_input*:

$$system_response_to_dangerous_power_reduction :- \\ mega_watt_power_level(Level), Level < 800, \\ system_input(withdraw_control_rods). \quad (8)$$

This states that the system responds to a dangerous power level if the power is below 800 MW and the system withdraws the control rods.

It is important to emphasise that the use of formal methods does **not** remove the need for skilled analysis. The action of the automated system described in (8) would pose a significant threat to the safety of similar reactors. This point can be clarified by a description of the reactor physics which compounded the operator's input omission, described in (1). RBMK reactors have positive void coefficients. In other words, the water in the coolant system actually hinders the reaction. When the Chernobyl operators initiated the test they increased the flow of water to the coolant system. This was necessary in order to test the capacity of the pumping equipment, as described in Section 2. The additional water reduced the power of the reaction. Clause (4) described how the operator assumed manual control as the power level fell to 30MW. They attempted to stabilise the situation by withdrawing the neutron absorbing control rods, just as the system would have done in (8). This exacerbated the problem because once the control rods were re-introduced the coolant turned into steam. This created a high positive power coefficient as the dampening effect of the water was lost. In other words, the 'solution' described by (8) would have jeopardised the safety of the system. The use of the logic notation does not, therefore, guarantee the identification of safe and acceptable solutions. It does, however, provide a precise means of representing design alternatives. For instance, the solution described in (8) might be rejected in favour of a protection system that prevented the control rods from being withdrawn. Systems engineers [37] have suggested that the accident might have been avoided if operators had been faced with interlocks and additional protection systems:

$$system_avoids_power_coefficients :- \\ mega_watt_power_level(Level), Level < 800, \\ (\neg operator_input(withdraw_control_rods), \\ (\neg system_input(withdraw_control_rods))\mathcal{U} \\ (mega_watt_power_level(Level), Level > 1600). \quad (9)$$

This states that the system avoids dangerous power coefficients if the power is below 800 MW and input to withdraw the control rods is blocked until the power is more than 1600MW.

It is important to consider the impact which such systems engineering solutions might have upon the usability of a control system. The previous clause specifies that opera-

tors must not issue input to *withdraw_control_rods* until the power level rises. Such delays can lead to frustration and error [20]. Unpredictable behaviour is likely to occur if the system were to process a backlog of delayed input [9]. Delayed commands might take effect at inappropriate moments during interaction. The presentation of a large amount of contextual information is required before a user can resolve such instances of unpredictability. Logic can also be used to represent such presentation requirements. Warnings could be issued to inform the operator of the underlying behaviour of the system. A *rod_lock_icon* might warn users that the rods could not be withdrawn:

$$
\begin{aligned}
&system_avoids_and_presents_power_coefficients :- \\
&\quad mega_watt_power_level(Level), Level < 800, \\
&\quad (display(low_power_warning), part(low_power_warning, rod_lock_icon), \\
&\quad \neg operator_input(withdraw_control_rods))\ \mathcal{U} \\
&\quad (mega_watt_power_level(Level), Level > 1600, display(power_level_reolved), \\
&\quad part(power_level_reolved, rod_withdraw_initiated)). \qquad (10)
\end{aligned}
$$

This states that the system avoids dangerous power coefficients if the power is below 800 MW and input to withdraw the control rods is blocked and a rod lock warning is part of a low power display until the power more than 1600MW and a rod withdrawal icon is part of a display indicating that power has stabilised.

Previous sections have argued that a logic notation can be used to reason about the events leading to accidents in human-machine systems. This notation provides a means of representing design objectives for future systems. Logic can also be used to reason about alternative means of achieving these safety requirements. It is important to emphasise, however, that the formal representation of interactive techniques, such as that described by (10), will not guarantee the safety of an application. For instance, inattention and fatigue may prevent operators from observing error messages such as *rod_lock_icon*. Executable subsets of temporal logics provide a means of detecting these problems. Simulations and prototypes can be directly developed from clauses such as (10). They can be shown to the different parties involved in accident investigations. They can also be shown to the potential operators of safety-critical systems. They are amenable to experimental analysis. They can be used to determine the cognitive, perceptual, physiological and sociological demands that a control system might place upon its users.

6 Simulation

Design specifications, expressed in formal notations, provide the non-formalist with little idea of what it would be like to interact with a system. Prototypes provide a far better impression of the look and feel of a final implementation. Several research groups are investigating ways of deriving partial implementations from temporal logic specifications [24]. We have developed Prelog. This prototyping tool integrates the Tokio temporal logic meta-interpreter [10] and the Presenter screen management system [34]. This provides a means of directly introducing presentation details into clauses such as

(10). For instance, the *rod_lock_icon* might be decomposed into a number of graphical primitives. The resulting logic specifications can be exploited to drive prototype implementations using Prelog:

$$dimension(rod_lock_icon, 0.02, 0.02). \quad (11)$$
$$pattern(rod_lock_icon, blank). \quad (12)$$
$$line(rod_lock_icon, 0.2, 0.3, 0.4, 0.33). \quad (13)$$

This states that the *rod_lock_icon* is 0.02 by 0.02 units in dimension; these units are calculated relative to the size of the *low_power_warning* display. The background of the image is blank. Its image includes a line from (0.2, 0.3) to (0.4, 0.33).

Prelog is being developed in two directions. The first relies upon the use of probabilistic data to simulate the behaviour of stochastic applications. For instance, operator commands to start pumping will not always be successful if pumps occasionally fail. Our approach is appropriate because temporal frequencies lie at the heart of reliability analysis. Failure rates can be explicitly represented using extensions to existing temporal logics. Figures from Associated Octel predict 76 failures per operating year for compressors [5]. Designers can specify that the probability of a particular compressor failing in the next second is 0.964×10^{-6}:

$$failure_rate(compressor_1, 0.000000964) \quad (14)$$

The probability of compressor 1 failing in the next second is 0.964×10^{-6}.

Such failure rates have important implications for interface development. Wickens argues that users devise polling strategies to exploit the probabilistic behaviour of their system [36]. Less attention will be allocated to the observation of low probability errors, greater attention is allocated to the detection of high probability errors. Prelog has already been enhanced so that designers can use probabilistic clauses to simulate the mix of high and low probability failures which characterise accidents in complex systems [15]. In our example, the probability of *compressor_1* failing, 0.964×10^{-6}, is compared with a random number in the range [0.0, 1.0]. If the random number is less than or equal to the probability then the fact is assumed to be true; the pump fails. If the random number is greater than the probability then the fact is assumed to be false; the pump does not fail. Current work is exploring means of introducing utility measures into our system. Low probability, high cost failures, typically, have a greater impact upon safety than high probability, low cost errors.

The second area of research into the simulation of interactive, safety-critical systems focuses upon the introduction of environmental information. Accidents often occur because important information is hidden from the normal line of sight of system operators [3]. The design of effective displays is of little value if users cannot see the devices which present them. Logic can be used to represent the position of a control panel within a working environment. Prelog is being extended so that three dimensional models of control rooms can be animated to show the layout of potential displays and input devices:

$$location(feedwater_panel, 6.7, 8.8). \quad (15)$$
$$dimension(feedwater_panel, 1.5, 0.5, 1.5). \quad (16)$$
$$present(low_power_warning, feedwater_panel) \quad (17)$$

The feedwater control panel is located at (6.7, 8.8) and has dimensions 1.5 by 0.5 by 1.5 meters and presents the $low_power_warning$ display.

This information can be used to reason about the allocation of data to presentation devices. High priority information and routinely monitored displays must be placed in locations that fall within the normal visual angle of system operators. Less important information may be placed on peripheral presentation devices. It is important to explicitly represent this allocation because these details are often neglected. For instance, the Electric Power Research Institute's review of nuclear power control rooms found meters that could not be read from normal working positions [31]. Critical displays were presented on the reverse of panels whose primary displays were devoted to non-critical information. Meters were obscured by the normal working positions of other operators.

7 Conclusion

This paper has argued that formal notations provide a precise means of representing the events that lead to accidents involving interactive systems. Temporal logics can be used to avoid some of the ambiguity that arises in natural language descriptions of interaction. A further advantage is that the same notation provides a means of specifying design requirements which are intended to preserve the safety of similar systems. Finally, executable subsets of the logic notation can be used to derive partial implementations of potential interfaces. These simulations can be shown to other investigators. They can be used to encourage agreement on the course of events prior to more detailed investigation. They can also be used to assess the utility of potential changes to human-machine interfaces.

Previous sections have not attempted to draw out general principles from the events leading to the Chernobyl accident. This is justified because accident reports are, typically, pitched at a relatively low level of detail. Other authors have used formal methods to describe generic techniques that can guide the development of many different interfaces [6, 32]. Further work intends to apply this approach to a number of accident reports. For instance, substitution error was a contributory factor to both the Kegworth [2] and Three Mile Island [18] disasters.

Further work needs to provide methodological support for the notations and tools described in this paper. The translation of accident observations into logic clauses and the derivation of requirements relies heavily upon the skills of the analyst. As noted in Section 5, it is perfectly possible to produce specifications that threaten the safety of an application. It has been argued that the development of simulations can help to identify such problems. One means of doing this might be to get analysts to "think aloud" as a Prelog prototype simulates the events leading to an accident. This would

build upon the 'story boarding' techniques that are already widely used during accident investigations.

Section 1 emphasised that our approach is in the early stages of development. It can be argued that the logic only provides a syntactic description. It does not capture the causes of particular traces of interaction. For instance, the formal notation has not been used to develop cognitive models of system operators. It would take a significant amount of work to resolve such limitations. For instance, Rouse reviews a number of techniques which have been used to model system operators [30]. Few, if any, of these can be applied to represent and reason about group cognition in multi-user applications. This is a research area in its own right. In anticipation of the products of such work, the scope of our approach has been more limited. The complexity of interactive, safety-critical systems has created an urgent requirement for pragmatic techniques that can represent the complex failures which lead to loss of life and environmental destruction.

8 Acknowledgements

This work was built upon initial research conducted into Petri net representations of aircraft accidents with Dr John McCarthy, Department of Applied Psychology, University College Cork, Ireland and Dr Peter Wright, British Aerospace's Dependable Computing Systems Centre, University of York, York, United Kingdom. Thanks must also go to the other members of the Human Computer Interaction Group, University of York. This work is supported in part by Science and Engineering Research Council grant GR/J07686.

References

[1] Air Accidents Investigations Branch, Department of Transport. *Report On The Incident Involving BAC 1-11 G-AYWB and Boeing 737 EI-BTZ On 12 April 1988 At Gatwick Airport*, number 2/89, London, United Kingdom, 1989. Her Majesty's Stationery Office.

[2] 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, London, United Kingdom, 1990. Her Majesty's Stationery Office.

[3] V. Bignell and J. Fortune. *Understanding System Failure.* Manchester University Press, Manchester, United Kingdom, 1991.

[4] A. Cruickshank. Chernobyl stays on-line. *ATOM*, (431):2, 1993.

[5] K.R. Davies. Techniques for the identification and assessment of major accident hazards. In J. Burgoyne, editor, *The Assessment And Control Of Major Hazards*, pages 289–308. Pergamon Press, Oxford, United Kingdom, 1985.

[6] A.J. Dix. *Formal Methods For Interactive Systems.* Academic Press, London, United Kingdom, 1991.

[7] K.D. Duncan. Fault diagnosis training for advanced continuous process installations. In J. Rasmussen, K. Duncan, and J. Leplatt, editors, *New Technology And Human Error*, pages 209 – 221. J. Wiley and Sons, New York, United States of America, 1987.

[8] J. Dynan and A. Francis. Smolensk gets GOMIS. *ATOM*, (431):28–29, 1993.

[9] C.A. Ellis and S.J. Gibbs. Concurrency control in groupware systems. *ACM SIGMOD Record*, 18(2):399–407, 1989.

[10] M. Fujita, S. Kono, H. Tanaka, and T. Moto-Aka. Tokio: Logic programming based on temporal logic and its compilation to PROLOG. In E. Shapiro, editor, *Third International Conference On Logic Programming*, LNCS 225, pages 695–708. Springer-Verlag, Berlin, FDR, 1986.

[11] M. Gamble and C.R. Taylor. The CCSDS protocol validation programme: Interagency testing using LOTOS. In J. Quemada, J. Mañas, and E. Vazquez, editors, *Formal Description Techniques III*, pages 319–326. Elsevier, North Holland, 1991.

[12] C.W. Johnson. Applying temporal logic to support the specification and prototyping of concurrent multi-user interfaces. In D. Diaper and N. Hammond, editors, *People And Computers VI: Usability Now*, pages 145–156. Cambridge University Press, Cambridge, United Kingdom, 1991.

[13] C.W. Johnson. A formal approach to the presentation of CSCW systems. In J.L. Alty, D. Diaper, and S. Guest, editors, *People And Computers VIII*, pages 335–352. Cambridge University Press, Cambridge, United Kingdom, 1993.

[14] C.W. Johnson. A probabilistic logic for the development of safety-critical interactive systems. *International Journal Of Man-Machine Studies*, 39(2):333–351, 1993.

[15] C.W. Johnson. Specifying and prototyping dynamic human-computer interfaces for stochastic applications. In J.L. Alty, D. Diaper, and S. Guest, editors, *People And Computers VIII*, pages 233–248. Cambridge University Press, Cambridge, United Kingdom, 1993.

[16] C.W. Johnson and M.D. Harrison. Using temporal logic to support the specification and prototyping of interactive control systems. *International Journal Of Man-Machine Studies*, 36:357–385, 1992.

[17] C.W. Johnson, J.C. McCarthy, and P.C. Wright. Using Petri nets to support natural language in accident reports. *To Appear In Ergonomics*, 1994.

[18] M. Kirkpatrick and K. Mallory. Substitution error potential in nuclear power plant control rooms. In R.C. Sugarman, editor, *Proceedings Of The 25th Annual Meeting Of The Human Factors Society*, pages 163–167. Human Factors Society, Santa Monica, United States of America, 1981.

[19] T.A. Kletz. *What Went Wrong? Case Histories Of Process Plant Disasters*. Gulf, Houston, United States Of America, 1985.

[20] W. Kuhmann. Stress inducing properties of system response times. *Ergonomics*, 32(3):271 – 280, 1989.

[21] Z. Manna and A. Pnueli. Verification of concurrent programs: Temporal proof principles. In D. Kozen, editor, *Logic Of Programs 1981 - Proceedings*, LNCS 131, pages 200–252. Springer-Verlag, Berlin, FDR, 1982.

[22] Ministry Of Defence. *Requirements for the Procurement of Safety Critical Software*, MOD DEF-STAN 0055, London, United Kingdom, 1991.

[23] M. Morrey, J. Brown, J. Williams, M. Crick, J. Simmonds, and M. Hill. A preliminary assessment of the radiological impact of the Chernobyl reactor accident on the population of the European Community. Technical report, National Radiological Protection Board, United Kingdom, 1986. CEC Contract 96-398.

[24] B. Moszkowski. *Executing Temporal Logic Programs*. Cambridge University Press, Cambridge, United Kingdom, 1986.

[25] National Aeronautic and Space Administration. *Advanced Orbiting Systems - Architectural Specification For The CCSDS Secretariat*, Washington DC, United States of America, 1989.

[26] C. Persson, H. Rodhe, and L. De Geer. The Chernobyl accident - a meteorological analysis of how radionuclides reached and were depositied in Sweden. *Ambio*, 1(16), 1986.

[27] R.M. Pew, D.C. Miller, and C.E. Feehrer. Evaluating nuclear control room improvements through analysis of critical operator decisions. In R.C. Sugarman, editor, *Proceedings Of The 25th Annual Meeting Of The Human Factors Society*, pages 100–104. Human Factors Society, Santa Monica, United States of America, 1981.

[28] P. Potter. The design of the Chernobyl Unit 4 reactor. In N. Worley and J. Lewins, editors, *The Chernobyl Accident And Its Implications For The United Kingdom - Report Number 19 Of The Watt Committee On Energy*, pages 9–18. Elsevier Applied Science, London, United Kingdom, 1988.

[29] J. Reason. *Human Error*. Cambridge University Press, Cambridge, United Kingdom, 1990.

[30] W.B. Rouse. Human-computer interaction in the control of dynamic systems. *ACM Computing Surveys*, 13(1), 1981.

[31] J.L. Seminara, W.R. Gonzalez, and S.O. Parsons. Human factors review of nuclear power plant control room design. Technical Report RI-NP-1977-309, Electronic Power Research Institute and Lockheed Missiles And Space Company, Palo Alto, United States of America, 1977.

[32] H.W. Thimbleby. *User Interface Design.* Addison Wesley, Wokingham, United Kingdom, 1990.

[33] H.W. Thimbleby. Combining systems and manuals. In J.L. Alty, D. Diaper, and S. Guest, editors, *People And Computers VIII*, pages 479–488. Cambridge University Press, Cambridge, United Kingdom, 1993.

[34] R. Took. Integrating inheritance and composition in an objective presentation model for multiple media. In F.H. Post and W. Barth, editors, *EUROGRAPHICS '91*, pages 291–303. Elsevier Science Publications, North Holland, Netherlands, 1991.

[35] The accident at the Chernobyl nuclear power plant and its consequence. Technical report, USSR State Committee On The Utilisation Of Atomic Energy, Vienna, Austria, 1986.

[36] C.D. Wickens. *Engineering Psychology And Human Performance.* C.E. Merrill Publishing Company, London, United Kingdom, 1984.

[37] N. Worley and J. Lewins, editors. *The Chernobyl Accident And Its Implications For The United Kingdom - Report Number 19 Of The Watt Committee on Energy.* Elsevier Applied Science, London, United Kingdom, 1988.

[38] P. Wright, B. Fields, and M. Harrison. Deriving human-error tolerance requirements from task analysis. In *Proceedings Of The International Conference On Requirements Engineering*, New York, United States of America, 1994. IEEE Press.