The Application Of Petri Nets To Represent And Reason About Human Factors Problems During Accident Analyses

Chris Johnson

Department of Computing Science, University of Glasgow,
Glasgow, G12 8QQ.
johnson@dcs.glasgow.ac.uk

Abstract Accident reports are intended to ensure that failures do not recur. They contain the analysis of many different experts, including human factors and systems engineers. The insights of these investigators are often separated into chapters that reflect the particular concerns and expertise of their authors. Such a separation often makes it difficult for readers to trace the ways in which human and system 'failures' combine to create the necessary conditions for an accident. The following paper argues that mathematically based modelling techniques can be used to overcome this problem. It is hypothesised that the application of formal notations can be extended from the domain of systems engineering in order to represent the findings of human factors analyses. In particular, it is argued that Petri Nets can be used to represent and reason about the concurrent behaviour of multiple operators and their systems. Tool support can be recruited to validate the resulting nets. The sequences of events leading to an accident can be simulated and shown to human factors and systems engineers. This, in turn, may elicit further observations about the causes of an accident. A near collision analysed by the U.K. Department of Transport's Air Accident Investigations Branch (AAIB) is used in order to evaluate this approach.

1 Introduction

We are concerned to identify techniques that support both human factors and systems engineering because the Commission of the European Community [12], the Japanese Fifth Generation Initiative [28] and United States' Presidential Task Forces [25] have all cited operator intervention as a primary factor in the cause and exacerbation of accidents. Unfortunately, many accident reports separate human factors considerations from the findings of systems engineers. They are frequently discussed in different chapters and appendices, for example see Air Accidents Investigations Branch [1, 2] or Worley and Lewis [29]. This paper starts from the premise that such distinctions are artificial. Human 'errors' cannot be fully explained without considering the demands that application functionality places upon its operators. System failures cannot be fully explained without considering the managerial and operational circumstances that affect maintenance and reliability. Some accident reports use tortuous cross-referencing to draw together these 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]. The following paper intends to show how mathematically-based notations can be used to avoid some of these problems. These notations provide means of capturing the concurrent behaviour of multiple operators and their systems. Analytical techniques can then be applied in order to identify ways in which the interaction

between operators and systems jeopardises safety. It is argued that tool support can be recruited to validate the findings embodied in Petri Nets. The sequences of events leading to an accident can be simulated. These animations can be shown to human factors and systems engineers in order to elicit further observations about the causes of an accident.

1.1 Alternative Formalisms

A number of alternative formalisms can be recruited to support human factors and systems engineering. For instance, the tabular form of Failure Mode and Effect Analysis can be used to represent the conditions that lead to an accident [9]. Failure modes, the effects of systems failures and additional observations are enumerated under three columns, this is illustrated in Table 1. These tables could include both operator and

Component	Failure Mode	Failure Cause	Criticality	Improvement
Pump Blow-back	Blockage	Valve collapse	Medium	Maintenance
Pump Seal	Over-heat	Dried	High	More inspection

Table 1: A Failure Mode, Effects And Criticality Analysis Table.

system 'failures'. Unfortunately, they can become large and unwieldy even for accidents with relatively simple causes. Interaction between failures in different parts of the system are not considered. These are significant limitations for analysts seeking to represent interaction between systems and their operators.

Fault-trees can be used to represent the findings of accident investigations. These use 'gates' to represent the logical combinations of events that must occur in order for an accident to take place. The fault-tree notation is analogous to that used to represent electrical circuits. It is in widespread use throughout the process industries. Human factors researchers have used similar notations to support the design of interactive systems [27]. Unfortunately, the European Federation of Chemical Engineering's International Study Group On Risk Analysis concludes:

> "Fault-trees have difficulties with event sequences... parts of systems where sequence is important are, therefore, usually modelled using techniques more adept at incorporating such considerations" (Cox, [8]).

This is a significant limitation because event sequences determine the quality of interaction between an operator and their system. The order in which displays are presented and commands issued is an important consideration during the development of human-machine interfaces.

Elsewhere we have argued that traces of interaction can be represented using interval temporal logics [15, 17]. These formalisms extend first order logics to provide operators, such as $\bigcirc$ (read as 'next'), that can be used to represent sequential information. Unfortunately, these notations provide non-formalists with an extremely poor impression of the human-machine interaction leading to an accident. Executable subsets of interval temporal logic can, however, be used to derive prototypes that simulate the behaviour of interactive systems [13]. These simulations can be shown to analysts in order

to determine whether the logic accurately models the events that led to an accident. Unfortunately, using temporal logics to construct multi-user, multi-system simulations can be extremely costly in terms of the time and expertise of the analyst.

The following pages intend to demonstrate that Petri Nets can be used to overcome the limitations of formalisms such as fault-trees and executable logics. Some notable attempts have been made to represent human factors requirements using this notation. For instance, Van Biljon [5] exploits Petri Nets to derive formal specifications of interactive systems at a very high level of abstraction. Bastide and Palanque [3] have used this notation to represent the design of an interactive database. Petri Nets have also been used to support the engineering of concurrent systems [20]. Chretienne [7] shows how they can be used to represent and reason about timing properties of system events. Hura and Attwood [11] have used Petri Nets to support accident analysis from the perspective of systems engineering. In contrast, this paper will demonstrate that Petri Nets can be used to analyse the operator-system interaction that can lead to accidents.

1.2 Outline Of The Paper

This section has briefly provided the background to this research and has introduced the main objectives of the paper. Section 2 describes the events that led to a possible collision between two aircraft. The AAIB accident report on this incident is used to illustrate our argument. Section 3 introduces the components of Petri Nets and argues that they can be used to represent the ways in which human and systems 'failures' can contribute towards an accident. Section 4 argues that graphical representations of Petri Nets provide a useful means of describing the event sequences that can lead to loss of life. Section 5 shows how this graphical notation can be used to represent the concurrent behaviour of multiple concurrent systems and their users. It is argued that this provides a means of capturing the many different interactions that frequently lead to accidents. Section 6 argues that Petri Nets can be used to identify potential instances of non-determinism, or conflict, within human-machine systems. The results of this analysis can be used to direct the allocation of design resources in order to avoid the circumstances that lead to accidents. Section 7 shows how hierarchical Petri Nets can be used to capture the more detailed findings of human factors and systems specialists. Sub-networks can represent the fine grained analysis of experts such as cognitive psychologists and avionics designers. It is argued that higher-level Petri Nets provide a cohesive structure, or context, for these more detailed findings. Section 8 briefly introduces Chiola's GreatSPN tool that has been used to produce and validate the networks presented in this paper [6]. Section 9 describes areas for further work and presents the conclusions that may be drawn from this research.

2 The Scenario

We have selected an incident analysed by the AAIB as an example for the remander of this paper [1]. A British Aerospace One-Eleven (BA-1-11) was returning to the United Kingdom from Venice. Its commander sat in the right seat. In the left seat, a first officer was undergoing familiarisation training. As they reached Abbeville, the crew discussed their approach to runway 08L at Gatwick Airport. This was known as the emergency runway. It was held in reserve for situations in which the main

runway, 08R, was unavailable. Both pilots were familiar with 08L. Both knew that it would come into operation at 21:00 hrs when the main runway was scheduled to be closed down. At 20:45 hrs they received an Automatic Terminal Information Service radio transmission from their destination. This gave weather details and reported that the current landing runway was 08R. Shortly afterwards, London air traffic control confirmed their clearance for 08L. They also informed the crew that this runway was being opened for them. The crew were cleared by the Tower for a visual landing on "08 LEFT". They replied with "CLEAR TO LAND 08 LEFT". There was then a brief discussion initiated by the co-pilot who was uncertain which runway they were approaching. This caused the commander to change his interpretation of the visual cues on his approach to the airfield. He re-aligned the aircraft with taxiway number 2 instead of the landing runway 08L. The aircraft touched the ground and rolled to a halt on the taxiway after 960 meters.

While the BA-1-11 approached the airfield, the crew of a Boeing 737 (B-737) were preparing to start their engines. They were directed along taxiways 7 and 2. As they turned onto taxiway 2, the crew of the B-737 observed the lights of the BA-1-11 as it came in to land. After a few seconds discussion with his co-pilot, the commander slowed the B-737 and flashed its landing lights. He then made a call to the Control Tower. This call was not acknowledged. The air traffic controller had noticed the BA-1-11's change of course and was busy confirming his observations using the Ground Movement Radar. This system did not extend beyond the airfield's boundary. The controller, therefore, had to wait until the BA-1-11 approached the taxiway before he could gain the necessary confirmation. The crew of the B-737 were told to hold their position as the BA-1-11 approached. Three seconds later the Tower ordered them to move off the taxiway. The fully laden aircraft turned onto the grass verge and its tires became bogged-down in the grass. The tail and left wing of the B-737 were left partially blocking the taxiway. The BA-1-11 stopped one hundred and ninety meters short of the B-737.

3 Petri Nets

Petri Nets can be used to represent the state of a human-machine system in terms of the conditions that are satisfied during intervals of interaction [24]. When particular conditions hold, certain events can occur. If an event takes place then it can alter the state of the human-machine system. Changes in state are represented by the new conditions that hold after an event has occurred. These new conditions enable further events to take place. More formally, Petri Nets are directed graphs; $PN = (P, T, E, M)$. They consist of a set of places, P, transitions, T, edges, E and markings, M.

Places can be used to describe the conditions that hold for operators and their systems during the course of an accident. In particular, they can be used to represent the pre-conditions for disaster. In our scenario, accident analysts might use a place to represent the fact that the BA-1-11 is on taxiway 2. Another place can represent the fact that the B-737 is blocking the path of the BA-1-11. Such places describe the causes of an accident at an extremely high level of abstraction. Places can also be used to represent causes that are specifically related to the human factors or systems engineering of an application. Places can be used to represent human factors observations about the behaviour of individual operators; the air traffic controller is monitoring the Ground Movement Radar. Places might also represent the behaviour of individual systems; the Ground Movement Radar is presenting the new course of the BA-1-11.

Transitions can be used to represent the events that cause accidents. In our scenario, the initiating events leading towards a collision can be identified as the landing of the BA-1-11 and the arrival of the B-737 on taxiway 2. These can be represented as transitions that change the state of the human-machine system into one in which both aircraft are on taxiway 2. Transitions can represent events initiated by both systems and their operators. For instance, human factors observations might indicate that the air traffic controller's decision to move the B-737 was a critical transition in the human-machine system. Similarly, systems engineers might identify the detection of the BA-1-11 by the Ground Movement Radar as an important event in our accident scenario. Isolating these critical transitions provides a focus for subsequent human factors and systems analysis. The air traffic controller might have been trained not to move the B-737 off onto the grass. The Ground Movement Radar might have been re-designed to detect the change in course before the BA-1-11 reached taxiway 2.

Edges connect places to transitions: $E \subseteq \{P \times T\} \cup \{T \times P\}$. They can be used to form the chains of events and conditions that lead to an accident. They can be described in terms of two functions. The function Op maps from each transition to its set of output places. The output places of a transition represent the conditions that hold after an event has occurred. For example, an output place can be used to represent the systems engineering observation that changes in the BA-1-11's approach caused changes to the display of the Ground Movement Radar. An input place function, Ip, maps from each transition to the set of input places for that transition. The input places of a transitions specify the conditions that must hold for an event to occur. The input place of a transition can be used to represent the human factors observation that the Ground Movement Radar's display led the air traffic controller to order the B-737 off taxiway 2.

4 Graphical Notation

Petri Nets can be represented graphically. Events, or transitions, are shown as bars (–). Conditions, or places, are denoted by unfilled circles (◯). Edges are shown as arrows linking places and transitions. Figure 1 shows a graphical Petri Net representation for our accident scenario. Rather than show the path of events that actually occurred, this network shows how the incidents reported by the AAIB could have led to a loss of life. Such hypotheses frequently form the focus of accident investigations in incidents where casualties were 'narrowly' avoided. Figure 1 provides a very high level representation of the events leading to a potential collision. It is intended to show how Petri Nets can provide a framework for subsequent human factors and systems analysis. This high level of abstraction is inappropriate for many of the findings presented in accident reports. Section 7 will show how Petri Nets can be refined to represent a finer level of detail.

In order to model the dynamic behaviour of a human-machine system, tokens are used to mark those places in a Petri Net that are enabled. A place is enabled if its conditions hold. The tokens in a net are said to characterise a marking state and are denoted graphically by filled dots (•). For instance, Figure 2 is marked to show that both the BA-1-11 and the B-737 are on taxiway 2. Analysts can alter the marking of a Petri Net to indicate the different conditions that hold for operators and their systems. These walk-throughs can be used to simulate the sequences of events and states that arise during accident scenarios. A transition can fire if all of its input places contain at least one token. After firing, a token is deposited in each of the output places of a transition. In Figure 2 it is possible for the transition indicating that BA-1-11 establishes a collision

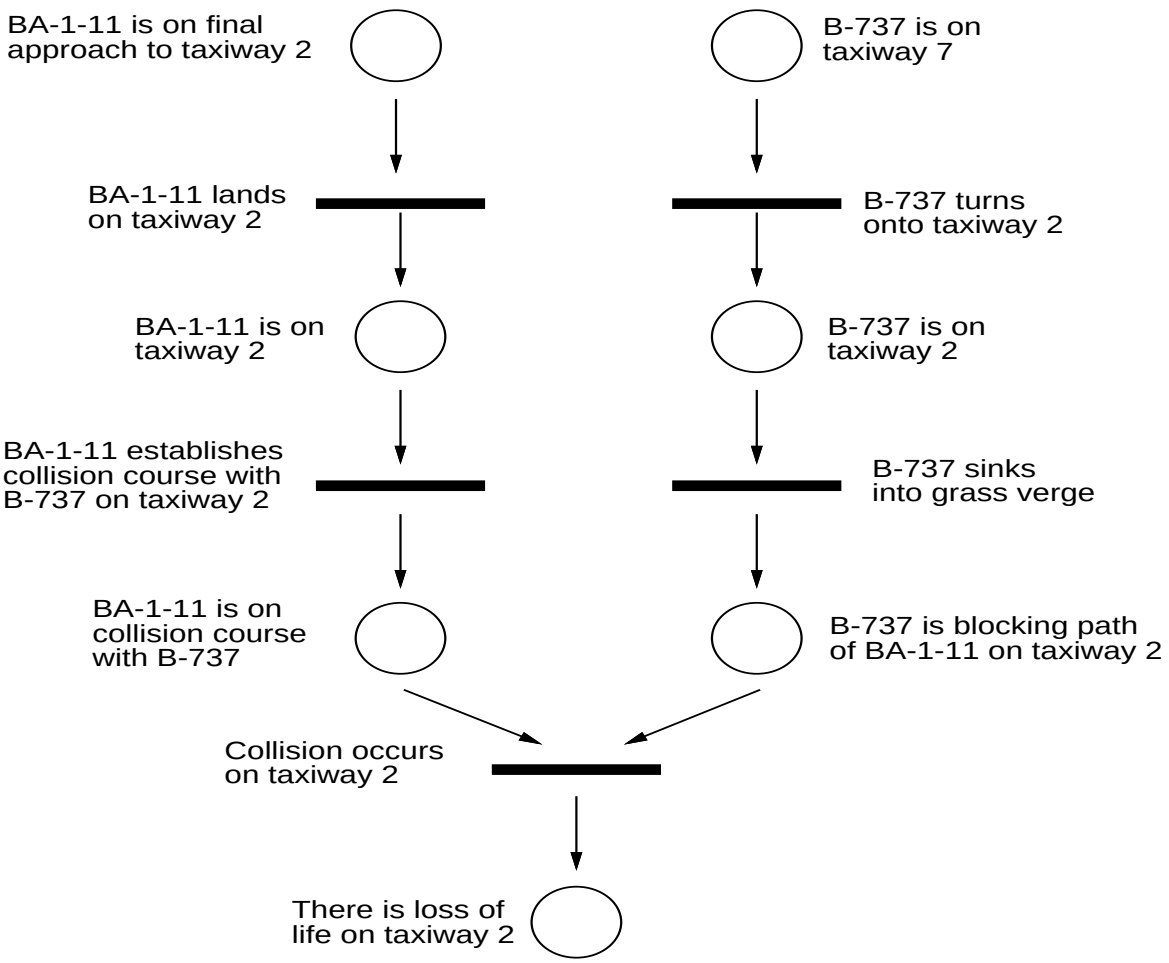

Fig 1. A Simple Petri Net.

course to fire. The transition representing the B-737 turning to block the path of the BA-1-11 can also fire. If these transitions fired then the places indicating that the BA-1-11 is on a collision course and that the B-737 is blocking taxiway 2 would be marked with tokens. The marking of these places would, in turn, enable the collision transition leading to a state in which loss of life occurs.

5 Concurrency

Figure 2 shows how Petri Nets can be used to represent sequential behaviour. The places and transitions that led the B-737 to block taxiway 2 can be traced along a single path of input and output edges. Disasters are, however, often caused by the interaction between concurrent users and systems [4]. It is important that Petri Nets can be used to represent the behaviours of these multiple agents. Figure 3 shows the results of a human factors analysis of the communications between the crew of the B-737 and the air traffic controller. The consequences of the controller's intervention are indicated by

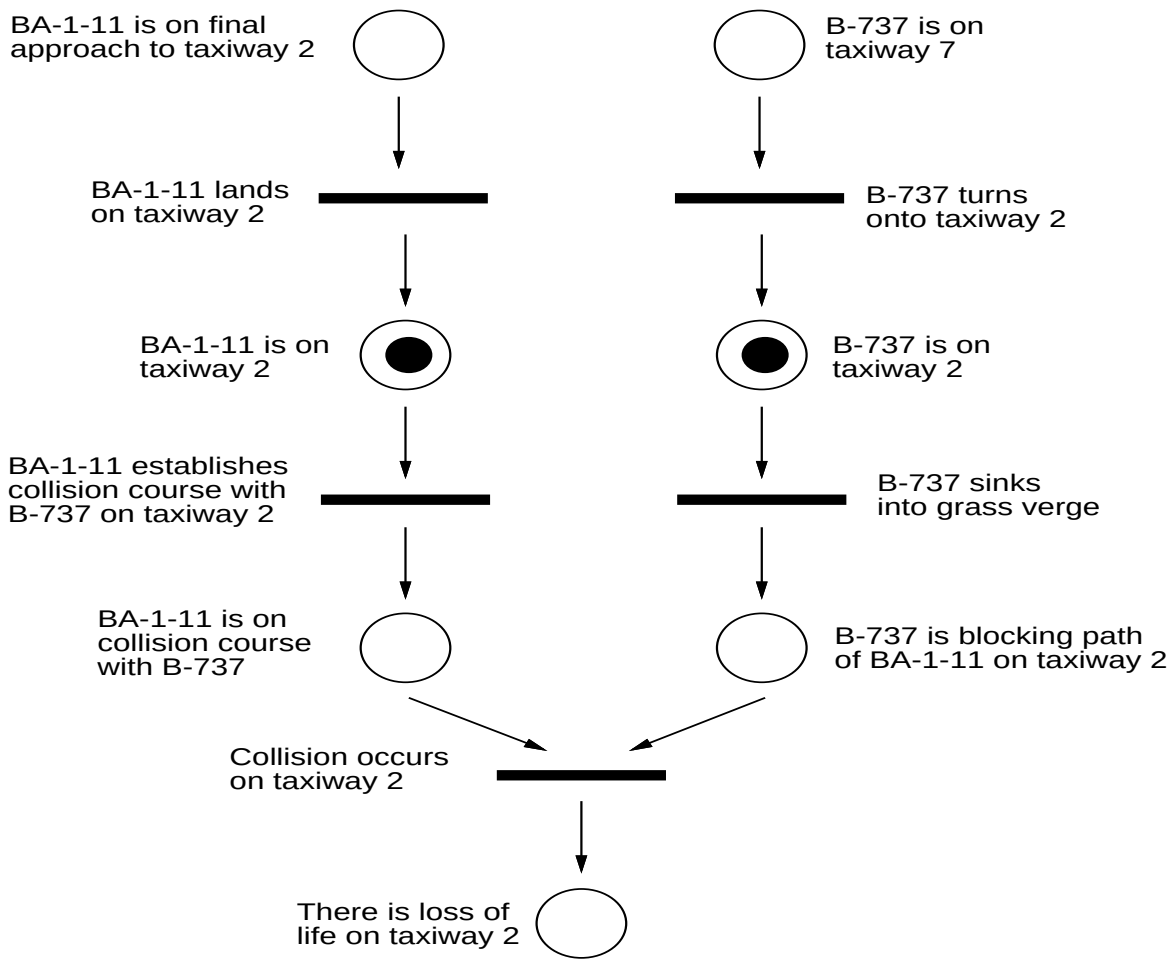

Fig. 2. A Petri Net With Multiple Tokens.

the input and output edges leading from the place labelled `Ground Movement Radar shows BA-1-11's new course for taxiway 2`. Figure 3 also shows how Petri Nets can be used to represent the effects of human intervention upon systems engineering. The controller's commands directly led to the B-737 sinking into the grass verge:

> "Because the landing gear (of the B-737) had left the paved taxiing surface, the appropriate inspections were performed. The only damage detected was a fuel leak on the upper surface of the left wing, where sealant within the fuel tank had become loose at a skin joint." (Air Accidents Investigations Branch, [1]).

The damage sustained by the BA-737 is represented by the transition labelled `B-737 fuel tank sealant is damaged`. In order for this transition to fire both of its input places must be marked. The B-737 must be on taxiway 2 and the air traffic controller must order it off the paved surface. These places, therefore, represent necessary but not sufficient conditions for damage to the B-737. It was the interaction between

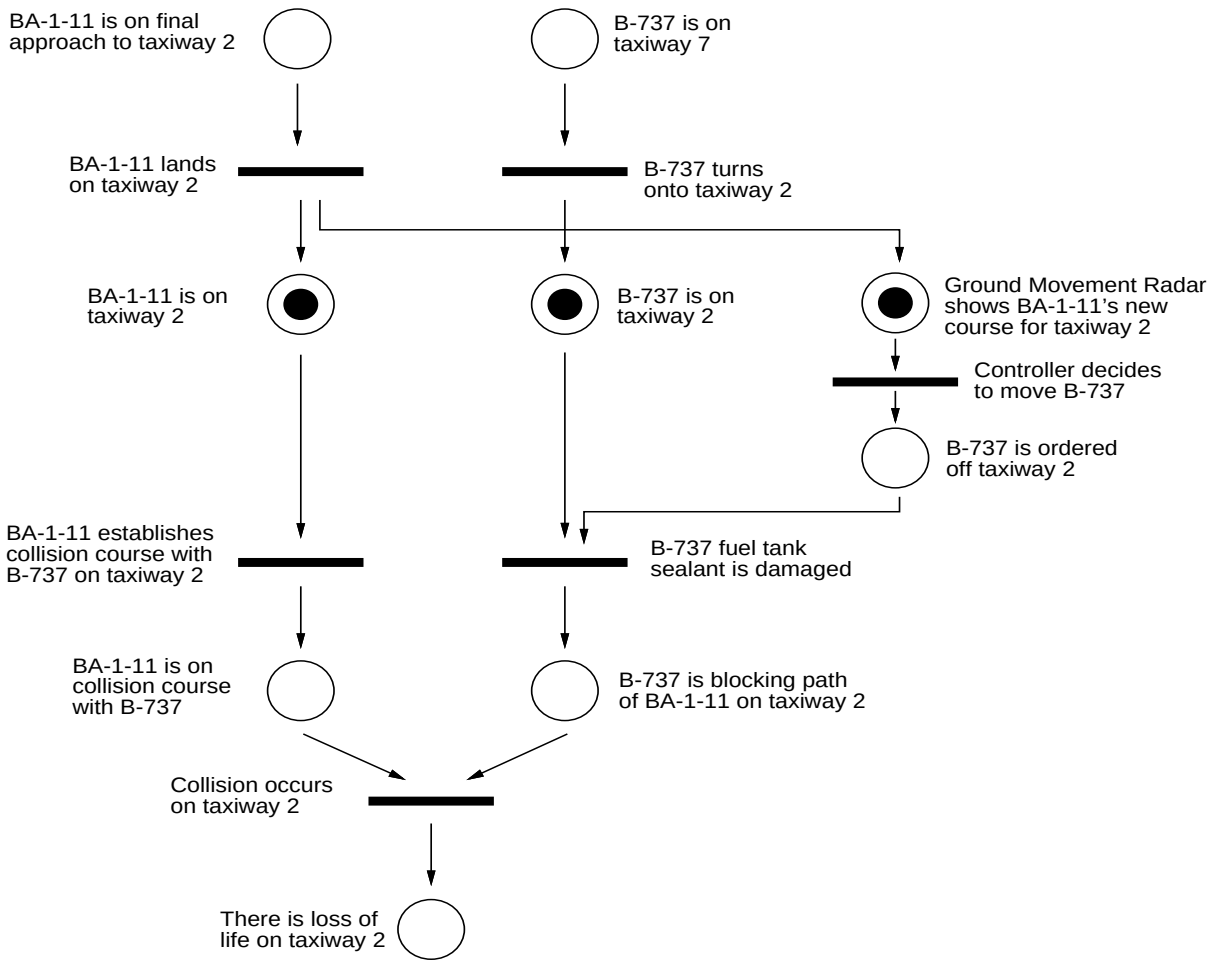

Fig. 3. A Petri Net Showing Concurrent Interaction.

the concurrent behaviour of the air traffic controller and the crew of the B-737 that contributed to the causes of the accident. If the crew had been warned of the BA-1-11 before the turning onto taxiway 2 or if the controller had not ordered the B-737 onto the verge than the B-737 might not have been left blocking the taxiway.

Petri Nets can capture the synchronisation requirements that must be satisfied in order for an accident to occur. For instance, the B-737 must be blocking the taxiway and the BA-1-11 must be on a collision course if loss of life is to occur. This is represented in Figure 3 by the transition that leads to the place labelled `There is loss of life on taxiway 2`. This links the sequences describing the behaviours of the BA-1-11 and the B-737. Such transitions represent interaction between concurrent human-machine systems. They also provide a focus for accident analysis. The ultimate aim of our accident analysis is to ensure that the transition labelled `Collision occurs on taxiway 2` never fires. This can be achieved if its input places are not both marked. In other words, it should never be the case that the B-737 is blocking the path of the BA-1-11 and the BA-1-11 is on a collision course for the B-737. The fact that this is readily apparent from a cursory glance at Figure 3 illustrates an important benefit for Petri Nets as communications tools. It is important to note, however, that Figure 3 simply repre-

sents the events that might have led to a possible loss of life. It does not recommend means of avoiding such an incident. This illustrates an important point about the use of formal notations. They provide precise and concise means of capturing the causes of failure in human factors and systems engineering. They provide communications tools and can be shown to the other participants in an enquiry. They do not provide a panacea for the problems of accident analysis. They do not replace the judgemental skills that must be developed by human factors and systems engineers. In our scenario, there is no automatic means of moving between the Petri Net representation and the remedies that can prevent an accident from recurring.

6 Conflict

The previous sections of this paper have shown how Petri Nets can be used to represent the causes of an accident. Human factors and systems engineers can also use Petri Nets to analyse techniques that might prevent a system from failing. For instance, Figure 4 represents two possible outcomes for our scenario. One terminating place shows loss of life. The other shows that the disaster is averted. Analysts can use such networks to focus attention upon techniques that might ensure that loss of life does not occur. Human factors and systems engineering must be exploited so that the transition, labelled `BA-1-11 establishes collision course with B-737 on taxiway 2`, never fires. The reason we are concerned to disable this transition is that it is one possible outcome from what is known as a conflict situation. The place labelled `BA-1-11 is on taxiway 2` is marked and so it is possible to fire either the transition indicating a collision course or the transition representing insufficient velocity. The network does not indicate which of these two possible transitions will fire. Given this marking we can, however, be sure that only one will fire and that they cannot occur simultaneously. Firing the collision course transition would remove a token from the place labelled `BA-1-11 is on taxiway 2`. This would disable the insufficient velocity transition. Petri Nets that include conflict situations are non-deterministic. If many different transitions are enabled then any one of them can be selected for firing. This provides a means of representing the non-determinism that is inherent in many complex multi-user, multi-system applications.

Conflict situations represent critical stages in an accident scenarios. Non-determinism indicates a loss of control over the behaviour of the human-machine system. Human factors and systems engineers can, therefore, seek to improve the design of a system by identifying and removing conflict from Petri Net representations. In our example, this might be done by ensuring that the air traffic controller issued an earlier warning to the crew of the BA-1-11 as it approached taxiway 2. They, in turn, might have responded by ensuring the prompt and sufficient application of reverse thrust to rapidly reduce their landing speed:

> "Figures calculated by British Aerospace for the ground roll from 121 kt at touchdown to stop, based on a dry runway with full brakes applied for 1.2 seconds after touchdown, did not exceed 490 meters, although the figure achieved depended upon the use of spoilers and the level of thrust applied. The calculated figures were considerably shorter than the 960 meters achieved during the incident by WB (the BA-1-11) and indicate that maximum braking and/or maximum reverse thrust were not used for the whole of the landing run" (Air Accidents Investigations Branch, [1]).

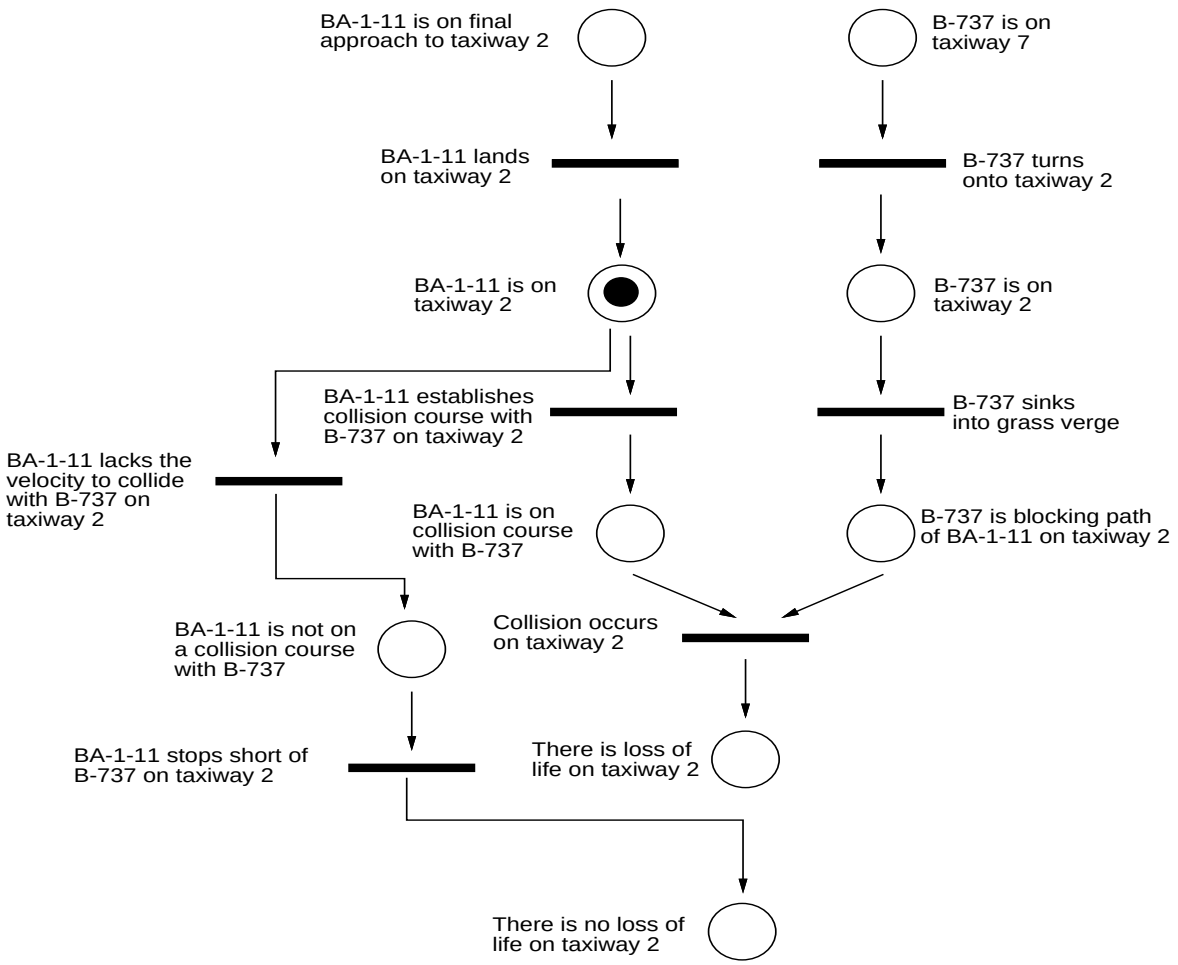

Fig. 4. A Petri Net Showing Conflict.

Human factors and systems engineering must both be recruited in order to achieve the common aim of ensuring that air traffic controllers warn aircraft approaching taxiways. Systems engineering could have extended the range of the Ground Movement Radar. Human factors engineering might have verified that the controller could sample the Ground Movement Radar frequently enough to observe the BA-1-11's change in course. Inhibitor arks provide a means of representing the common aim for both of these disciplines. Transitions that are linked by an inhibitor can only fire if the place from which the inhibitor comes is not marked. Inhibitors are represented graphically as an edge with a small empty circle on one end. In Figure 5 an inhibitor arc is shown running from the place labelled `Air traffic controller is informing BA-1-11 of B-737` to the transition marked `BA-1-11 establishes collision course with B-737 on taxiway 2`. In our example, the place is marked. It has no other output transitions and so the collision course transition cannot be enabled.

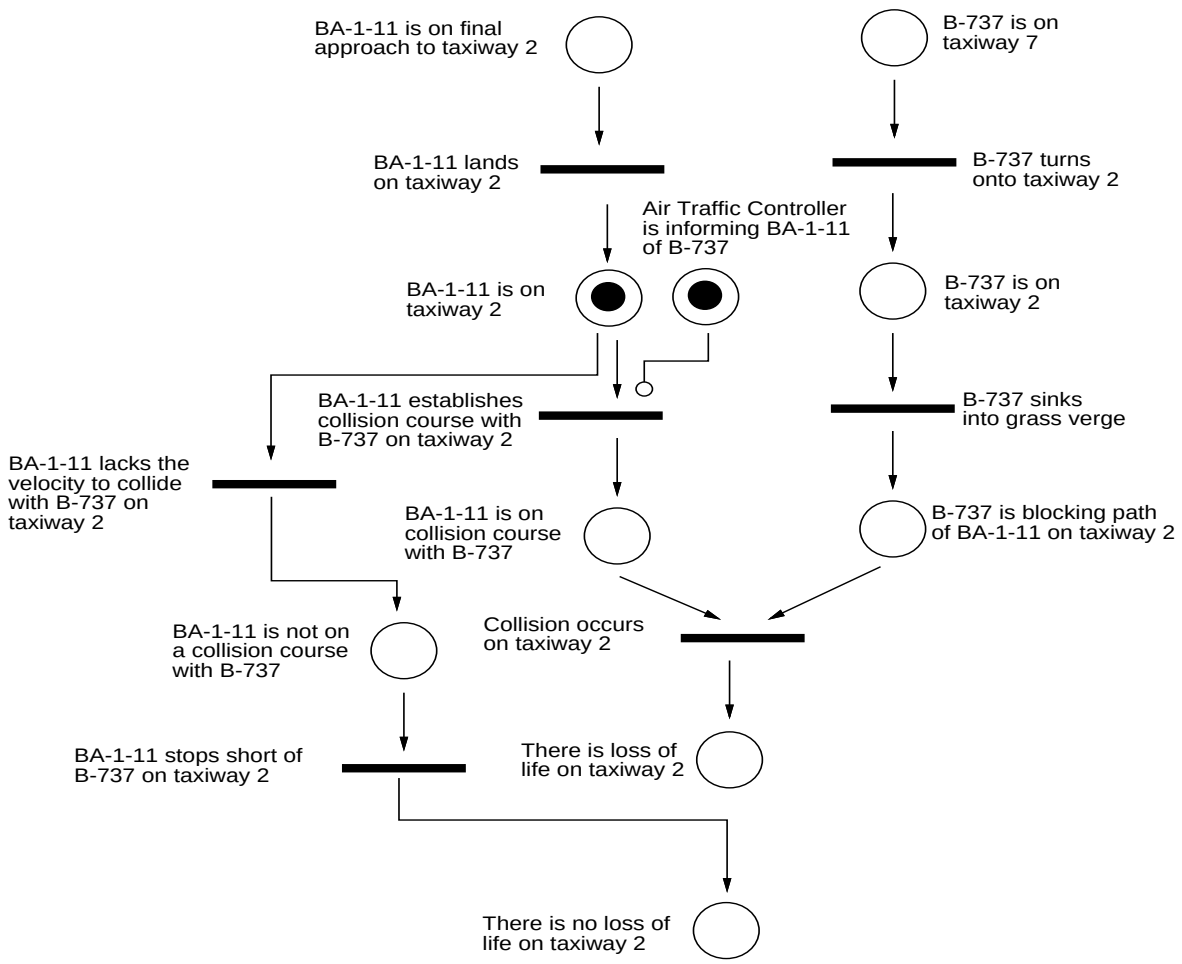

Fig. 5. A Petri Net With An Inhibitor Avoiding Conflict.

7 Hierarchical Petri Nets

The networks in previous sections of this paper represent our scenario at an extremely high level of abstraction. This is inappropriate for many of the findings that emerge from human factors and systems analyses. The exact ways in which systems gather and store data can have a profound impact upon the course of an accident. For example, the Automatic Terminal Information Service at Gatwick did not change the message identifier to indicate that the active runway had changed from 08R to 08L. Detailed cognitive and perceptual factors can influence an operator's response to potential disasters. For instance, the AAIB argued that the interaction between the crew of the BA-1-11 was a prime cause of the near-collision with the B-737. The more dominant and decisive commander was in command. The more contemplative and less assertive individual, who could have questioned the approach to the taxiway, was undergoing training. Petri Nets can also be used to model these details. Places and transitions can be replaced by sub-networks to provide finer grained representations. The place labelled `BA-1-11 is on final approach to taxiway 2` can be refined into the sub-network shown in Figure 6. This captures some of the environmental factors that influenced the actions

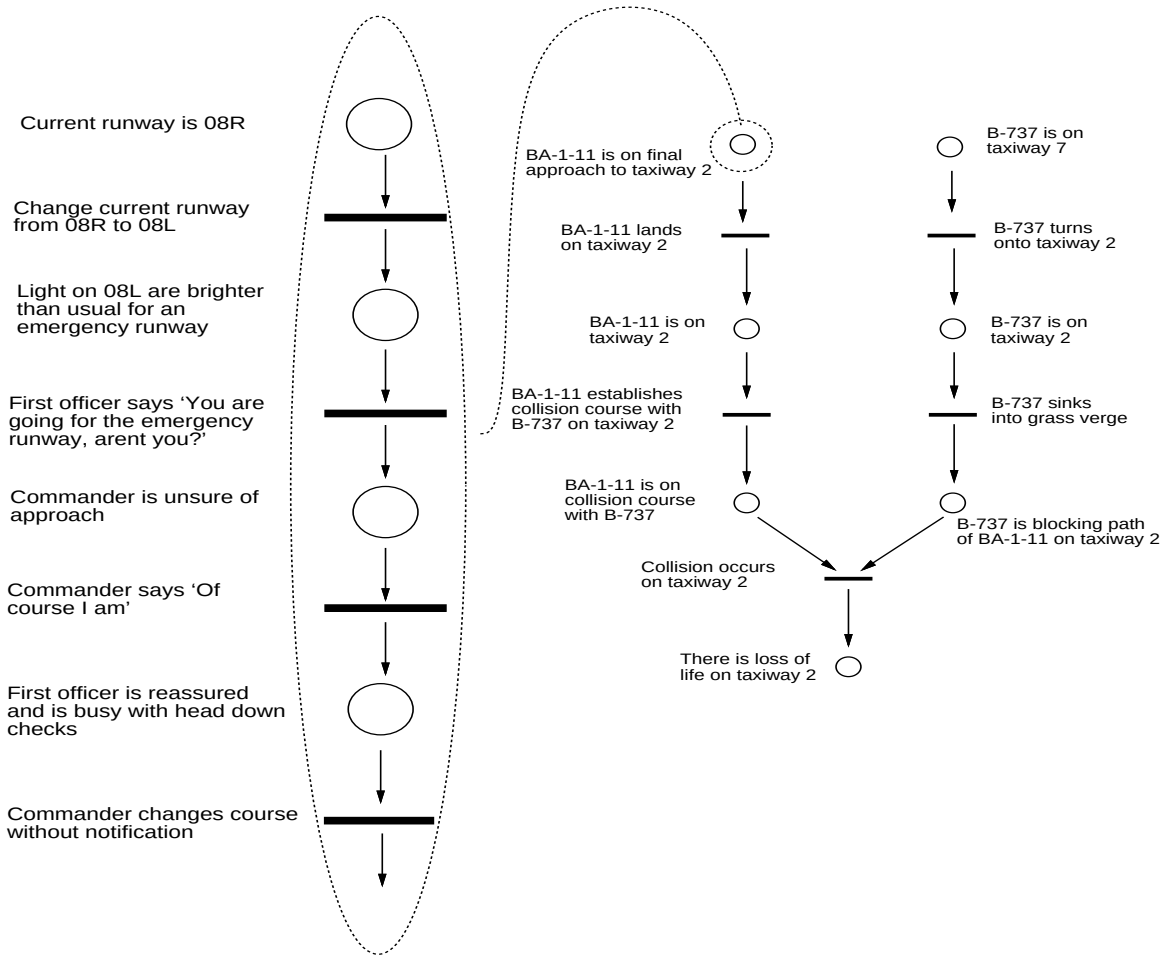

Fig 6. A Sub-Net Showing Crew Interaction.

of the BA-1-11's crew. The fact that the lights on 08L were much brighter than the first officer expected for an emergency runway led him to ask "You are going for the emergency runway, aren't you?". This is represented in the Petri Net by the place, labelled `Lights are much brighter than expected`, followed by the transition labelled with the first officer's utterance. Here the place must be marked in order to enable the transition. In other words, the lights must be brighter than expected for the first officer to make their remark. The Petri Net, therefore, represents the causal link that the AAIB identified between the perceptual cues provided by the taxiway lighting system and the first officer's question:

> "The first officer made this remark because he could see the lights of Gatwick and was a little concerned that the dominant set of lights visible to him (those of 08L) seemed to be of a much higher standard than he was expecting for the 'emergency runway'... These doubts had not crystallised into a clear cut mental model for the first officer and it was certainly in an attempt to provoke some sort of discussion of the situation, and from the discussion to gain a clear model with which he was happy, that he made

his remark, 'You are going for the emergency runway, aren't you?' " (Air Accidents Investigations Branch, [1]).

This quotation illustrates the way in which cognitive factors, "mental models", can influence the course of events leading to accidents. Figure 6 also shows how Petri Nets can represent some of the assumptions that analysts make about the cognitive states of individual participants. In particular, the transition labelled with the first officer's remark enables a place in which the commander is unsure about their approach. Again this illustrates a causal link in the sequence of events investigated by the accident analysts. The commander's change of interpretation was caused by a query from the first officer in an attempt to resolve his uncertainty about which runway was being approached. A similar causal relationship can be seen between the commander's utterance which confirmed that he was sure of his approach and the place which indicates that the first officer is re-assured.

Accident analyses should, ideally, identify techniques that can preserve the safety of human-machine systems. For instance, the first officer might have been made aware of the commander's actions by requiring both crew members to confirm any change of direction during their approach. Operating procedures could stipulate that crew members reach a verbal agreement before issuing certain commands during an approach. The human factors section of the AAIB report stresses the need to improve crew coordination through flight deck management training. Unfortunately, such protocols might not have guaranteed the safety of the aircraft:

> "The nature of the flight deck would have made such an intervention even less likely; an individual being assessed will wish to be very sure of himself before appearing to question the competence of his assessor... The properties of these crew members which made it possible for them to land on taxiway 2 were their ... relative personalities, roles and status: the more dominant and decisive individual was in command and control, with the more contemplative and less assertive individual in a position of both subordination and evaluation." (Air Accidents Investigations Branch, [1]).

The fact that the first officer did not call for an overshoot when he eventually noticed the commander's 'error' confirms the suspicion that that this human factors technique might not have been effective. System support could have avoided such problems. For instance, both the commander and the first officer could have been required to provide input before the approach was changed. Avionics might have required both crew members to 'vote' for certain commands. Figure 7 shows how Petri Nets can be used to represent such systems engineering techniques at finer levels of detail. If the transition representing the first officer's input does not fire then no token will be deposited in the place representing the fact that their command is processed. This, in turn, would prevent the change in direction transition from being fired. Unfortunately, the problems of systems engineering limit the utility of this approach. Not the least of these are the cost and technical complexity of modifying existing avionics systems. There are also human factors limitations. It is inappropriate to enforce voting mechanisms in cockpits where rapid intervention may be required in order to correct the approach of an aircraft, for instance during strong cross-winds. These problems reinforce the point that formal notations need not represent systems that are achievable in the 'real world'. There is no guarantee that they will provide a faithful model of an application domain. This is entirely dependent upon the skill of the analyst and their ability to elicit the salient

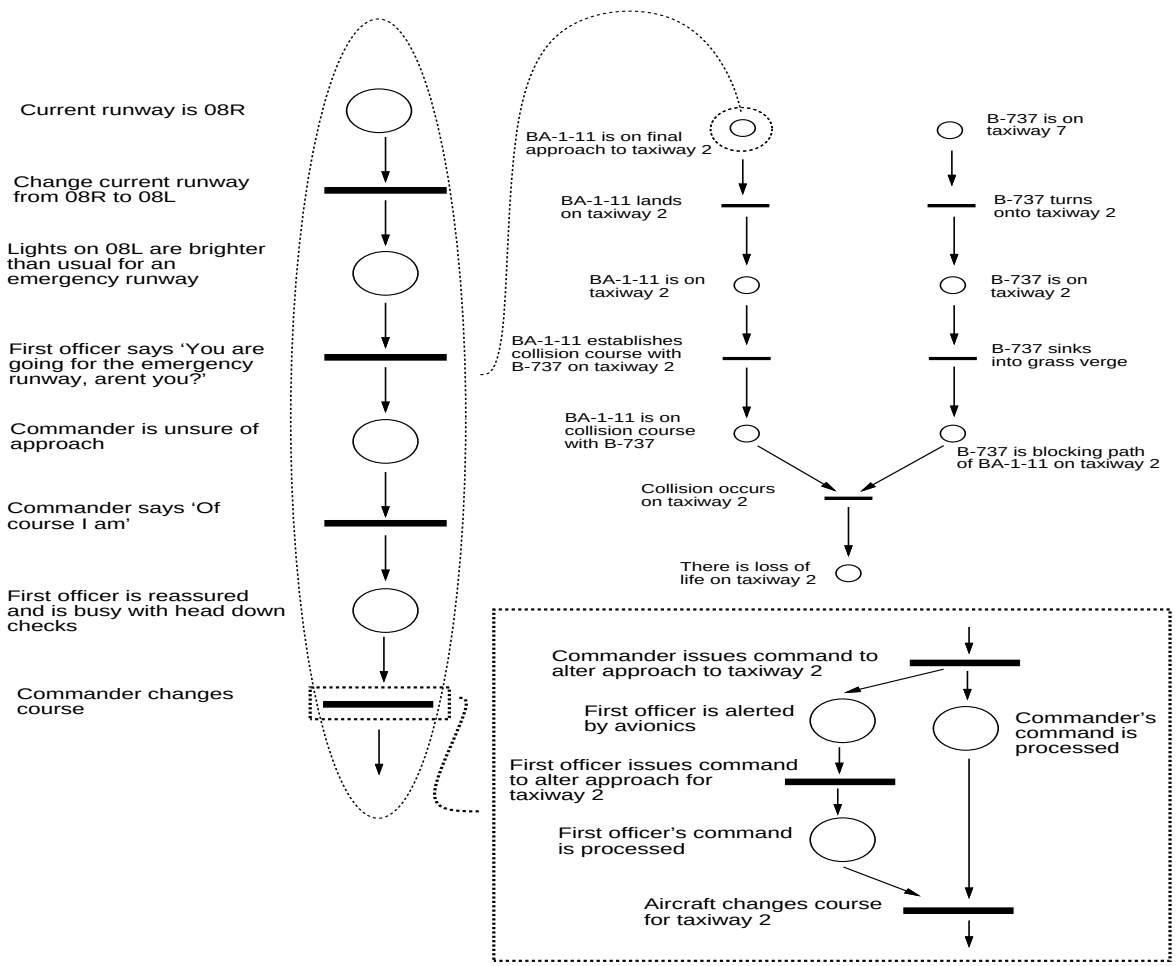

Fig. 7. A Sub-Net Showing A System Command Protocol.

features of an accident sequence. The power of the Peti Net notation is that it provides a means of explicitly representing these 'salient features'.

8 Simulation

It is important that human factors and systems engineers have a clear understanding of the behaviour of the systems represented by Petri Nets. This task can be impaired by the additional complexity that is introduced through the use of sub-networks. It can be difficult to trace the likely passage of tokens through the many places and transitions that might used to represent cognitive, perceptual and environmental details. Fortunately, this task can be eased by tools that animate the enabling and firing of transitions as tokens pass from place to place in a Petri Net. For instance, Chiola's GreatSPN can be used to view tokens as they pass through a network [6]. Human factors and systems engineers can record which places are marked and which transitions are enabled. The "possibility of playing (such) token games highly simplifies the development of correct models, thus eliminating the need for more complex 'net debugging'." A further ad-

vantage is that the resulting animations provide powerful communications tools that can be shown to the many different teams that must collaborate during an accident analysis. All of the Petri Nets presented in this paper were generated and checked using the GreatSPN editor. This tool can be used to create arbitrary sub-networks to form hierarchical Petri Nets. Such facilities are vital if complex networks are to be constructed.

9 Conclusions And Further Work

In this paper we have argued that Petri Nets can be used to represent the findings of accident analyses. Petri Nets are well suited to this task because they are capable of representing concurrent interaction between multiple operators and multiple systems. Simple analytical techniques can be applied to these networks in order to identify instances of potential non-determinism. Such situations provide a focus for the allocation of design resources in order to guarantee safety. The use of hierarchical networks provides a means of representing fine grained human and system behaviours. Simulation tools can be used to validate Petri Nets representations of accident scenarios.

This paper is the result of an initial investigation. Much work remains to be done in order to fully explore the potential of Petri Nets as both a design and analysis tool for human-machine systems. In particular, the networks presented in this paper have not described the device or presentation characteristics that can have a profound impact upon the cognitive and perceptual performance of system operators. Previous work, by the author, has shown how these details can gradually be introduced into logic specifications using structured graphics and device abstractions [16]. Future work intends to extend the application of this approach to capture device and presentation information within Petri Nets. It is anticipated that the introduction of these details might provide a means of deriving prototype systems from sub-networks that describe interactive dialogues.

Future work intends to resolve a number of technical problems that limit the utility of our modeling techniques. For instance, we have used transitions to represent the events that contribute towards major accidents. This decision is justified by the observation that both transitions and events affect the state of interaction [18]. Unfortunately, this approach can lead to problems for more complex descriptions. When an event triggers several actions, it must be represented by a number of duplicate transitions. Future research intends to develop an approach suggested by Palanque and Bastide (personal correspondence). They use dedicated places to represent critical events during the course of interaction. This avoids the duplication created by the use of transitions. Similarly, we intend to explore the use of Petri Net composition techniques to separate out both user and system models during accident investigations. This would considerably simplify some of the larger networks shown in this paper.

The networks presented in this paper have not considered the real-time delays that can occur before a transition fires. This is a significant omission because temporal properties frequently have a profound impact upon the safety of human-machine systems. Delays in system responses can lead to frustration and error [21]. It can be difficult for operators to predict the consequences of their interaction if periods of quiescence allow the system to process a backlog of user input [10]. It is, however, possible to explicitly represent temporal properties within a Petri Net. Zuberek [30] associates firing times with transitions in order to model the low-level behaviour of computer hardware.

Merlin and Faber [23] have exploited a similar approach by associating minimum and maximum firing times with each transition. Further work intends to show how such techniques might be recruited to enhance the modelling power of Petri Nets for both human factors and systems engineering.

The Petri Nets shown in this paper do not represent probabilistic information. This has not been a severe limitation for our scenario but it might well prove to be significant omission in accidents that are caused by the periodic failure of system components. Researchers have enhanced the Petri Net notation to include probabilistic information. Marsan, Balbo and Conte argue that:

> "With the introduction of Stochastic Petri Nets a link is established between two important classes of computer system models: the above mentioned graph models (e.g., Petri Nets) and the probabilistic models familiar to performance analysts" (Marsan, Balbo and Conte, [22]).

Future work intends to investigate whether a further link might be established between these approaches and human factors techniques. This is important because the probabilistic behaviour of an interactive system can frequently influence the control strategy of its operators [14]. If a component is know to fail frequently then users will, typically, anticipate the failure by devoting greater resources to monitoring its state [26].

10 Acknowledgements

This paper is the result of collaborative research with Dr. Peter Wright of British Aerospace's Dependable Computing Systems Center, University of York and Dr. John McCarthy, Department of Psychology, University of Cork. The members of the Human Computer Interaction Group in the Department of Computer Science, University of York provided valuable advice and encouragement for this research. Thanks are also due to Itana Gimenes who helped to provide access to GreatSPN. This research was supported by a joint British Council/EOLAS grant, number 9284(1992).

11 References

1. Air Accidents Investigations Branch. *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. Her Majesty's Stationery Office, (1989).

2. Air Accidents Investigations Branch. *Report On The Accident To Boeing 737-400 G-OBME Near Kegworth, Leicestershire on 8th January 1989*, number 4/90, London, United Kingdom. Her Majesty's Stationery Office, (1990).

3. Bastide, R. and Palanque, P. Petri net objects for the design, validation and Prototyping of user-driven interfaces. In D. Diaper, D. Gilmore, G. Cockton, and B. Shackel, editors, *Human-Computer Interaction—INTERACT'90*, pages 625–631. Elsevier Science Publications, North Holland, Netherlands, (1990).

4. Bignell, V. and Fortune, J. *Understanding System Failure*. Manchester University Press, Manchester, United Kingdom, (1991).

5. Van Biljon, W.R. Extending Petri nets for specifying man-machine dialogues. *International Journal of Man-Machine Studies*, 28(4):437–455, (1988).

6. Chiola, G. GreatSPN users' manual. Technical report, Departmento di Informatica, Universita' delgi Studi di Turino, Turino, Italy, (1987).

7. Chretienne, P. Timed Petri nets: A solution to the minimum-time-reachability problem between two states of a timed-event graph. *Journal of Systems and Software*, 6(1-2):95–101, (1986).

8. Cox, A.P. editor. *Risk Analysis In The Process Industries: The Report Of The International Study Group On Risk Analysis*. EFCE No. 45. The Institute Of Chemical Engineers, Rugby, United Kingdom, (1985).

9. Davies, K.R. 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).

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

11. Hura, G.S. and Attwood, J.W. The use of Petri nets to analyse coherent fault trees. *IEEE Transactions On Reliability*, 37(5):469–473, (1988).

12. International Atomic Energy Agency and The Commission of the European Community. *Critical Survey of Research On Human Factors And The Man-Machine Interaction*, IAEA-SM-26B/29, Vienna, Austria, (1984).

13. Johnson, C.W. 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).

14. Johnson, C.W. Specifying and prototyping dynamic human-computer interfaces for stochastic applications. To appear in *People And Computers VIII*. Cambridge University Press, Cambridge, United Kingdom, (1993).

15. Johnson, C.W. Using Z To Support The Design Of Interactive, Safety-Critical Systems, BCS/IEE Software Engineering Journal, (10)2:49-60, (1995).

16. Johnson, C.W. and Harrison, M.D. Declarative graphics and dynamic interaction. In F.H. Post and W. Barth, editors, *EUROGRAPHICS '91*, pages 195–207. Elsevier Science Publications, North Holland, Netherlands (1991).

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

18. 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).

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

20. Kramer, B. Introducing the GRASPIN specification language SEGRAS. *Journal of Systems and Software*, 15(1):17–31, (1991).

21. Kuhmann, W., Boucsein, W., Schaefer, F. and Alexander, J. Experimental investigation of psychophysiological stress-reactions induced by different system response times in human-computer interactions. *Ergonomics*, 30(6):933 – 943, (1987).

22. Marsan, M.A., Conte, G. and Balbo, G. A class of generalised stochastic Petri nets for the performance evaluation of multiprocessor systems. *ACM Transactions On Computer Systems*, 2(2):93–122, (1984).

23. Merlin, J.A. and Faber, D.J. Recoverability of communications protocols - implications of a theoretical study. *IEEE Transactions On Communications*, COM-24(9), (1976).

24. Peterson, J.L. Petri nets. *Computing Surveys*, 9(3):223 – 252, (1977).

25. President's Task Force On Aircraft Crew Compliment. *United States' Government Report On Aircraft Crew Compliment*, Washington DC, United States of America (1981).

26. Reason, J. *Human Error.* Cambridge University Press, Cambridge, United Kingdom, (1990).

27. Wagenaar, W.A. and Groeneweg, J. Accidents at sea : Multiple causes and impossible consequences. In E. Hollnagel, G. Mancini, and D.D. Woods, editors, *Cognitive Engineering In Complex Dynamic Worlds*, pages 133 – 144. Academic Press, London, United Kingdom (1988).

28. Watson, I.A. Review of human factors in reliability and risk assessment. In J. Burgoyne, editor, *The Assessment And Control Of Major Hazards*, pages 323–337. Pergamon Press, Oxford, United Kingdom (1985).

29. Worley, N. and Lewins, J., 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).

30. Zuberek, W.M. Timed Petri nets and preliminary performance evaluation. In *Proceedings Of The 7th Annual Symposium On Computer Architecture*, pages 88–96. ACM and IEEE, New York, United States of America (1980).