THEA:
Human Error Analysis for Requirements Definition

Bob Fields, Michael Harrison, Peter Wright

Human-Computer Interaction Group
Department of Computer Science
University of York
York, YO1 5DD
bob, mdh, pcw @cs.york.ac.uk

Abstract

THEA is a technique developed to help designers in interactive systems (originally in the aviation domain, but hopefully applicable in other contexts) to anticipate interaction failures or "human errors" that may be problematic once their designs become operational. The techniques is intended for use early in the development lifecycle, as design concepts and requirements concerned with safety and usability, as well functionality are emerging.

This report uses examples from two flight deck based case studies to illustrate how to use the THEA technique for carrying out a human error analysis during early design. The aim is that this document should accompany a one day tutorial and should be sufficient to capture the essence of the design method. This document introduces material that is an evolution of the THEA techniques and includes some new material and is intended to preserve the flavour of the earlier document that gave a "how to do it" guide to techniques developed both in the Dependable Computing Systems Centre at York, and elsewhere, aimed at practitioners.

Contents

EXECUTIVE SUMMARY 3

PART I. THE HUMAN ERROR ASSESSMENT PROCESS 4

1. Introduction 4

2. Scenario Use and Description 6
2.1 What's in a scenario? 6
2.2 Where do scenarios come from? 7
2.3 When have you collected enough scenarios? 8
2.4 Example Scenario 1 8
2.5 Example Scenario 2 9

3. Understanding the Task Context 11
3.1 Hierarchical Goal Decomposition 11
3.2 Plans 11
3.3 Task Descriptions in Scenario 1 12

4. Understanding the System Context 12
4.1 Confusion and complexity 13
4.2 Authority limiting 13
4.3 Modes and mode transitions 14

5. Understanding Actions in Context 14

6. Understanding Operator Error 15
6.1 Cognitive failure 16
6.2 Deviations from expected behaviour 17

PART II. ANALYSIS 18

7. The Error identification process 18
7.1 Applying the cognitive error analysis 19

8. An analysis example 21

9. References 21

PART IV. APPENDIX 23

10. Detailed scenario descriptions 23
10.1 Scenario 1 23
10.2 Scenario 2 25

11. Error analysis example 28

Executive Summary

The aim is...

To describe a technique for the iterative analysis and design of dependable interactive systems. The means by which this is done is to analyse how the behaviour of human operators contributes to overall system dependability, and to use this understanding relatively early in the design process when requirements and concepts for the user interface design of a product are emerging.

The aim isn't...

To support the process of making quantitative estimates of the likelihood of human errors occurring. Rather, the aim of the techniques described here is to help designers to reason about errors early in the design lifecycle for interactive systems, and to take account of such reasoning when the design is still fairly fluid and flexible.

Users and intended audience

The intended users of this document and of the technique it describes are primarily systems engineers who are involved from the early stages in the design lifecycle of products with substantial interactive components. No particular background in human factors, cognitive engineering, or psychology will be assumed, though engineers using the approach may, from time to time, need the assistance of human factors specialists to resolve specific issues. While it is intended that human factors expertise is not essential for the process, an understanding of the domain and the context in which a new system is to be used is much more important. Indeed, the technique can be seen as a way of allowing engineers to bring their application domain expertise to bear on user interface design problems and the dependability implications of interface design decisions.

Structure of the document

This document describes techniques that can be used by a designer to analyse human error and its effects on a system under design. The primary input to the analysis technique is a collection of *scenarios* that help the designer to envisage how a system currently being developed will be used in future. The primary output will be a description of a number of problem areas in design and its operation that may be the cause errors.

Part I of this document describes several important constituents of the description. These include physical and environmental setting of an episode of system use, the tasks that humans in the scenario will carry out, and task knowledge they will possess, the functionality and user interface characteristics of various technical systems that are relevant, and so on. In addition, in Part I a model of human error based on both behavioural and cognitive views of error (the latter structured by a high-level model of human information processing) is described. The models of scenarios and human error developed in Part I are used as the input to an error analysis process, described in Part II. This process is based around a questionnaire that can help designers to anticipate where some of the error problems in the operation and use of a new system might lie. The questionnaire is structured around the model of error, and the information required to answer questions will be found in the scenario descriptions.

Part I. The human error assessment process

1. Introduction

The THEA (Techniques for Human Error Assessment) approach has its roots in the class of methods of *Human Reliability Analysis*, for the most part developed in the nuclear power industry. Their aim is to assist in analysing the dependability and reliability of systems with a human component. Human error is a significant factor in the success of take-up of any system and it is particularly of concern where activities are safety critical. Our aim has been to produce a technique that is not expensive to apply and has a role in the process of developing a design.

> For more information about Human Reliability Analysis techniques in general, see
>
> - B. Kirwan: *A Guide to Practical Human Reliability Analysis*. (Kirwan 1994)
> - B. Kirwan: *Human Error Identification in Human Reliability Assessment. Part 1: Overview of approaches*. (Kirwan 1992)

The main components of the THEA assessment process are:

Understanding the work a system will be used for

- Scenario elicitation and representation: taking representative examples of the use of the system that can be used as a basis for establishing requirements for the new design, particularly those requirements that relate to human error vulnerabilities.
- Task description: a representation of the work that the operator(s) are intended to do in terms of goals, plans and actions.

Understanding the device being designed

- System description: a specification of relevant aspects of the new system's functionality and interface, and how it interacts with other systems in the application domain.

Understanding how errors can arise

- Model of human cognition (the execution-evaluation model of human information processing). This can be used to help understand some of the cognitive causal factors that can lead to error.
- Error analysis: the identification and explanation of human error that may arise in the operation of the system (possibly as a result of the way it is designed).

Designing for error

- Impact analysis and design iteration: assessment of the likelihood of the human error and the implications for design.

The Analysis Process

Figure 1 gives an overview of the main components of the analysis process. The primary source of information used in the process is a collection of *scenarios*, or descriptions of concrete episodes and situations in which a system or device being designed is used. A scenario description therefore contains not only the functional behaviour of the device, but also the initial conditions for the scenario, the tasks for which it is designed, the humans who participate in the scenario, and so on.

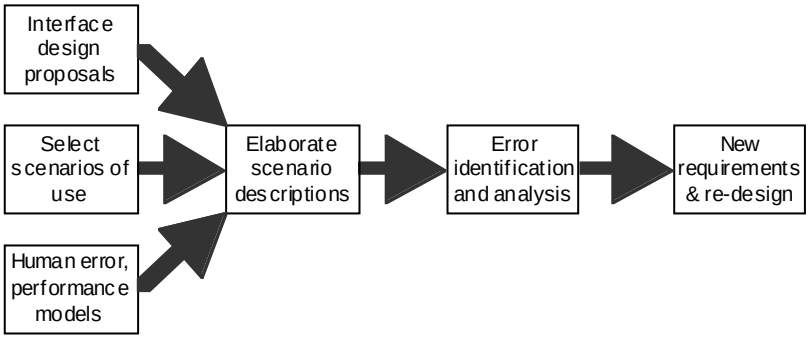

Figure 1: Structure of the method

The process implied by Figure 1 is intended to be applied iteratively in the sense that decisions made and conclusions drawn later in the diagram may influence those appearing earlier. The aim is that this process of human error assessment should begin early in the design process. The purpose of the analysis is to establish requirements on the design that will enable a more human error resilient system design. The method requires as raw data: a model of a proposed user interface, an understanding of the work that has to be done by the system exemplified by a number of carefully chosen scenarios, and a model or means of thinking about human error. The analysis process involves two steps: a process of potential error identification and an analysis of the consequence and significance of the error. The aim is that this process should lead to a revision of the design as contained in the specification or the work description.

The information that can serve as input to this process, and helps the analyst select and construct scenarios, comes from a number of sources. Three types of information that can help to suggest scenarios that might be of value are: proposals for how the new systems will work; how similar systems were used in the past; and models of human error and human behaviour.

Information about the system design

When an error analysis is carried out there will typically be some concrete proposals for how the new device or system will work, and what functions and features its interface will provide (and indeed, it is these proposals which are, to some extent, being analysed by the method). Knowledge about the system and interface design, will be an important input to the error analysis process. It is often the case that new designs are not created from scratch, but are modifications or re-designs of some existing product. In such situations, understanding the differences between the old and new versions will be highly informative.

Historical information and operational experience

When a new system is a re-design of an existing system, there will often be historical information in existence about how the old system performed, how it was used in practice, what the good and bad features of the old technology were, and so on. Even if the new system has been designed from scratch, there will frequently be plenty of historical data on the past use of similar systems, or systems performing a similar function.

Some of the important sources for such data are:

- Prescriptions of how the system should be used, in the form of instructions, manuals, standard operating procedures, training material, task analyses, and so on.
- Descriptions of particular problems and incidents that took place. In safety critical areas such as aviation, these are often formally collected and published, for example as aircraft accident investigations.
- Accounts provided by real practitioners, designers, and other stakeholders of how they carry out their work using existing systems. This includes where the problem areas and weak points are, what situations and circumstances are particularly challenging, and how changes in technology might cause new problems or alleviate old ones.

Information about behaviour and human performance

A number of models, theories and collections of empirical data about human performance and human error exist and can be useful in deciding which scenarios will be important to look at, and how

participants will act in a given scenario. In this document we make use of a particular model of human behaviour in order to structure our analysis of errors, but other models can be useful and informative (see, for example, (Hollnagel 1993; Reason 1990)).

2. Scenario Use and Description

One of the most important antecedents of the error analysis process is to develop an understanding of how the technological system or sub-system being designed will be used in practice. In order to do this we suggest the identification and collection of "usage scenarios" that represent the use of a system *in context* (Carroll 1995; Greenbaum and Kyng 1991). Very simple scenarios are often used in the aerospace industry as a means of assessing the consequences and possibilities of a design, in the form of "forcing missions". The choice of missions is often based on criteria concerned with mission effectiveness of a system, and involves making judgements about the difficulty of the achievement of mission goals. In the THEA approach we are more concerned with choosing usage scenarios that highlight how a design creates opportunities for human error, thereby having an impact on dependability.

The purpose of THEA is to use systematic methods of asking questions and exploring interactive system designs based on asking focused questions about how a device functions *in a scenario*. The purpose of doing this is to provide a systematic and structured way of critiquing a design, and developing further requirements.

The basic claim of the scenario-based approach to development is that the design process should take the specific and concrete, rather than the general and abstract as its primary input. The justification for this view is that concrete examples allow practitioners to better envisage and articulate how they would behave in a given situation, in turn allowing designers to envisage how their designs may be used.

2.1 What's in a scenario?

The purpose of using scenarios in design is to give designers and analysts a way of capturing how a proposed design will be used. This means that a description of a scenario must cover not only the *actions* that take place in a given situation, but also the *contextual factors* that surround the action, allow it to happen, and provide opportunities for "errors".

The aspects of context that should be recorded in scenario description encompass the physical environment and situation in which participants find themselves, the task context and the system context. In addition to these "contextual factors" we will also describe the actions that take place, and how they relate to the context, as well as any likely alternative courses of action.

A "template" form for describing scenarios, with spaces for recording this contextual information, is shown in Figure 2.

Agents

- The human agents involved and their organisation
- The roles played by the humans, and the goals and responsibilities they have

Rationale

- Why is this scenario an interesting or useful one to have picked?

Situation and Environment

- The physical situation in which the scenario takes place
- External and environmental triggers, problems and events that occur in this scenario

Task Context

- What tasks are carried out?
- What formal procedures are there, and are they followed as prescribed?

System Context

- What devices and technology are involved? What usability problems might they have?
- What effects can users have?

Action

- How are the tasks carried out *in context*?
- How do the activities overlap?
- Which goals do actions correspond to?

Exceptional circumstances

- How might the scenario evolve differently, either as a result of uncertainty in the environment or because of variations in agents, situation, design options, system and task context?

Figure 2: Template for describing scenarios

2.2 Where do scenarios come from?

In order to identify situations that may be significant we make use of the information sources:

- The stories and **experiences** of practitioners (pilots, operators, other crew members — the "users") and of other domain experts (the designers, human factors experts, maintenance or training personnel, etc.). Some developers recruit experts who have extensive experience of **earlier versions of the system**.
- Historical reports about problem areas, incidents, likely events. For example, (Fischer, Orasanu et al. 1995) use real situations (taken from the ASRS aviation incident database[1]) as a way of eliciting pilot's assessments of the factors governing various kinds of decision making. **Incident and accident reports** are a useful source of historical information.
- Frequent conditions and normal operation. This could be based on expert judgment or logs of use of an existing system.
- **Changes in technology**, organisation, function allocation, etc. from a previous or existing system. Here the scenarios will focus on changes in the system, for example a move from 3 to 2 crew on an aircraft flight deck might suggest the use of scenarios where the role of the flight engineer is particularly tested.
- Situations that are independent of technology and systems support, taking a problem driven approach and focusing on situations that will arise whatever technological support is provided to human practitioners. For example, a move from conventional air traffic control to "free flight" may suggest scenarios focusing on air traffic conditions that are complex and hard to

[1] ASRS, the Aviation Safety Reporting System is a confidential incident reporting scheme run by NASA. More information is available from http://olias.arc.nasa.gov/ASRS/ASRS.html.

understand, *whatever* control regime and supporting technology is in place (see (Dekker and Woods 1997)).

2.3 When have you collected enough scenarios?

An obvious question to ask is whether a set of scenarios provides a "good enough" coverage of situations that could be encountered once the system is fielded so as to allow the designer to consider the most important requirements. For the moment, we leave this aspect of the selection to "expert judgement". In the case we are dealing with here, the actions themselves are highly critical and must be carried out in a short space of time. They do not, however, involve the crew in complex decision making, and at least some of the actions will be routine and well practised. We would need to balance this with other situations, for example, ones which involve much more complex reasoning, using detailed knowledge about the function of aircraft systems, in order to diagnose the cause of a systems failure.

For more information about the use of scenarios in system design, see:

- J. Carroll: *Scenario-Based Design: Envisioning Work and Technology in System Development.* (Carroll 1995)

We shall use two examples in this document. The first (Scenario 1) is fictitious and concerns the execution of navigation and flight management activities on a flight deck. The second (Scenario 2) is based on the state of the design of a multi-person crew flight deck.

2.4 Example Scenario 1

The first example scenario highlights some of the tasks carried out by the crew of a commercial airliner in making a change to the aircraft's flight path in order to comply with an air traffic control clearance. The scenario is an adaptation of one described by (Palmer, Hutchins et al. 1993). The focus will be looking at how successfully the flight management system plays its role in the scenario. In this section, we only present an overview of the description of this scenario (Figure 3), but the appendix contains more details.

Agents

The scenario takes place on the flight deck of a commercial airliner, flown by two flight deck crew, and also involves human Air Traffic Control (ATC) agents. The primary job of the two crew is to *pilot the aircraft* to the destination, *maintaining safety* and complying with the *instructions of ATC.*

Rationale

This scenario highlights an instance of a problem documented elsewhere. Cases of aircraft making "altitude deviations" by failing to respond in the expected way to ATC clearances constitute a substantial number of the cases reported under anonymous incident reporting systems such as ASRS.

Situation and Environment

The scenario involves making a change to a vertical flight plan in an aircraft equipped with a modern flight Management System (FMS). The scenario begins when the controller decides that an altitude restriction is necessary, and passes a new clearance on to the aircraft in the form of a target altitude that is to be achieved at a way point along the projected flight path. If it is possible to comply with the restriction, the pilot confirms this, and makes the necessary changes to the FMS.

Task Context

In the execution of this scenario, the pilots carry out a number of tasks and will need to draw on substantial task knowledge that they possess as the result of experience and training The tasks include communicating with ATC, selecting the new altitude in the altitude alert (to generate an alert when the target altitude is reached) and altering the flight path in the FMS.

System Context

The two pilots are supported in their work by modern electronic information displays. In particular, the scenario involves the altitude alert and FMS.

Action

This section describes a concrete sequence of events that unfold in the context already described. The description of the sequence of events records four primary aspects: the system status, the overt, physical actions (mostly inputs or communications) of the pilot flying and pilot not flying, and the system response. The system status includes information, such as warnings and other indications, that will be of use to the pilots. The system response is a record of the effect of the pilots' actions on the aircraft and avionics systems

In addition to this information it is useful to record the resources that are available and are used to guide the action (such as airmanship skills, written procedures, checklists, status displays, etc.).

Exceptional circumstances

An alternative course of action occurs if the pilots decide that they are unable to comply with the instruction from ATC.

Figure 3: Overview of Scenario 1

2.5 Example Scenario 2

Whereas the previous scenario described a situation using an already extant design and mode of using it, this scenario concentrates on one snapshot of an emerging design, and hypotheses about how it will be used. It is therefore impossible to rely directly on historical records of system operation and the problems that might arise (though these will provide useful background material). Instead, the scenario was used as a way of eliciting from experts (operators of a previous, similar flight deck as well as designers) how they think the scenario might unfold, and where they think the problems might be. One important difference between the new and old flight decks is that the size of the crew is reduced from three members to two: in future, there will be no flight engineer, and in order to compensate for the loss, the remaining two pilots will be assisted by more computerised technology.

This scenario, therefore, involves a situation where the activities of the flight engineer would, in the old flight deck, be particularly significant, since such a situation represents the greatest unknown quantity in terms of the combined performance of the system of pilots and new technology.

The scenario, described in Figure 4, concerns emergency conditions rather than normal operation, involving a number of tasks that in themselves are fairly simple and do not require a great deal of decision making on the part of the crew. In order to achieve more coverage in our analysis is advisable to look also at scenarios which involve more knowledge intensive activities such as fault diagnosis.

Agents

The proposed design will be flown by two flight deck crew (in contrast to the three currently present on the flight deck). The primary job of these two pilots is to fly the aircraft safely to their destination.

Rationale

This scenario is important as it involves activities in which, in the old system, the flight engineer was heavily involved. This will be a good test of whether the new technology can be an effective replacement for the knowledge and skills of the FE and the "spare cognitive capacity" available on a 3-person flight deck.

Situation and Environment

The starting conditions for this scenario is that the aircraft is at low level (200 feet, during daytime) over water, photographing a fishing vessel. To conserve fuel, the aircraft is flying on three engines: numbers 2, 3 and 4.

The aircraft suffers a massive bird strike on the right side, with two engines running. As a result of the bird ingestion in engines 3 and 4, both these engines fail, producing engine failure and engine fire warnings. The engine problems will cause the failure of the generators in these engines, which will, in turn lead to the remaining generators being overloaded, resulting in a series of warnings or cautions being signalled after a short delay.

Task Context

The crew must take immediate action in order to keep the aircraft flying, and will then commence the drills in response to the engine fire/failure and any secondary warnings that occur. The immediate response in order to keep the aircraft in the air will follow the following prioritisation: power; drag; trim; engine restart.

The pilot flying will attempt to gain altitude, though a single engine may not be sufficient to climb or maintain the current altitude; hence the importance of restarting the number 1 engine. After these actions have been carried out, the crew must carry out the engine fire and failure drills. Both consist of a combination of *immediate actions* and *subsequent actions*; typically, the immediate actions for all the current warnings will be carried out before proceeding to any of the subsequent actions.

System Context

The procedures above will be available on the electronic procedures format of the lower ECAM screen, as well as being written down in the flight reference cards (and, presumably in the pilots' memory).

Exceptional circumstances

See the more detailed description in the appendix of the actions that are carried out in this somewhat more complex scenario, and the alternative courses of action that are possible.

Figure 4: Overview of Scenario 2

3. Understanding the Task Context

In the description of scenarios above, tasks and task knowledge were highlighted as an important part of the ongoing activity. In this section we say a little more about how a person's tasks may be described. Many types of task analysis are described in the HCI literature, each with their different strengths and weaknesses. The error analysis process does not require any particular task analysis technique to be used, nor is any specific notation mandated for describing tasks. If an analyst or engineer applying THEA is familiar with a particular technique, or a task analysis has already been done as part of the project, then it's advisable to re-use as much work and expertise as possible. However, a number of features of a task description technique are desirable:

- Work is described in terms of the *agents* and *roles* that are responsible for carrying it out.
- With each role are associated the *goals* for which that role is responsible.
- Goals may be decomposed into lower level *sub-goals* and *actions*.
- Constraints on the order in which sub-goals and actions should be carried out are described by a *plan*
- The performance of tasks is triggered by *events*, produced by the environment or, the result of some internal cognitive process.

The technique known as Hierarchical Task Analysis (Kirwan and Ainsworth 1992; Shepherd 1989) possesses most of these features and is a useful way of understanding tasks; a variant is outlined below. However, in some cases, the approach can be simplified. For example, if the interaction is simple, it may be sufficient to write down the goals each operator will be engaged in, and the actions needed to achieve each goal — thus avoiding the complexity of HTA's plans and sub-goal hierarchies.

3.1 Hierarchical Goal Decomposition

Hierarchical Task Analysis (HTA) is a technique that can be used to describe operator's tasks in terms of the goals and sub-goals that the person is trying to achieve and the actions he or she uses to achieve these goals. It is *hierarchical* task analysis because task goals are broken down into a structure of sub-goals that have to be achieved in order that the top-level goal is satisfied. For example, the pilot's goal of changing course in Scenario 1 can be decomposed into sub-goals of receiving the clearance, confirming that it is possible to meet the clearance, effecting changes to the aircraft's flight path, and so on. These goals may themselves be decomposed into smaller sub-goals if it is deemed necessary.

One of the problems with carrying out an HTA is deciding at what level of detail to stop the hierarchical decomposition. In general there is no single answer to this question because it depends upon the purpose of the HTA. If the purpose is to consider training needs the analysis might well stop at a higher level than if the purpose is to consider what displays and controls an operator might need. Our purpose here, is to consider the possibility that the human operator will make a mistake in the performance of the task. Ultimately then, a complete analysis may well have to decompose the task down to the level of individual operator actions. However we argue that the process of error analysis is an iterative one and that error analysis can and should start with the fairly high level goals associated with the task. The particulars of a task will determine whether, once this high level analysis is done, there is a need to pursue all nodes in the hierarchy down to individual actions.

3.2 Plans

A goal decomposition describes how a problem can be broken down into simpler sub-problems, but says nothing about when the sub-problems must be addressed and in what order. Clearly, it's only possible to carry out some sub-problems in one order (a clearance can't be confirmed until it has been received), but for some cases, the order substantially affects the final outcome (such as making a change to the flight path without having received clearance). Given the importance of sequence and ordering, it is useful to introduce a special *plan* description to capture this information.

Plan description makes this ordering explicit and provides the analyst with additional power by allowing him or her to specify conditional goals. So for example a plan might include statements about what to do if a particular goal is not achieved (such as if clearance is refused). Plans can also be used to specify the triggering conditions under which certain optional sub-goals can be commenced. These may be failure conditions of either the system or the operator. If a plan description and a goal description has been done properly, every goal mentioned in the goal description should also be mentioned in the plan

and vice versa. In addition any restrictions on the order in which goals can be achieved should be mentioned in the plan. These two features can be used to check that the analysis has been done correctly. Plans therefore describe the flow of control through the task and document how the sub-goals and actions of a task are combined to satisfy the higher level goal. A notation you can use for describing plans is shown in Figure 5.

Conditional:	**if** <*condition*> **then** <*plan*>
Triggering:	<*condition*> **triggers** <*plan*>
Sequence:	<*plan*> **;** <*plan*>
Repetition:	**repeat** <*plan*> **until** <*condition*>

Figure 5: A notation for describing plans

3.3 Task Descriptions in Scenario 1

Rather than show a complete HTA description of the tasks carried out by all the agents in the scenario, we give a single example of a task carried out by the pilots: changing the course of the aircraft. In order to achieve this goal a plan is described, showing the temporal and causal relationships between the sub-goals. Since many of the tasks involve co-operation and co-ordination between agents, it is useful to record which agents and roles are involved in each goal. The example task description in Figure 6 shows three agents: PF (the pilot flying), PNF (the pilot not flying) and ATC (the Air Traffic Control facility). In complex information processing tasks, it is often useful to record in a separate table what information is processed in each sub-goal, how it is represented, and how it is propagated to other sub-goals.

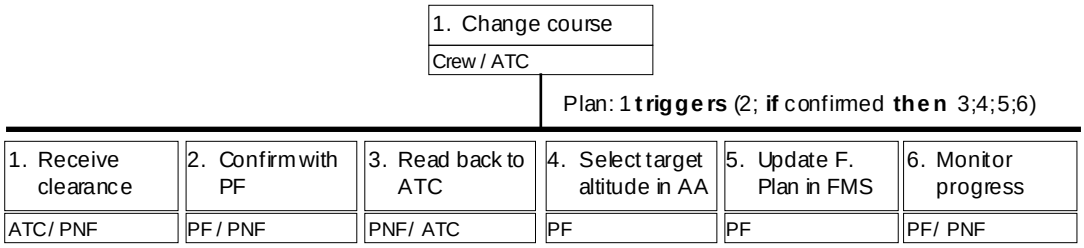

Figure 6: Example HTA task description

It should be emphasised that HTA is just one of many techniques for describing and analysing operators' tasks. HTA may not always be necessary, for example if interaction with the system of interest is relatively simple, then it is probably sufficient simply to identify the goals users have, and write down the list of the actions necessary to achieve the goals. If the interaction is more complex, then a more formal approach to capturing tasks and goals, such as HTA, may be needed. If you're familiar with techniques for doing this, or your organisation has "in house" standards or methods for task description, then they can be used within the THEA approach.

The following book surveys the area of task description and analysis, and the use of task analysis in design.

- Kirwan, B., and Ainsworth, L. (1992) *A Guide to Task Analysis*. (Kirwan and Ainsworth 1992)

4. Understanding the System Context

In the previous sections we have shown how work scenarios can help to envisage the ways in which a system being designed may be used. In this section we recognise the fact that the design of human-machine interfaces may contribute, positively or negatively, to the production of behaviour that is either "correct" or "erroneous". There are many aspects of interface design that can be used to inform an analysis of error, but in this section we consider only three: ways in which superficial aspects of an interface may confuse a user, the effects of restricting a user's authority, and problems arising from system moding.

This section suggests a number of issues the analyst might think about in interface design, which, when combined with a model of error, in the context of a particular usage scenario can help to understand where error problems might arise in the operation of a new system. The aim is not to give a detailed presentation of formalised techniques for analysing interfaces, but to help designers to raise some of the important questions. In order to answer them, designers' intuition and experience may well be sufficient, particularly when the interface is not especially complex. However, a number of other techniques may be applicable and are discussed in literature (for example, see (Harrison and Torres 1997)).

The aim is to show that if a design or design concept exists, then either by constructing models of the design and analysing the models, or simply by asking the right questions about the emerging design, we can uncover insights valuable to our error analysis process. Rather than providing a complete method for carrying out the analysis of design concepts, the aim is to give, largely by example, an understanding of what is required of such techniques. The reason for this is twofold. Firstly, it is not desirable to mandate the use of a particular technique, as expertise with others may be available in parts of the company. Secondly, it is desirable to allow the human error techniques to fit as seamlessly as possible into existing design processes and contexts, without forcing new notations and languages upon them.

4.1 Confusion and complexity

Perhaps the simplest and most obvious way of anticipating sources of "system induced" error is to look for places where an interface may be complex or may be a cause of confusion. We can ask a number of questions about a design that can help to expose the potential for problems. If the answer to any of the questions is "yes", then there may be an error problem with the interface in question.

Appearance — do displays or control panels look cluttered? are displays arranged so as to make the more important information and controls more difficult to find?

Complexity — are complex or fiddly command sequences, manipulations of data, or perceptual or mental operations necessary? Will users find it hard to understand or predict what the effects of carrying out commands or actions will be? Do actions have complex side effects?

Discriminability — are different controls made to look or feel the same? Are data that mean different things displayed in visually indistinguishable ways?

Consistency — are similar tasks carried out in different ways? Are similar data displayed in different formats using several forms of representation?

Affordance — does the appearance of controls obscure their function and method of activation? does the representation of data fail to make apparent the ways in which they can be manipulated?

This whole area of identifying areas where the appearance of an interface may be confusing, or where the superficial design of an interface may appear arbitrary with respect to its semantics is probably quite well catered for in current human engineering practice in the company.

4.2 Authority limiting

Another aspect of a system's behaviour that is highly relevant when considering the relationship between interfaces and errors is the way in which constraints are imposed on what the human is able to do to the system. Approaches like this which aim to limit the authority of the user to only a "safe" or "acceptable" influence on the system are often used to prevent or reduce the likelihood of particular errors, for example:

Lock-ins — prevent actions from being omitted from a sequence (example: can't take money from a cashpoint machine without removing the bank card).

Interlocks — prevent certain sequences from being carried out or certain states from being reached (example: in railway signal boxes, certain combinations of signal settings cannot be selected).

Guards — make certain high-consequence actions harder to perform or make them involve a number of sub-actions (example: physically guarding important switches, or requiring explicit confirmation of certain actions)

Protections — allow the human to carry out actions but limit the effect that they can have on the controlled process (example: an aircraft's flight control system can provide protection against stalling, overspeed, and so on).

4.3 Modes and mode transitions

A problem commonly reported to be a causal factor in accidents and incidents where safety may be at stake is *mode error* (e.g. see (Hughs and Dornheim 1995)). Broadly speaking, a mode error occurs when a system is operating in one of its possible modes, and the operator acts as if the system were in a different mode.

4.3.1 What's a mode?

By the term 'mode' we mean a configuration of the system that defines how it interprets user input, or how its output should be understood by the user.[2] If a system behaves in different ways (either because actions have different effects, or because outputs mean different things) at different times, then we say that the system can be in one of a number of modes at different times. Transitions between modes, and therefore between configurations of behaviour, can be caused by user actions, or by the system itself.

As an example, consider a manual data entry panel in an aircraft cockpit. The panel is designed to support a number of different data entry tasks, allowing the pilot to enter different types of information to several aircraft systems. Since the physical space available for this device is limited, all its functionality cannot be made available at once, and a number of modes are provided for carrying out the different tasks. A "Comms" mode exists for entering communications data: the numeric keys and the display are used to enter and present radio frequencies. Similarly, a "Nav" mode is provided for manipulating navigational data such as waypoints and headings. A number of buttons allow the current mode of the device to be changed.

The moding structure of a system can be made more opaque by the fact that modes can be decomposed into sub-modes. A simple example of this is where a "system" contains two or more moded devices. The mode of the whole system can be thought of as the composite of the modes of its sub-parts. Even a single device, though, can be several in modes concurrently. For example, a process control system can have a "training mode" and "operational mode". In addition to this, it may have "safe" and "emergency" modes. The whole system is then in a composite mode, e.g., "training" + "safe" mode.

5. Understanding Actions in Context

In Section 2.1, it was said that one of the principal components of a scenario is a description of the actions that take place. This can simply be written down as a list of actions and events, or as a trace or timeline. For example, the table in Figure 7 show some of the actions and sub-tasks that take place in the early part of Scenario 2, with time increasing in a downwards direction. What this shows is the actions performed by each agent (the two pilots and the "system") and also provides a place for describing what information will be used by the pilots to take the actions they do.

System status	Pilot flying	Pilot not flying	Information sources	System response
Engine 3 fire warning	Throttle 2 max.		Airmanship	Select ENG ECAM page
Engine 4 fail warning	Press master warning		Airmanship	
	Throttle 1 idle	Close bomb bay doors		
		Flap 0		
		Rudder trim		Start engine
	Throttle 1 max.	Warn crew		
	Navigate safe exit route	Throttle 3 Close	Eng fire 3 drill	
		LP cock 3 shut		
		Fire ext 3; shot 1		

[2] 'Mode' is also used by systems engineers to describe "internal" behavioural configurations of a system or external states of the environment. Our more user-centred definition is similar, but it is important to note that user interface modes need not be co-incident with internal system modes.

Figure 7: Partial action sequence from Scenario 2

What the tabular presentation also begins to highlight is the fact that the two pilots are, at the same time, doing different, possibly contradictory things (the pilot flying is attempting to restart engine to produce more thrust, while the non-flying pilot is shutting down the faulty engines, causing reduced thrust). The simple tabular presentation fails to capture the links between actions (opening and closing throttles) and the surrounding context (the goals to which the actions are directed) — which was one of the reasons for thinking about scenarios in the first place. As a remedy, we can describe the actions of the scenario and the order in which they occur, together with the goals (derived from the task analysis) to which they are directed. For example, the goal structured action sequence for Scenario 2 is shown in

Figure 8. The same actions as in Figure 7 are shown; in addition, however, are goals that drive the interaction and triggers that bring the goals into being.

Another distinction that Figure 8 makes clear is the distinction between work goals and the actions that contribute to them in a direct way (shown as light grey boxes) and interaction goals and actions (shown as darker grey boxes at the bottom of the figure). The so-called interaction actions do not contribute in a direct way to the accomplishment of the work-level goals, and are purely concerned with manipulating the user interface.

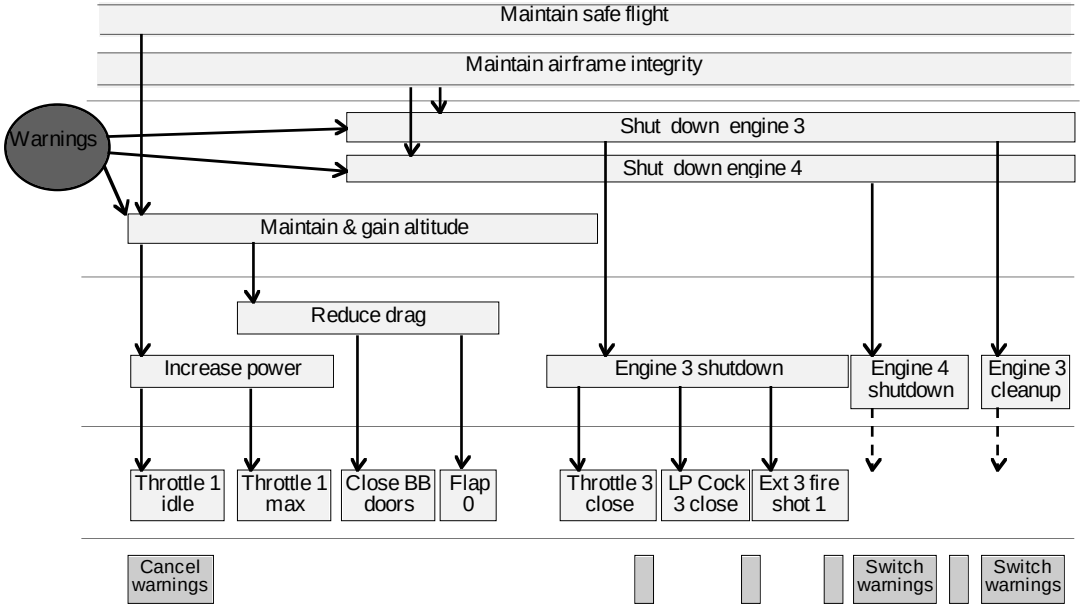

Figure 8: Hierarchical goal structuring of scenario actions

Presenting the scenario actions in this way shows a number of features of the scenario that were not immediately evident from any of the previous representations. In particular, it shows which goals and tasks become active and are active concurrently in the scenario (not present in a task analysis like Figure 6, which shows only a single task), and which actions are related by being directed towards the same goals (not present in a simple event listing such as Figure 7 which makes no mention of goals).

6. Understanding Operator Error

The error identification process that will be described in the next part of the document is based on two views of how human behaviour can be described. In this part, we describe these two views. By describing them as input, we are aiming to suggest that they are two possible techniques and that other explanations for the cause of human failure might also be used as an input to this analysis process. This material forms part of the error model. Here we assume on the one hand that a user's actions arise as emergent behaviour of a cognitive system comprising the user's internal cognitive processes, the objects of the user's work, interactive systems, and other human agents. On the other hand, human behaviour can be described simply in terms of the physical (and possibly cognitive) actions that are observed or assumed to take place without much regard to the processes and mechanisms by which these actions are generated. Both views have their place in error analysis, and lead to different views of the nature of error. In fact we shall use the two techniques in combination.

6.1 Cognitive failure

Errors can be regarded as failures in cognitive processing. Figure 9 shows an outline of a variant of the execution-evaluation model of human information processing (Norman 1988).

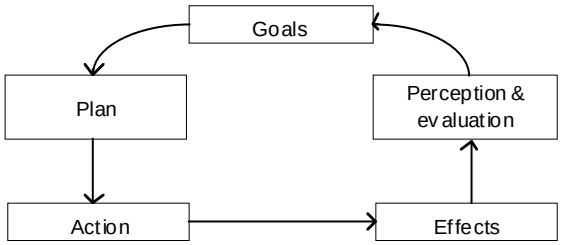

Figure 9: Cyclic model of human information processing

From this, we can identify a number of **cognitive failures** or ways in which human information processing can fail, possibly resulting in "incorrect" behaviour.

- Failures in the triggering and activation of goals (goals not triggered at the right time, the wrong goal being activated, or goals being lost).
- Failures in the goals themselves (goals not achievable in the current conditions, or sets of goals that are in conflict)
- Faulty plans (plan that fail to achieve the goal or whose execution is impossible).
- Failures to execute actions adequately (e.g., "slips" or "lapses" where an action is missed or carried out incorrectly).
- Perceptual failures (failure to see what the effect of an action is or failure to notice some external event or condition).
- Failures of interpretation and evaluation of perceptions (incorrect interpretation of perceived data, failure to realise when a goal has been completed).

Some examples of types of cognitive failure are shown in Figure 10.

Stage	Cognitive failure mode	Example
Goals	Lost goal	In Scenario 2: Forget to return to engine fire "cleanup" actions; fail to notice and act on a warning (trigger).
	Unachievable goal	Aim to make impossible course change (Scenario 1).
	Conflicting goals	Conflict between goals to maintain thrust and to shut down engine (Scenario 2).
Plans	Faulty or wrong plan	Mis-remember action sequence for programming flight management computer (Scenario 1); close the wrong engine (Scenario 2).
	Impossible plan	Plan involving the selection of a menu item that does not exist.
Actions	Action slip / lapse	Forget action or sequencing; fail to carry out action correctly.
Perception, interpretation	Failure to perceive correctly	Mis-read the current setting in the altitude alert window.
	Mis-interpretation	Read a value from the MCP and interpret it as angle of descent (instead of vertical speed).

Figure 10: Examples of cognitive failure

In Part II, we will ask questions about the performance of each of the cognitive components in relation to the use of the system, in order to try and anticipate where cognitive failures might occur and lead to behavioural errors.

6.2 Deviations from expected behaviour

In the behavioural view of error, we describe errors in terms of deviations from some prescribed or normal course of action. In doing this it is useful to guide the search for error problems by a set of "keywords" that capture classes of behavioural deviation (cf. techniques such as HAZOPS (Kletz 1992)). A useful set of keywords (based on those used in the nuclear power industry) is shown in Figure 11.

Keyword	Description	Example
Omission	Fail to carry out an action or the actions associated with a sub-goal.	In Scenario 1, fail to enter the target altitude in the altitude alert.
Commission:		
•Incorrect	Carry out the correct action or sub-goal, but to so incorrectly.	???
•Substitution	Substitute an incorrect action or item of data for a correct one.	Shut down the wrong engine in response to a fire warning.
•Insertion	Insert an extraneous action into the stream of behaviour.	??
Sequence	Perform the right actions or sub-goals, but in the wrong order.	??
Repetition	Repeat actions or sub-goals unnecessarily,.	??
Quantitative	Carry out a correct action, but with some "quantitative error" (too much / too little / too long / too short etc.)	??

Figure 11: Keywords for describing error types

Two of the best books on human error and its causes are:

- J. Reason: *Human Error*.(Reason 1990)
- D. Norman: *The Psychology of Everyday Things*. (Norman 1988)

A much wider perspective on the nature and causation of human error is reflected in

- Woods, D. et al.: *Behind Human Error: Cognitive Systems, Computers and Hindsight*. (Woods, Johannesen et al. 1994)

Part II. Analysis

Part I identified a number of factors that provide us with an understanding of the context in which human action — and therefore error — takes place. In this section, we describe how these pieces of information can be drawn together in an analysis technique that helps in identifying where human error may be a problem.

The approach is based around a set of questions based on the failures suggested by the model of human information processing described in Section 6.1. The questions help a designer or analyst to envisage ways in which things may go wrong, leading to a failure in cognitive processing. Once a potential cognitive problem has been identified, it is possible to think about how that failure will be manifested in "incorrect" behaviour, and what the ultimate effect on the state of the entire system will be.

The answer to these questions will, in general, be found in the scenario, described in the way suggested in Part I.

7. The Error identification process

The next sub-section contains a list of questions that a designer can ask about a scenario in order to help uncover places in the scenario where cognitive failure modes may occur. This is a preliminary step in a process of identifying possible cognitive failures, and tracing their effects through behavioural failures to an impact on the task or system being controlled, as in Figure 12.

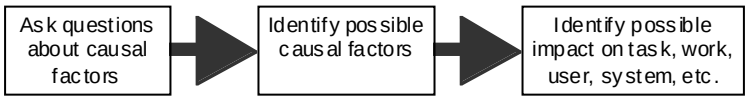

Figure 12: Identification of potential errors

Precisely how the questions are asked and the analysis is carried out is largely a matter of choice, but one possibility is to use the structure of the scenario action description (as in Section 5) to guide the enquiry process. In other words, follow the goal hierarchy structure from top to bottom asking each question about each goal or action. Clearly, this is likely to be a very lengthy and time consuming process, involving a lot of replication. Another option is to ask the questions about the whole scenario, and to use them to find error problems.

In other words, the analyst may use the questions to select parts of the scenario where problems might arise, then conduct a more detailed analysis of behavioural error and impact where appropriate. In some cases there will clearly be the potential for a cognitive failure, but with no obvious behavioural manifestations. A good example of this is where goals come into conflict: it is often not at all clear what the behavioural implications of a conflict will be, though the problem is still potentially serious (particularly if the goals involved are important or their resolution may be complex). In such cases, the cognitive failure can be taken to be the "problem" to which a design solution may be sought.

The results of the analysis can be recorded in a fairly ad hoc way, depending on the requirements of the project at hand. However, it has proved useful in some situations to record the results in a tabular form similar to that illustrated in Figure 13.

Question	Causal issues	Consequences	Design issues
Identifier of the question (as an aid to traceability)	Issues raised by the analyst as a result of asking the question.	Consequences of the causal issue. These can take a number of forms: cognitive failures of behavioural errors whose likelihood may be increased; additional cognitive or behavioural work that might be generated; effects of the task and work; impact on the system (particularly from a safety point of view).	Notes, suggestions, comments, re-design ideas.

Figure 13: Format of tables for recording error analysis results

7.1 Applying the cognitive error analysis

The questions are based in the failures that are possible in the execution-evaluation cycle model of human information processing.

Questions	Consequences	Examples & design questions
Goals, Triggering and initiation		
G1. Are items triggered by stimuli in the interface, environment, or task?	If not, goals (and the tasks that achieve them) may be lost, forgotten, or not activated, resulting in **omission** errors.	Are triggers clear and meaningful? Does the user need to remember all the goals?
G2. Does the user interface "evoke" or "suggest" goals?	If not, goals may not be activated, resulting in **omission** errors. If the interface does "suggest" goals, they may not always be the right ones, resulting in the **wrong goal** being addressed	E.g.: graphical display of flight plan shows pre-determined goals as well as current progress.
G3. Do goals come into conflict?	If so additional cognitive work (and possibly errors) may result from resolving the conflict. If the conflict is unresolvable, one or more goals may be lost, abandoned, or only partially completed.	Can attempt to design out conflicts or give participants the resources to resolve them.
G4. Can a goal be achieved without all its "sub-goals" being correctly achieved?	The sub-goals may be lost (resulting in **omissions**).	E.g.: goal of photocopying achievable without sub-goal of retrieving card.
Plans		
P1. Are there well practised and pre-determined plans?	If a plan isn't well known or practiced then it may be prone to being forgotten or remembered incorrectly. If plans aren't pre-determined, and must be constructed by the user, then their success depends heavily on the user possessing enough knowledge about their goals and the interface to construct a plan. If pre-determined plans to exist and are familiar, then they might be followed inappropriately, not taking account of the peculiarities of the current context.	
P2. Can actions be selected in-situ, or is pre-planning required?	If the correct action can only be taken by planning in advance, then the cognitive work may be harder. However, when possible, planning ahead often leads to less error-prone behaviour and fewer blind alleys.	
P3. Are there plans or actions that are similar to one another? Are some used more often than others?	A more common but similar plan may be confused for the intended one, resulting in the substitution of an entire task or sub-task.	
Performing actions		
A1. Is there physical or mental difficulty in executing the actions?	Difficult, complex , or fiddly actions are prone to being carried out incorrectly.	
A2. Are some actions made unavailable at certain times?		
A3. Is the correct action dependent on the current mode?	Creates a demand on the user to know what the current mode is, and how actions' effects differ between modes. Problems with this knowledge can manifest themselves as a **substitution** of one logical action for another.	

A4. Are additional actions required to make the right controls and information available at the right time?	The additional goals may be lost (resulting in **omissions**) and users will be unable to carry out the main goals. The overall effect may be to cause **confusion** and disorientation for the user.	
Perception, Interpretation and evaluation		
I1. Are changes (resulting either from user action or autonomous system behaviour) perceivable?	If changes are not perceivable, the user must retain a mental model of the system state. Particularly problematic if changes happen autonomously.	
I2. Are the effects of actions perceivable immediately?	If there's no feedback that an action has been taken, the user may **repeat** actions.	
I3. Does the item involve monitoring, vigilance, or continuous attention?	The user's attention can easily be diverted away from monitoring tasks, meaning that changes that confirm goals achievement (leading to **repetition** of actions or carrying out actions **too late**) or that trigger new goals may be missed (resulting in omission of the associated actions).	
I4. Can the user determine relevant information about the state of the system?	If not, the user will have to remember the information they require, thus making it prone to being lost or recalled **incorrectly**.	
I5. Is the relation of information to the plans and goals obvious?	If the relationship to plans isn't clear, then a source of feedback about correct execution of the plan, and therefore a factor that mitigates against error, is lost. If the relationship to goals is unclear, then the user may be unaware of when a goal is achieved, leading to termination of a sub-task **too early** or **too late**.	
I6. Is complex reasoning, calculation or decision making involved?	If cognitive tasks are complex, they may be prone to being carried out **incorrectly**, to being the cause of other tasks carried out **too late**, or to being **omitted** altogether.	
I7. Is the correct interpretation dependent on the current mode?	Creates a demand on the user to know what the current mode is, and to how the appropriate interpretation of information differs between modes. Problems with this knowledge can manifest themselves as a **substitution** of one logical information item for another.	

7.1.1 Determining Causal and Mitigating factors

The "Causal issues" column of the table (Figure 13) will be filled in a fairly unsystematic way with factors that are likely to influence a human agent's predisposition to make errors in either a positive or negative way. The questions forming the "checklist for cognitive analysis" will be used as a guide to the kind of things that it will be useful to write down. Some remarks are made at the end about the list of questions and some additions that could be made to it.

Within the context of the causal analysis described above particular error forms described by the behavioural error keywords (Figure 11) may be considered. The purpose of the keywords is not particularly to define what the error is, but is to act as a trigger for the analyst to think of the ways in which a task can fail (cf. HAZOPS — see (Kletz 1992)). A few things are worth noting. Not every keyword will make sense in the context of every scenario (for example, because physical constraints make it impossible, or because its hard to imagine either how such a deviation could occur or what it would mean. E.g. repetition error of an aircraft's take-off sequence).

For a particular task, a guide word may have a number of different interpretations. In particular, it may refer to deviations on the *function*, *target* or *data* of the task. For example, consider a task like entering the altitude value into the altitude alert window. For such a task, three possible interpretations of the **substitute** keyword are possible:

- Doing something other than entering the data (such as comparing with what's already displayed there);
- Targeting the task at another object (entering the data into a different device);
- Substituting another piece of data (entering the distance value instead of the altitude).

Generally, the commission-type errors (**substitute**, **incorrect** and **insert**) are fairly problematic because they aren't very constraining as guides. In other words, there are generally a large number of substitutions, insertions etc. that could possibly take place, and the keyword method leaves the analyst without many clues. Maybe it's in the area of commission errors where the more cognitive analysis could be more helpful.

The "Consequences" column serves a number of purposes for the analyst: specifically, it can be used to record the consequences that the identified causal issue might have on the performance of the work and the successful outcome of the scenario, on the workload of the participants in the scenario, and on the state of the systems involved and the hazardous conditions that might result.

Finally, the "Design issues" column provides the analyst with a space for documenting ideas about how the design could be changed to avoid some of the problems that have been identified.

8. An analysis example

An illustration of how the analysis may be conducted is shown in Figure 14. Only two of the questions from the list above are shown (G1, about the mechanisms that trigger or activate goals, and G3 about the potential for conflicting goals). Asking question G1 about Scenario 2 yields a number of possible answers, since different collections of goals have different triggering properties. Some are fairly innocuous and do not suggest potential problems (e.g., "Shut down engine" is triggered quite directly by the warning) whereas others are less directly triggered and may be more prone to being omitted e.g., "Engine 3 cleanup").

A more complete version of the analysis is included in the Appendix, Section 11.

Question	Causal issues	Consequences	Design issues
G1	Many goals triggered fairly directly (e.g., "Shut down engine 3"). Timing of lower level goals arises as a combination of triggering and group decision making (e.g., Engine 3 shutdown). Some goals rely on general airmanship skills for their activation (e.g., power, drag). Some goals poorly triggered, especially if there are several goals with only a single trigger on the display (e.g., "Engine 4 shutdown" or "Engine 3 cleanup").	Main behavioural consequence is that triggers for cleanup actions exist in the display, but are removed when other tasks intervene (switching to "Engine 4 shutdown" removes indications for "Engine 3 cleanup"). It's possible that "Engine 4 shutdown" or "Engine 3 cleanup" might be omitted or delayed.	
G3	Goals to Increase power and Engine 3 shutdown are in conflict (though this is inevitable).	Resolving the conflict satisfactorily requires negotiation between PF and PNF. The time required for this negotiation may lead to a non-optimal (too late) decision.	

Figure 14: Example application of error questionnaire to Scenario 2

9. References

Carroll, J. M., Ed. (1995). *Scenario-Based Design: Envisioning Work and Technology in System Development*. J. Wiley and Sons.

Dekker, S. and D. Woods (1997). *Management by exception in future air traffic control: An empirical study of coordination in an envisioned distributed system.* DCSC Technical Note DCSC/TN/97/25.

Fischer, U., J. Orasanu and M. Wich (1995). *Expert Pilots' Perceptions of Problem Situations*. Ed. 8th International Symposium on Aviation Psychology, Ohio State University, Columbus, Ohio,

Greenbaum, J. and M. Kyng, Ed. (1991). *Design at Work: Cooperative Design of Computer Systems*. Design at Work: Cooperative Design of Computer Systems. Lawrence Erlbaum Associates Inc.

Harrison, M. D. and J. C. Torres, Ed. (1997). *Proceedings, 4th Eurographics Workshop on Design, Specification, Verification of Interactive Systems*. Springer Computer Science. Springer Wien New York.

Hollnagel, E. (1993). *Human Reliability Analysis — Context and Control.* Academic Press.

Hughs, D. and M. A. Dornheim (1995). *Automated Cockpits Special Report, Part 1. Aviation Week & Space Technology.* 52-65.

Kirwan, B. (1992). Human error identification in human reliability assessment. Part 1: Overview of approaches. *Applied Ergonomics* **25**(5): 299-318.

Kirwan, B. (1994). *A Guide to Practical Human Reliability Analysis.* Taylor & Francis.

Kirwan, B. and L. K. Ainsworth (1992). *A Guide to Task Analysis.* London, Taylor & Francis.

Kletz, T. (1992). *HAZOP and HAZAN: Identifying and Assessing Process Industry Hazards.* Institution of Chemical Engineers.

Norman, D. A. (1988). *The Psychology of Everyday Things.* Basic Books.

Palmer, E. A., E. L. Hutchins, R. D. Ritter and I. v. Cleemput (1993). *Altitude Deviations: Breakdowns of an Error Tolerant System.* NASA Technical Memorandum DOT/FAA/RD-92/7.

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

Shepherd, A. (1989). Analysis and training in information technology tasks. D. Diaper, Ed. *Task Analysis for Human-Computer Interaction*. Ellis Horwood. 15-55.

Woods, D. D., L. J. Johannesen, R. I. Cook and N. B. Sarter (1994). *Behind Human Error: Cognitive Systems, Computers and Hindsight.* CSERIAC State-of-the-Art Report SOAR 94-01.

Part IV. Appendix

10. Detailed scenario descriptions

10.1 Scenario 1

The first example scenario highlights some of the tasks carried out by the crew of a commercial airliner in making a change to the aircraft's flight path in order to comply with an air traffic control clearance. The scenario is adapted from a description by (Palmer, Hutchins et al. 1993). The focus will be looking at how successfully the flight management system plays its role in the scenario.

10.1.1 Agents

The proposed design will be flown by a flight deck crew of two and also involves human agents who are doing the tasks associated with air traffic control. The primary job of these two pilots is to fly the aircraft safely to their destination and since we are concerned with the role of the flight management system.

10.1.2 Rationale

This scenario highlights an instance of a problem documented elsewhere. Cases of aircraft making "altitude deviations" by failing to respond in the expected way to ATC clearances constitute a substantial number of the cases reported under anonymous incident reporting systems such as ASRS.

10.1.3 Situation and Environment

The example is based on the problem of making a change to a vertical flight plan in an aircraft equipped with a modern flight management system (FMS). A typical sequence of events described by these tasks is as follows.

- Either the pilot requests an altitude change (for example, to avoid turbulence), or the controller decides that an altitude restriction is necessary.
- The controller passes the clearance on to the aircraft in the form of a crossing restriction specified by an altitude and an offset from (either before or after) a way point on the current flight plan
- If it is possible to comply with the restriction, the pilot confirms this, and makes the necessary changes to the FMS.

Air traffic congestion around terminal arrival areas frequently prompt late or sudden changes to previously planned flight paths. In this instance, the change is initiated by Air Traffic Control (ATC), who request aircraft to descend to a particular altitude at a particular point on the flight path. The aircraft is instructed to descend so as to reach an altitude of 11000 feet, at a point 20 nautical miles before the next waypoint in the previously planned flight path.

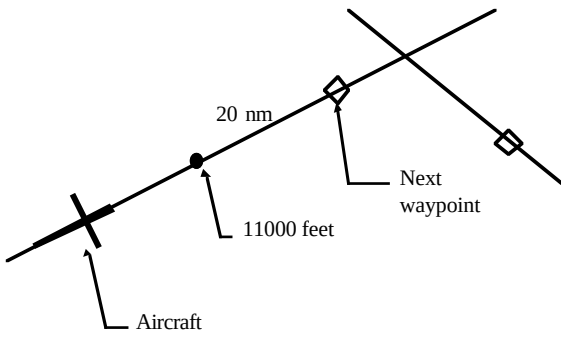

Figure 15: Navigation scenario

10.1.4 Task context

In the execution of this scenario, the pilots will need to draw on substantial task knowledge that they possess as the result of experience and training. The tasks that the pilots are required to carry out, and the order in which they should occur are shown in Figure 16. We are less concerned with the tasks of ATC because these activities do not directly involve the technology that is being analysed using the technique. For certain tasks, particularly those carried out in response to emergency conditions (see Scenario 2 below for an example) some of the task knowledge may be captured in a more durable form by written procedures in a Quick Reference Handbook or on an ECAM-type display.

	Agent	Tasks
1.	Pilot Not Flying	Informed of clearance by ATC
2.	Pilot Not Flying	Confirm receipt of clearance with Pilot Flying
3.	Pilot Flying	Dial in new altitude
4.	Pilot Flying	Set in new flight path
5.	Pilots Flying/Not Flying	Monitor execution of altitude/flight path change

Figure 16: Tasks carried out in Scenario 1

10.1.5 The System Context

The two pilots are supported in their work by modern electronic information displays.

10.1.6 Scenario Actions

This section describes a concrete sequence of events that could plausibly unfold given what has already been described. The description of the sequence of events records four primary aspects of the unfolding action: the system status, the actions of the pilot flying and pilot not flying, and the system response. The system status includes information, such as warnings and other indications, that will be of use to the pilots. The pilot actions are overt, physical acts (mostly inputs or communications) carried out by one or other pilot. The system response is a record of the effect of the pilot's actions or the behaviour of some automated component of the system.

In addition to this information, the tabulation of the action in this scenario also records some of the explicit conditional and time dependent parts of the action (though there are only a few instances); and resources that are available used to guide the action (principally airmanship skills or written procedures and checklists). This description of the flight deck presumes a two crew configuration.

System status	Pilot flying	Pilot not flying	Information sources	System response
ATC request to change altitude	Fly the aircraft	Receive clearance	Automation, Training	
		Acknowledge clearance		
	Pull altitude select button on MCP			
	Enter 11,000 in altitude alert window on MCP			
	Line select VOR in FCU			Create new waypoint
	Add characters "/-20"			
	Line select to 1 left			
	Enter "/110" in scratchpad			
	Line select to 1 right			
	Press CDU EXEC	Monitor execution of changes		Flight path changed

10.1.7 Exceptional circumstances

An alternative course of action occurs if the pilots decide that they are unable to comply with the instruction from ATC.

10.2 Scenario 2

Whereas the previous scenario described a situation using an already extant design and mode of using it, this scenario concentrates on one snapshot of an emerging design, and hypotheses about how it will be used. It is therefore impossible to rely directly on historical records of system operation and the problems that might arise (though these will provide useful background material). Instead, the scenario was used as a way of eliciting from experts (operators of a previous, similar flight deck as well as designers) how they think the scenario might unfold, and where they think the problems might be. One important difference between the new and old flight decks is that the size of the crew is reduced from three members to two: in future, there will be no flight engineer, and in order to compensate for the loss, the remaining two pilots will be assisted by more computerised technology.

This scenario, therefore, involves a situation where the activities of the flight engineer would, in the old flight deck, be particularly significant, since such a situation represents the greatest unknown quantity in terms of the combined performance of the system of pilots and new technology.

The scenario, described below, concerns emergency conditions rather than normal operation, involving a number of tasks that in themselves are fairly simple and do not require a great deal of decision making on the part of the crew. In order to achieve more coverage in our analysis it is advisable to look also at scenarios which involve more knowledge intensive activities such as fault diagnosis.

10.2.1 Rationale

This scenario is important as it involves activities in which, in the old system, the flight engineer was heavily involved. This will be a good test of whether the new technology can be an effective replacement for the knowledge and skills of the FE and the "spare cognitive capacity" available on a 3-person flight deck.

10.2.2 Situation and Environment

The proposed design will be flown by two flight deck crew (in contrast to the three currently present on the flight deck). The primary job of these two pilots is to fly the aircraft safely to their destination. The two pilots are supported in their work by modern electronic information displays, similar to those on civil airliners such as the Airbus series. These show status information, system pages, warning information and procedures to be carried out in response to the warnings (the details of some of these will be discussed later).

The operational requirements for this aircraft, however, mean that tasks carried out by the aircraft, and therefore the work of the crew, differ markedly from those on an airliner. One example is that the mission may require the aircraft to fly at low altitude over the sea for significant time periods. Another example is the requirement to remain on task for long periods, resulting in fuel saving strategies like operating on only three engines, and continuing with a mission even if the aircraft has suffered minor failures.

The starting situation for this scenario is the aircraft at low level (200 feet, during the daytime) over water, photographing a fishing vessel. In order to conserve fuel, the aircraft is flying on only three engines: numbers 2, 3 and 4[3] . The aircraft suffers a massive bird strike on the side with two operational engines (it is common practice, under certain conditions, when making a pass to photograph a boat, to present the side with two running engines to the boat). As a result of the bird ingestion in engines 3 and 4, both these engines fail, producing engine failure and engine fire warnings. The engine problems will cause the failure of the generators in these engines, which will, in turn lead to the remaining generators being overloaded, resulting in a series of warnings or cautions being signalled after a short delay.

The primary problems (the engine failures) have a number of knock-on effects, leading to secondary warnings, and in order to get the complete picture, these must be considered too. One such example is the failure of generators connected to the failed engines, and the subsequent partial loss of power.

---

[3] Engines are numbered left to right 1-2-3-4.

Loss of generator The loss of the generators associated with failed engines will almost certainly result in generator failure warnings and generator overload warnings for the remaining generators, creating the need for electrical load to be shed. The automation will attempt to do this by shutting down non-essential equipment (e.g., in the mission systems area). However, the crew may elect to shut down certain equipment that is known to be unnecessary from the point of view of the mission at hand (such as the acoustics consoles) while leaving intact the power supply to systems that are judged more important (the radar console, say).

10.2.3 Task Context

In a situation such as this, the crew will tend to take some immediate actions in order to keep the aircraft flying, and will then commence the drills in response to the engine fire/failure and any other secondary warnings that might occur. The immediate response in order to keep the aircraft in the air will follow the following prioritisation: power; drag; trim; engine restart.

Power	Maximum throttle on the remaining engine (2).
Drag	Close external doors etc.
Trim	From one side to the other.
Engine Restart	No. 1 throttle forward past the trigger point – Autothrottle / FADEC starts the engine.

While this is going on, the pilot flying will attempt to gain altitude, though a single engine may not be sufficient to climb or maintain the current altitude; hence the importance of restarting the number 1 engine. After these actions have been carried out, the crew begins to carry out the engine fire and failure drills. Both consist of a combination of immediate actions and subsequent actions; typically, the immediate actions for all the current warnings will be carried out before proceeding to any of the subsequent actions. As an example, the engine fire drill, in roughly the form it appears in the Flight Reference Cards, is shown in Figure 17.

ENGINE FIRE PROCEDURE		
Immediate Actions		
1.	Crew	Warned
2.	Throttle No	Close past trigger point
3.	LP cock	No … SHUT
4.	Fire extinguisher	Lift guard, fire 1st shot
Subsequent actions		
1.	Lookouts	Manned and report
2.	Check systems page	
3.	Generator	No… OFF Busbars reconfigured
4.	Hydraulic map	No … OFF
5.	ECS	No … Engine bleed OFF. ECS reconfigured
6.	If after 30 sec warning persists	Fire 2nd shot
When warning ceases		
Check generator loading, load shed if necessary Check ECS operating from unaffected side.		

Figure 17: Engine fire drill, as in the Flight Reference Cards or QRH

10.2.4 System Context

The procedures above will be available on the electronic procedures format of the lower ECAM screen, as well as being written down in the flight reference cards (and, presumably in the pilots' memory). A number of differences exist between the computerised and paper versions and this forms one aspect of the redesign:

- Since the warnings system makes no attempt to monitor the pilots' actions, a facility is provided for "checking off" actions as they are carried out (the pilot uses a switch located on

the side console to indicate that an item has been carried out). Items may be skipped over without being checked, and it is possible to return and check them later.

- Although several procedures will be active (i.e., will correspond to a current warning, and having uncompleted actions), only one can be displayed at once. The pilot can select from a list, (also on the procedures format) which of the several available procedures is to be displayed.
- The old design provides flight reference cards. These may possibly be carried over to the new design and the crew will use some mechanism for distinguishing between "immediate" and "subsequent" actions. However it is also possible that this information be recorded in the electronic checklists.

Certain other information that is present in the reference cards may not be replicated in the electronic display (e.g., that the second fire extinguisher should be operated 30 seconds after the first).

10.2.5 The Scenario Actions

The preceding subsections have described some of the situation in which a crew may find themselves, and some of the task knowledge which is presumed in the context of the flight deck. This section describes a concrete sequence of events that could plausibly unfold given the contextual factors already described.

The description of the sequence of events records four primary aspects of the activity as it unfolds: the system status, the actions of the pilot flying and pilot not flying, and the system response. The system status includes information, such as warnings and other indications, that will be of use to the pilots. The pilot actions are overt, physical acts (mostly inputs or communications) carried out by one or other pilot. The system response is a record of the effect of the pilot's actions or the behaviour of some automated component of the system.

In addition to this information, the tabulation of the action in this scenario also records some of the explicit conditional and time dependent parts of the action (though there are only a few instances); and resources that are available used to guide the action (principally airmanship skills or written procedures and checklists). This description of the flight deck presumes a two crew configuration.

System status	Pilot flying	Pilot not flying	Info. sources	System response
Engine 3 fire / Engine 4 fail	Fly the aircraft		Airman-ship	Select ENG ECAM page
	Throttle 2 max.		Airman-ship	
	Press master warning	Close bomb bay doors		
		Flap 0		
	Throttle 1 idle	Rudder trim		Start engine
	Throttle 1 max.	Warn crew		
	Navigate safe exit route	Throttle 3 Close past trigger position	Eng fire 3 drill	
		LP cock 3 shut		
		Fire ext 3; shot 1		
		Throttle 4 Close past trigger position	Eng fire 4 drill	
		LP cock 4 shut		
		Lookouts manned and report		
		Check systems page		
		Lookouts manned and report		
		Gen 3 OFF Busbars reconfigured		
		Hydraulic pump 3 OFF		
		ECS 3 Eng bleed OFF		
		ECS reconf		

		Fire ext 3: fire 2nd shot Gen 4 OFF Busbars reconfigured Hydraulic pump 4 OFF ECS 4 Eng. Bleed OFF ECS reconf		Warning ceases
Gen 3, 4 failure Gen 2 overload		Shed load Warn crew Monitor voltages and frequencies 115 v transformer (right) COUPLE Yaw damper on Check CPU fault lights Check services lost **Busbars couple Generator 2 Check kW/kVAR Determine faulty generator	Double gen fail drill Gen overload drill	Select ELEC ECAM page

This presentation of the scenario actions does not, of course, make a connection between the actions that are carried out and the goals that they are intended to achieve. However, this connection can be made and documented in diagram of the kind shown in Figure 8 (which covers only a part of this scenario, but is not completed here).

10.2.6 Exceptional circumstances

There are a number of possible variations on this scenario, and here we list a few of them, without going into the full details of the actions that occur.

Failure of Hydraulics Pumps

The scenario can be made more complex (and more difficult for the crew members involved) by considering additional tasks arising from secondary failures (i.e., failures that are themselves caused by the primary problems of engine failure and fire). The generator-related tasks already discussed come into this category, and another example is the failure of the pumps, driven by the failed engines, which pressurise the hydraulics systems.

Additional navigation tasks

The geography of the area in which the incident occurs can make this scenario even more hazardous than it already is, and can burden the flight deck crew with even more tasks. For instance, if the birdstrike occurs close to land, then the business of navigating safely away from the area is made rather more critical and complex.

Unsuccessful fire drill

The drills, even if carried out correctly, can fail to be effective in a number of ways. For example, it is entirely possible that the fire extinguishers will not be adequate to put the fire out; the crew may be unable to restart the number 1 engine; the aircraft may be heavy, and therefore unable to gain altitude. All of these conditions will result in the crew (or the captain) considering whether or not to ditch the aircraft, in which case a completely different set of tasks concerned with evacuation will be carried out.

11. Error analysis example

The table below shows the results obtained when the full set of error analysis questions are asked about Scenario 2. A number of the questions yield several “causal issues” being raised. The consequences of these (on the work, the scenario, the users and the state of other systems) are then documented for the problematic cases in the “consequences” column. Entries for some questions have been left blank, indicating that the question didn’t appear to reveal any interesting insights. The “Design issues” column has been intentionally left blank in the current example.

Question	Causal issues	Consequences	Design issues
G1	1. Many goals triggered fairly directly (e.g., "Shut down failed engines"). 2. Timing of lower level goals arises as a combination of triggering and group decision making (e.g., Engine 3 shutdown). 3. Some goals rely on general airmanship skills for their activation (e.g., power, drag). 4. Some goals poorly triggered, especially if there are several goals with only a single trigger on the display (e.g., "Engine 4 shutdown" or "Engine 3 cleanup").	Main behavioural consequence (4) is that triggers for cleanup actions exist in the display, but are removed when other tasks intervene (switching to "Engine 4 shutdown" removes indications for "Engine 3 cleanup"). It's possible that "Engine 4 shutdown" or "Engine 3 cleanup" might be omitted or delayed.	
G2			
G3	Goals to Increase power and Engine 3 shutdown are in conflict (though this is inevitable).	Resolving the conflict satisfactorily requires negotiation between PF and PNF. The time required for this negotiation may lead to a non-optimal (too late) decision.	
G4			
P1	Most functional aspects of the tasks will be well practiced and planned in advance. Less well planned are interactions with the technology and management of the various goals. E.g. Breaking off from Engine 3 tasks to do engine 4 ones, and resuming the engine 3 tasks later.	1. At the level of actions, plan following is well supported, but at the level of goals (e.g. Eng 4 shutdown) prioritisation and interleaving is not well practiced. 2. The fact that actions are well planned may make prioritisation more error prone.	
P2	Interaction will tend to be a mixture of pre-planned procedure following (how to shut down an engine) and on the fly decision making (when to shut the engine down).	See P1. Because the time of shutdown can't be planned in advance, it is prone to errors in on-the-fly decision making.	
P3	Engine 3 fire & engine 4 failure similar and engine fire procedure more well practiced.	Actions from engine fire procedure may be done on engine 4. But this is a superset of engine failure actions.	
A1	Work tasks not problematic, but interface tasks (e.g. checking off actions) are awkwardly located.	May omit, or repeat.	
A2	Once a fire extinguisher shot has been used, it is no longer available.	Possible confusion and substitution of shot 1 and shot 2 buttons may be significant.	
A3	Retracting flaps below MinMan speed may stall aircraft.	Decision about when to retract flaps is both necessary and critical.	
A4	Additional task required to switch between different warnings and check off actions reducing time available.		
I1	1. Work tasks provide good feedback (tactile, auditory, visual). 2. Interaction tasks provide less direct feedback (e.g. When a plan has been completed).		
I2			
I3	In general no, but there are some requirements to monitor intervals of time between actions (second shot 30 seconds after the first one).		
I4	Information relevant to the interaction tasks (as opposed to work tasks) can only be		

	determined if user has checked off items etc.		
I5			
I6			
I7			