THEA – A Reference Guide

Steven Pocock, Bob Fields, Michael Harrison, Peter Wright

Human-Computer Interaction Group

Department of Computer Science

University of York

York YO10 5DD

stevep, mdh, pcw @cs.york.ac.uk, b.fields@mdx.ac.uk

Abstract

THEA is a technique designed to help anticipate human-computer interaction failures. It has been developed within the Dependable Computing Systems Centre (DCSC) at York for use by systems engineers. Human factors knowledge is not necessary for the application of the method. It is intended for use early in the development lifecycle as design concepts and requirements concerned with safety, usability and functionality are emerging. The technique employs an embedded cognitive error analysis based on Norman's model of human information processing and does not presuppose any formal knowledge of human factors or cognitive psychology on its users.

This report is based upon, and updates, a previous University of York Technical report (YCS294) (Fields et al., 1997).

Table of Contents

Tables

Figures

Screenshots

Executive Summary

Operators working within technologically sophisticated safety-critical domains such as nuclear power production, aviation and medicine, interface with systems possessing intricate defences to reduce the likelihood of accidents. Notwithstanding this, accidents and incidents continue to occur. To identify ways in which interfaces may be vulnerable to erroneous operator action, a descriptive error analysis, as distinct from more ‘traditional’ quantitative human reliability analysis (HRA) approaches, can be valuable in clarifying how the design needs to change. THEA is one such descriptive technique supporting iterative analysis and design of dependable interactive systems.

An important reason for the development of THEA is that the technique may be carried out by system engineers who are likely to possess only a limited grounding in human factors. The method is designed to inform human-computer interface design at an early stage of development, perhaps as part of concept evaluation and review, or assessing a design prototype.

It is worth mentioning that it is not the aim of this report to support the process of making **quantitative** estimates of the likelihood of interaction failure. Rather, the aim is to help designers reason about errors early in the design lifecycle of the interactive system, and to take account of such reasoning while the design is still flexible enough to be modified without excessive time and expense. One concern is to establish how work is *actually* practiced rather than the way it is *envisaged* as being carried out.

Users and intended audience

The primary users of this document, and of the technique it describes, are intended to be systems engineers involved from the early stages in the design lifecycle of products possessing substantial interactive components. No particular grounding in human factors, cognitive engineering or psychology is assumed, although occasional assistance from human factors personnel may be appropriate for specific difficulties that may be encountered. Of prime importance is an understanding of the domain and context in which a new system is to be fielded. Indeed, the technique can be seen as affording designers and engineers a highly structured means of applying their domain expertise to assessing user interface designs from a human factors point of view.

Structure of the report

The first part of this document describes the THEA technique including its underlying composition around a model of human information processing. The physical and environmental setting of the proposed system is detailed through use of scenarios, as well as the tasks that humans will be required to carry out in such scenarios. This information forms the principal input to a questionnaire-based error analysis (EA), and affords designers the ability to anticipate areas of potential interaction failure of the system being assessed.

The second part of the report describes a case study from the aviation domain and also provides a fully worked example of the THEA technique applied to a less complex domain involving the programming of a domestic video recorder. It is intended that this will demonstrate clearly how the technique may be applied to individual, more complex, projects.

Finally, we briefly describe “ProtoTHEA”, a prototype software tool to assist with THEA analyses.

1 The human error assessment process

1.1 Introduction

Testing a design for usability is playing an increasingly important role in system development as human-computer interaction becomes more sophisticated. An approach, known as 'empirical usability testing', aims to improve the usability of a product through observing real users performing real tasks. The data is analysed and problems diagnosed so that changes may be incorporated to fix them. While comprehensive, such an approach is costly in terms of time and expense. It takes place late in the design process, and requires at least a working prototype. In response to such concerns, methods taking account of human mental processes, based on underlying models of human cognition, have been devised that can be carried out with early versions of the design without representative users. THEA is one such method, aimed at establishing requirements on a design to afford a more error resilient system design. It has its roots in human reliability assessment (HRA) methods (Kirwan, 1994), but unlike these, it is specifically designed to inform human-computer interface design at an early stage of development. The need for prior human factors experience or familiarity with cognitive psychology is not required since the method possesses a cognitive model (discussed in Section 6) embedded within the error analysis questionnaire (Section 7). It is a suggestive technique, guiding the analyst in a structured manner to consider areas of a design for potential interaction difficulties.

1.2 The THEA process

The main components of the THEA process are illustrated in Figure 1, and include:

- Understanding the device being designed (Box 1)
  - *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 the work for which a system will be used (Box 2)
  - *Scenario description*: Taking representative examples of the use of the system 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.

- Goal decomposition (Box 3)
  - *Task analysis:* To structure and interpret information contained in scenarios, hierarchical task analysis (HTA) is a practical, but by no means the only, way of achieving goal decomposition. We describe an operator's tasks in terms of the goals and sub-goals that the person is trying to achieve and the actions used to achieve them.

- Understanding how errors can arise (Boxes 4 & 5)
  - *Error analysis (EA)*: The identification and explanation of human erroneous action that may arise in the operation of the system, possibly as a result of the way it is designed. The EA poses questions about the scenario to reveal areas of design where cognitive failures may occur, and to assess their possible impact on the task or system being controlled.
  - *Model of human cognition* – 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 examine, and how participants will act in a given scenario. In this document we make use of a particular model known as the "execution-evaluation" model of human information processing (Norman, 1988). This can be used to help understand some of the causal factors which can lead to error.

- Designing for error (Box 6)
  - *Impact analysis and design iteration*: assessment of the likelihood of the human erroneous action and the implications for design.

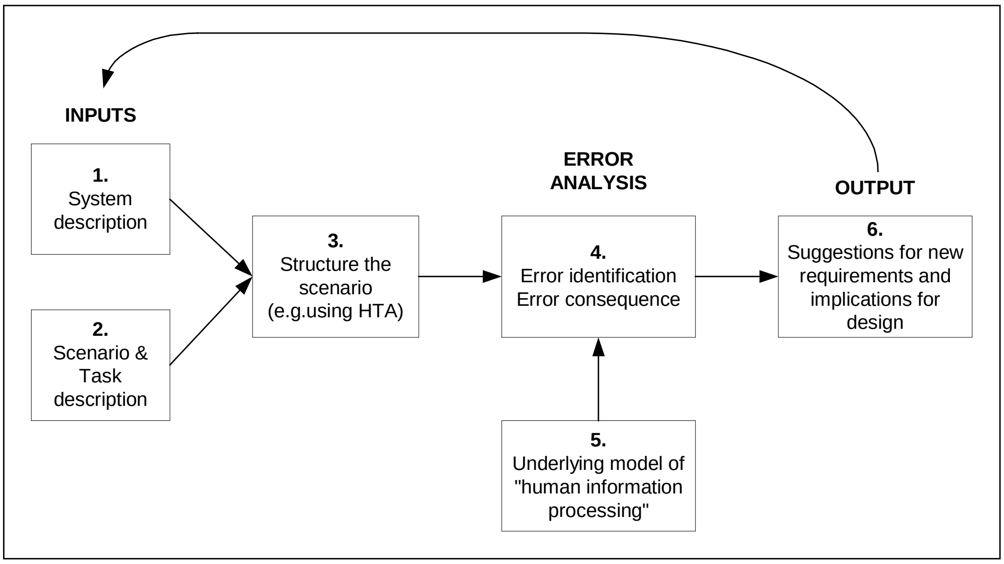


Figure 1 - The THEA process

THEA takes the view that it is context, or the circumstances in which actions are performed, which is a primary determinant of human performance. Scenario actions are surrounded by contextual factors that allow those actions to happen and provide opportunities for human error. THEA analyses capture through detailed scenarios the conditions that may result in unanticipated and unintended interactions. Knowing how a device functions in a scenario provides a systematic and structured means of critiquing a design and developing further requirements.

The backwards pointing arrow in Figure 1 illustrates that the THEA process is intended to be applied iteratively to refine the design specification or work description. New designs are infrequently created from scratch but are most likely to be modifications or re-designs of some existing product. In such situations, understanding the differences between the old and new versions is usually highly informative. The next section discusses briefly why this is so.

1.3 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 include:

- 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 as, for example, aircraft accident investigations;
- Accounts provided by 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.

2 Scenarios

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. To achieve this, it is recommended that scenarios are identified and collected that represent the use of a system *in context* (Kyng, 1995). Scenarios are often used in industry as a means of assessing the consequences and possibilities of a design. For example, in the military, ‘forcing missions’ are chosen, based on criteria concerned with mission effectiveness of a system. Judgements are then made concerning the difficulty of achieving mission goals. 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 envisage and articulate how they would behave in a given situation more effectively. This allows designers to envisage how their design may be used.

In the THEA approach, we are more concerned with choosing usage scenarios which highlight how a design creates opportunities for error, thereby impacting dependability.

2.1 How to structure a scenario

The purpose of using scenarios in design is to provide designers and analysts with a way of capturing how a proposed design will be used. 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 afford opportunities for 'error'. The aspects of context that should be recorded in a scenario description are as follows:

- The physical environment and situation in which participants find themselves;
- The task context;
- The system context.

A template for describing scenarios is shown in Table 1. Generic completion information is shown in each cell, but naturally in a blank template, only the row headings will be displayed, with space beneath for recording analyst information.

Table 1 - A template for describing scenarios

AGENTS - The human agents involved and their organisation - The roles played by the humans, together with their goals and responsibilities
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 exist, and will they be followed as prescribed?
SYSTEM CONTEXT - What devices and technology are involved? - What usability problems might participants 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?
ASSUMPTIONS - What, if any, assumptions have been made that will affect this scenario?

2.2 Where do scenarios come from?

To identify situations that may be significant, we make use of the following information sources:

- The stories and experiences of practitioners (pilots, operators, other crew members i.e. the users) and other domain experts (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 problems experienced, incidents, likely events;

- Frequent conditions and normal operation. This could be based on expert judgement or usage logs of an existing system;
- Changes in technology, organisation, function allocation etc. from a previous or existing system. Here, scenarios will focus on changes in the system, such as changing from a three- to two-man crew on an aircraft flight deck. In this example, an appropriate scenario might be where the tasks of the now obsolete third crew member (e.g. the flight engineer) are particularly tested and place new requirements on the remaining crew;
- 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 practitioners. For example, a move from voice-based communications in air traffic control (ATC) to digital data-linking may evoke scenarios which focus on complex and difficult air traffic conditions *whatever* control regime and supporting technology is in-situ.

2.3 Have sufficient scenarios been collected?

The question often arises as to how many scenarios are required to capture the usage context in sufficient detail. Has a “good enough” coverage been obtained, so that when the system is fielded, the most important requirements can be confidently ascertained? The answer really relies on expert domain judgement. If domain experts are not the people performing the analysis, it will be desirable to have at least one domain expert involved in the scenario construction process.

2.4 Example scenario

We present in this report a case study regarding an emerging design, and concentrate on one snapshot of this design and hypothesise about how it will be used. Historical records of system operation could not be relied on here, so this scenario was used as a way of eliciting from experts how they think the scenario might unfold, and where they think problems might occur.

The scenario takes place on the flight deck of a reconnaissance aircraft which is monitoring fishing vessel activities, and flown by a two-person crew. The flight deck previously had a third crew member, the flight engineer, but who has now been replaced by automation. This scenario is important as it involves activities in which, in the old system, the flight engineer was heavily involved. Table 2 provides an overview of the scenario.

Table 2 - Overview of flight deck bird-strike scenario

AGENTS The proposed design will be flown by two flight deck crew (in contrast to the three currently present). 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, under 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 three-person flight deck.
SITUATION AND ENVIRONMENT The starting conditions for the scenario are an aircraft flying at low level over water (200ft, during daytime) photographing a fishing vessel. To conserve fuel, the aircraft is flying on three of the four engines (2,3,4). The aircraft suffers a massive bird strike on the right side which has two engines running (3,4). As a result of bird ingestion, 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. This will result 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, before commencing the drills in response to the engine fire/failure and any secondary warnings that occur. The immediate priority response sequence to keep the aircraft airborne is power, drag, trim, and engine re-start. The pilot will attempt to gain altitude, although a single engine may not be sufficient to climb or

maintain the current altitude – hence the importance of restarting the number 1 (left-most) engine. After completing these actions, the crew must perform 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 electronic centralised aircraft monitoring (ECAM) display. Additionally, these are written on flight reference cards and also, presumably, in the pilots' memory.
ACTION The pilots' actions are overt, physical acts (mostly inputs or communications) carried out by one or other pilot. See Section 5 for a detailed description and explanations.
EXCEPTIONAL CIRCUMSTANCES There are a number of possible variations in this scenario, including: ▪ Failure of hydraulics pumps (additional crew tasks arising from secondary failures); ▪ Additional navigation tasks (if bird strike close to land, the additional burden on the crew to navigate safely away from the area becomes more critical and complex); ▪ Unsuccessful fire drill (extinguishers may not be adequate to put fire out; engine 1 may not restart; aircraft may be heavy and unable to climb. Question of ditching aircraft becomes a consideration requiring a completely different set of tasks be carried out).
ASSUMPTIONS No specific assumptions have been made. For simplicity, the 'exceptional circumstances' are not considered.

3 Understanding TASK Context

We mentioned previously that tasks and task knowledge play a pivotal role in the ongoing activity. In this section we discuss in greater detail how tasks may be described. HCI literature describes many methods of analysing tasks, each with associated strengths and weaknesses. The error analysis process does not mandate the use of any specific task analysis method, nor is any specific notation required for describing tasks. If an analyst or engineer applying THEA is familiar with a particular technique, or a task analysis has already been carried out as part of the project, then it makes sense to re-use as much work and expertise as possible. Nevertheless, a number of features of any task description are desirable:

- Work is described in terms of the *agents* and *roles* that are responsible for carrying it out;
- Associated with each role are 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 in the environment or the result of some internal cognitive process.

A technique which possesses such characteristics is Hierarchical Task Analysis (HTA) (Kirwan, 1994), and is described in the next section. If, however, the interaction under examination is relatively straightforward, it may only be necessary to write down the goals each operator will be engaged in, together with the actions required to achieve each goal. In such situations, the unnecessary hierarchical complexity of employing an HTA can be avoided.

3.1 Hierarchical goal decomposition

Hierarchical Task Analysis (HTA) is a technique that can be used to describe an operator's tasks in terms of the goals and sub-goals that the person is trying to achieve and the actions used to achieve them. It is *hierarchical* 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. A simplified example in Figure 2 is based on the bird strike scenario described in Table 2:

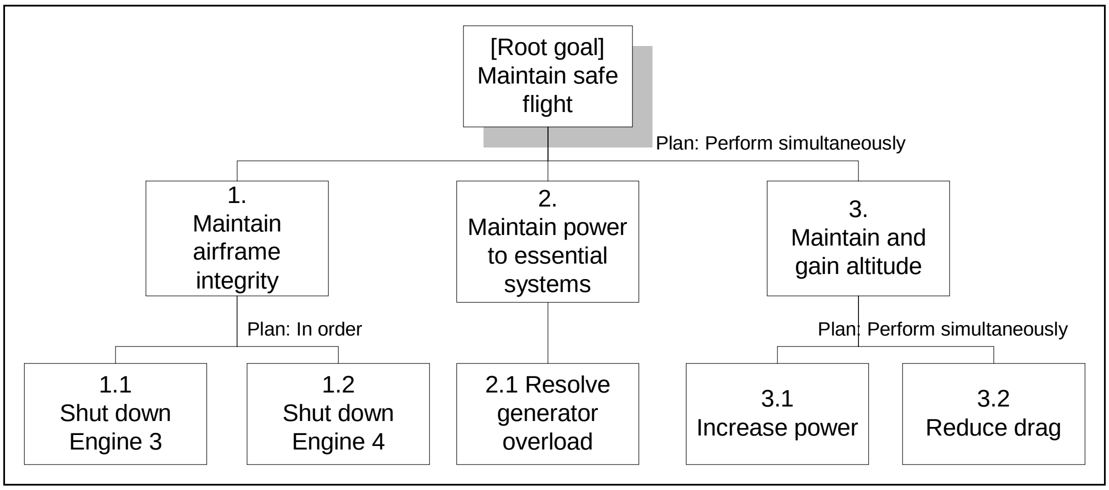


Figure 2 - Example of a simple hierarchical task analysis (HTA)

If necessary, the lowest level sub-goals may themselves be decomposed into smaller sub-goals, depending on the level of detail required.

3.2 When to stop hierarchical decomposition

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 on the purpose of the HTA. If the purpose is to consider training needs, then analysis might well stop at a higher level than if the purpose is to consider what displays and controls an operator might need. Ultimately, a complete analysis may well need to decompose the task to the level of individual operator actions.

However, we argue that the analysis process is an iterative one and that it can, and should, commence with fairly high-level goals associated with the task. The particulars of a task will determine whether, once this high-level analysis is complete, there is a need to pursue all nodes in the hierarchy down to individual actions. With experience, it is usually apparent when the necessary level of decomposition has been achieved. An incidental benefit encountered in practice has been the occasional need to insert previously unconsidered nodes or sub-nodes (and sometimes node removal) as a result of completing, or working through, the error analysis questionnaire.

3.3 Plans

A goal decomposition describes *how* a problem can be broken down into simpler sub-problems. What they do not describe, however, is *when* the sub-problems must be addressed and in what order. Clearly it is possible to carry out some sub-problems in one order only (for example, a pilot receiving clearance from air traffic control cannot be confirmed until it has been received), but in some instances, 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. In Figure 3, this is the text between the hierarchical levels (*'Perform simultaneously', 'In order', etc.)*. The description makes the ordering sequence explicit and it is also possible to specify conditional goals such as 'if…then'. In this way it is possible to include statements about what to do if a particular goal is *not* achieved (such as if air traffic clearance is refused). 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. Plans can also be used to specify the triggering conditions under which certain optional sub-goals can be initiated, which may be failure conditions of either the system or the operator. If plan and goal descriptions have been carried out correctly, every goal mentioned in the goal description should also be mentioned in the plan, and vice versa. Additionally, any restrictions on the order in which the goals can be achieved should be mentioned in the plan. These two features are useful for validating the task analysis structure.

A useful notation for describing plans is shown in Table 3.

Table 3 - A notation for describing plans

Conditional:	**If** <condition> **then** <plan>
Triggering:	<condition> **triggers** <plan>
Sequence:	<plan> ; <plan>
Parallel	<plan> ‖ <plan>
Any order	<plan>, <plan> **in any order**
Repetition:	**Repeat** <plan> **until** <condition>

4 Understanding SYSTEM context

When we refer to some aspect of the technology under scrutiny about which we may have particular concerns, we are actually referring to the context of use of that technology. To assess such context, it is highly desirable to ascertain:

- what devices and technology are involved,
- what usability problems might they present to users, and
- what effects can users have on the technology.

A particular area of concern regarding usability often cited as a particular problem and a causal factor in many accidents and incidents, is that of moding or, more specifically, "mode error". THEA addresses moding issues as part of the error analysis (see the error analysis template in Appendix B – Blank error analysis template, questions A3+I8 explicitly, and also A2/I3), specifically to minimise potential for mode errors. The next section explores moding in greater detail.

Appendix D – Non-scenario-based briefly discusses some alternative issues that should be considered during interface design. Combined with an error model and a particular usage scenario, these can also assist with revealing potential errors in a new design. The aim is not to provide a detailed presentation of formalised techniques for analysing interfaces, but rather to help raise important design questions. These may be answered from experience and intuition, especially when the interface is not especially complex. Many other techniques exist and are discussed extensively in the literature (see, for example (Dix et al., 1998)).

4.1 What is a mode?

When using the term 'mode', we refer to a system configuration that defines how it interprets user input, or how its output should be construed by the user[1]. 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 either by user actions, or autonomously by the system itself.

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 exists 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. However, even a single device can exist in several modes concurrently. For example, a process control system can have a "training mode" and an "operational mode". Additionally, it may have "safe" and "emergency" modes. The whole system can be thought of as a composite mode e.g. "training" + "safe" mode. The potential therefore exists for what is known as ***mode ambiguity*** where certain user actions can have different effects on the system depending on its current state. This

[1] '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.

can be compounded by inappropriate mode feedback and ultimately a state of ***mode confusion*** can exist within the user. Once an operator is confused about the mode a system is in, this can lead to the commitment of ***mode error*** from the *system's* point of view. From a *user's* point of view, this leads to situations where the automated systems act in some way 'outside the expectations of their human supervisors' (Woods et al., 1994). This discrepancy between a user's expectation and the systems acting might be due to a lack of ***mode awareness***. Figure 3, based on (Loer et al., 1999) illustrates one representation of these different aspects of moding together with their interrelationships. This is followed by a brief explanation of the terms presented.

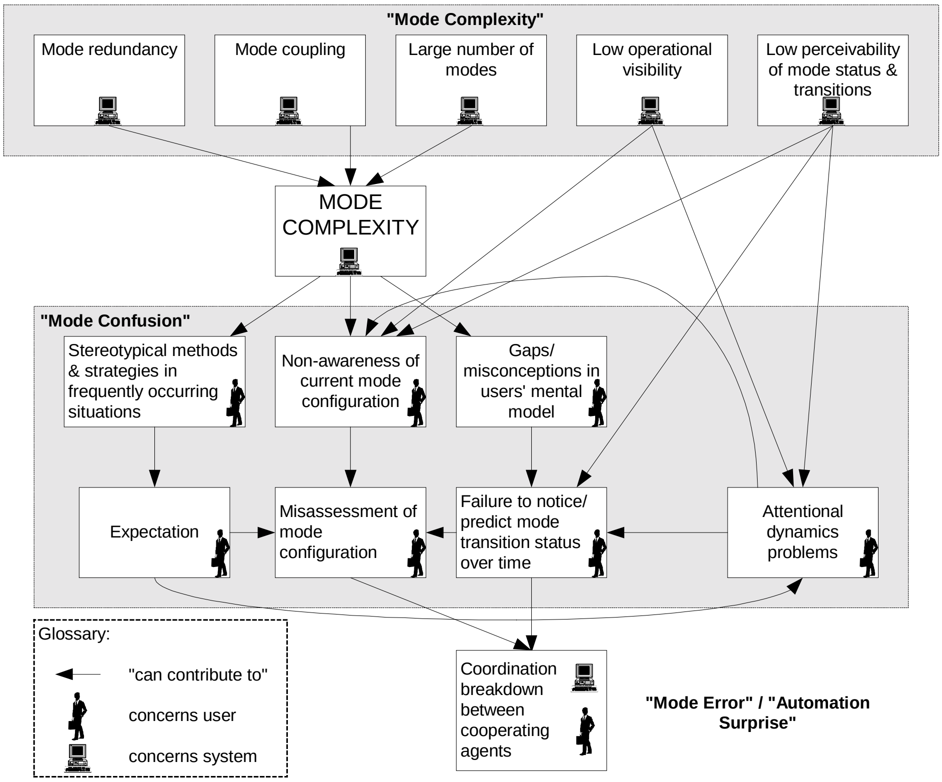


Figure 3 - Moding issues from a system and human perspective

- Coordination breakdown between co-operative agents:
  A coordination breakdown between these agents is regarded as equivalent to the system-centred view of mode error and the human-centred view of automation surprises.

- Expectation
  Includes both the user's expectation based on the history, experience and perceived system output (visual, tactile, and so on), as well as the system's 'expectation' based on the interpretation of sensory inputs and program code.

- Attentional dynamics
  Attentional drifting and being side-tracked (for example, being required to deal with a higher priority task during execution of regular duties e.g. checklists). Can lead to missed mode transitions or a failure to appreciate the significance of a transition.

- Stereotypical methods
  Use of the same methods based on experience of dealing with frequently recurring situations. In non-standard conditions there is an increased likelihood of performing the often-performed action(s), which in this situation constitutes erroneous action as a result of the standardised mental model.

- Mode complexity

A mode structure that is difficult to comprehend.

- Mode redundancy
  Different combinations of modes can have the same effect.

- Mode coupling
  Interdependencies of modes exist within the same, or between different, systems or sub-systems.

4.2 Assessing moding problem causality

Figure 3 helps determine possible causes of potential moding problems that may be identified in a THEA error analysis. Such causes have important implications for the construction of system requirements. An example may serve to clarify this.

Consider one of the questionnaire's moding questions 'A3': "*Is the correct action dependent on the current mode?*" In other words, is the technology being used required to be in a particular mode before a user is able to perform the action correctly? If so, there is a chance that the mode currently selected may be the inappropriate one for the task in hand. An unintended system response, possibly with an undesired consequence, may occur as a result. For example, when trying to set the time on a domestic clock radio, one may inadvertently de-tune the previously set frequency. The same 'up/down' buttons are used to set both the time and frequency, but setting the time requires an additional action (pressing a separate button) to be performed while operating the 'up/down' keys. In other words, the unit must be in the clock 'mode'.

Figure 3 can help determine possible causes for incorrect operation, and might lead us to examine areas of the design where re-appraisal may be needed. In our simple example, a user failing to press the clock mode button is either unaware of the unit's current mode configuration, or unaware that an additional separate action needs to be performed in order to change the time digits on the display. How might this arise?

The two relevant boxes on the diagram are "Non-awareness of current mode configuration" and "Gaps/misconceptions in users' mental model" (larger of the two shaded regions, top row). From the arrows leading in to each box, we can see that possible causes for the user's incorrect action may result from 'mode complexity' (see arrows leading into *this* box for possible causes), low visibility of the button, or perhaps because the present mode is enunciated poorly or not at all. The latter two seem the most probable. Another possibility, 'Attentional dynamics problems' (and therefore its causal factors), can be discounted here as this is very unlikely. By following plausible pathways in the diagram, potential causes can be ascertained, permitting further questions to be asked of the design. In such a way, solutions may be found, and the design modified appropriately.

5 Understanding actions in context

We have seen that one of the principal components of a scenario is a description of the actions which take place. The scenario discussed in Section 2.4 involves actions performed by each agent (the two pilots and the "system") and is illustrated in Table 4, listing the actions and events taking place in the early part of the scenario, with time increasing in a downwards direction:

Table 4 - A timeline showing individual crew and system actions

System status	Pilot	Co-Pilot	Info sources	System response
Engine 3 fire warning	Throttle 2 max.		Airmanship	Select 'ENG ECAM' page
	Press master warning		Airmanship	
	Throttle 1 idle	Close ext. doors		
Engine 4 failure warning		Flaps 0		
	Throttle 1 max.	Rudder trim		
		Warn crew		Start engine
	Navigate safe exit route			
		Throttle 3 close		
		Engine 3 LP cock shut	Engine 3 fire drill	
↓ *Time*		Engine 3 fire ext: shot 1		

Table 4 illustrates 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. What this tabular presentation also begins to highlight is the fact that the two pilots are performing possibly contradictory tasks at the same time. For example, the pilot is attempting to restart engine 1 to obtain more thrust, while the co-pilot is performing actions to close down the damaged engines (3 and 4) which will *reduce* overall thrust.

However, what the table does not show are links between actions (opening and closing throttles) and the surrounding context i.e. the goals to which the actions are directed, which was one of the reasons for constructing scenarios in the first place. To address this deficiency, we might modify Table 4 to take account of the task ordering and the goals to which they are directed, as derived from the task analysis. The result is shown in Figure 4.

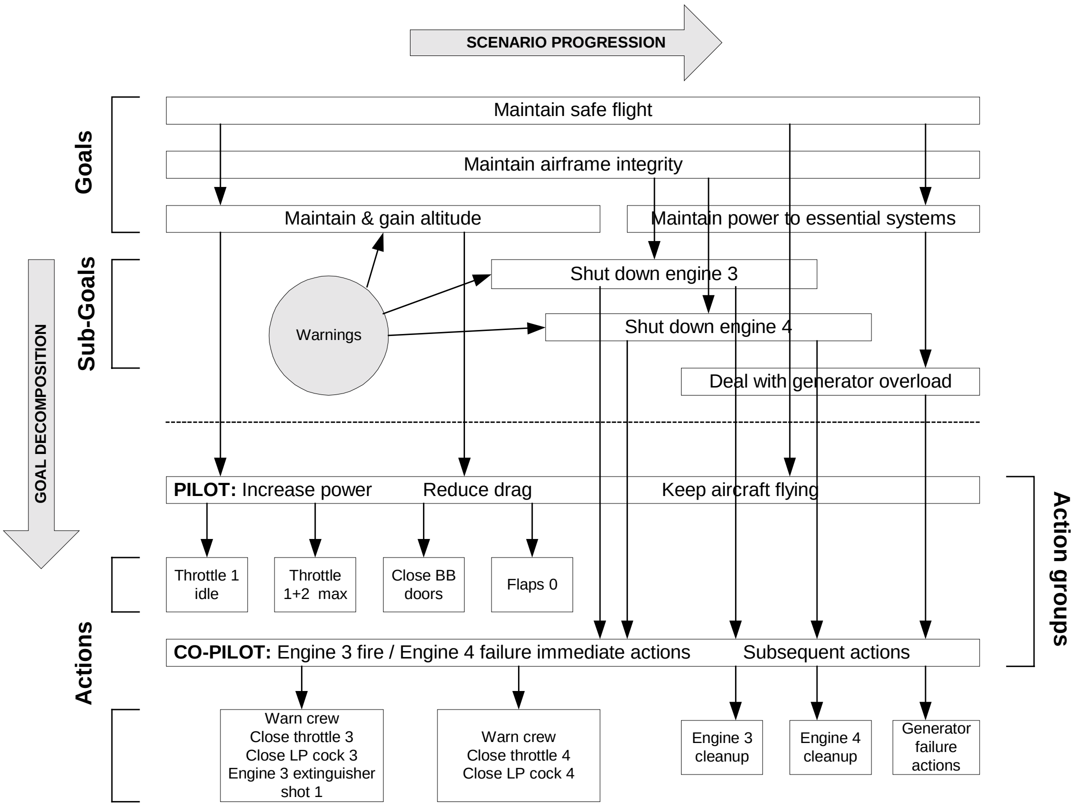


Figure 4 - Alternative hierarchical goal structuring diagram of bird-strike scenario

Presenting actions in this manner highlights a number of scenario features not immediately apparent in previous representations. In particular, it shows which goals and tasks become active, and are active concurrently in the scenario (not present in a simple task analysis such as Figure 2), and which actions are related by being directed towards the same goals (not present in a simple event listing such as Table 4 which makes no mention of goals).

6 Understanding erroneous operator actions

The error identification process discussed in the latter part of this report is based on two views of how human behaviour may be described, and forms part of the error model. On the one hand, we can assume 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. Alternatively, human behaviour may 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 the 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 conjunction.

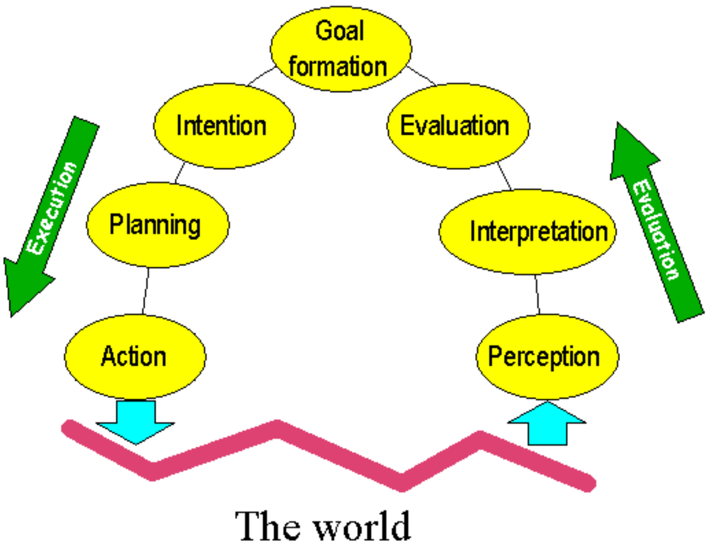


Figure 5 - Norman's "Execution-Evaluation" model of human information processing

6.1 Cognitive failure

Errors can be regarded as failures in cognitive processing. Figure 5 shows a model of human information processing (Norman, 1988). This illustrates that human action has two aspects – execution and evaluation. The stages of execution start at the top with the *goal* i.e. the state that is to be achieved[2]. The goal is translated into an intention to do some action which must be translated into a set of internal commands, an *action sequence,* that can be performed to satisfy the intention. However, nothing happens until it is executed, and performed on 'the world'. Evaluation commences with our perception of 'the world', which must be interpreted according to our expectations and then evaluated with respect to both our intentions and our goals. Using this model, we can identify a number of **cognitive failures**, or ways in which human information processing can fail, which have the potential for leading to erroneous behaviour.

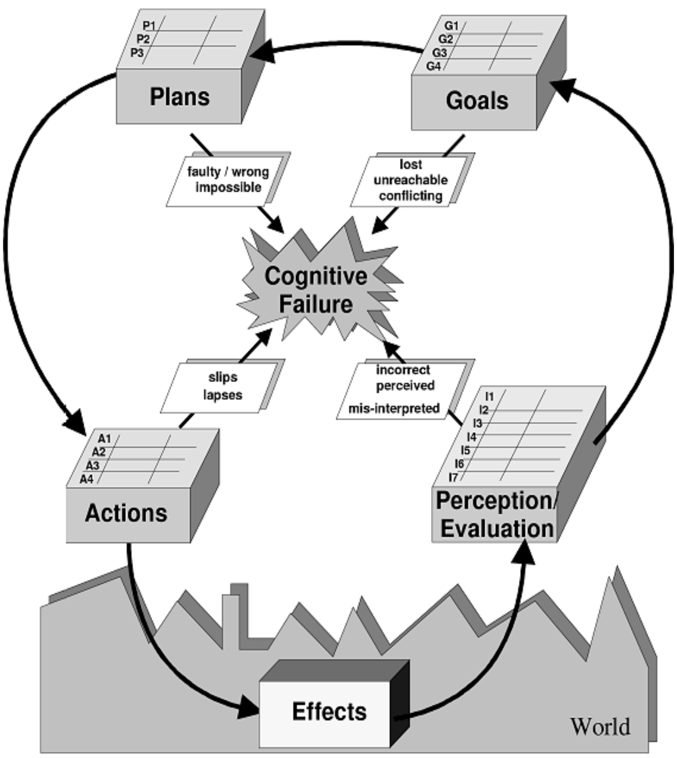


Figure 6 - Sources of cognitive failure based on Norman's model and the THEA error analysis questionnaire

Failures in human information processing include:

- **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 conflicting goals);
- **Faulty plans** (plans 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 or perceived data, failure to realise when a goal has been completed).

Figure 6 illustrates how such failures can be linked into the model of human information processing, as well as their relationship to the THEA error analysis questions discussed in Section 7.2. Table 5 provides some examples of cognitive failure.

[2] Note that the cycle can also begin with a perception of some condition in 'the world'.

Table 5 - Some examples of cognitive failure

Cyclic model stage	Cognitive failure type	Example
Goals	Lost goal Unachievable goal Conflicting goals	In the case study, forgetting to return to engine 3 fire 'cleanup' actions. Fail to notice and act on a warning (trigger). Attempting to make an impossible aircraft course change. Conflict between goals to maintain thrust and shut down engine.
Plans	Faulty or wrong plan Impossible plan	Shutting down the wrong engine. 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/mis-perception Misinterpretation	Mis-read the current setting in the altitude alert window. Read a display value and erroneously interpret the figure as angle of descent instead of vertical speed.

In Section 8, we ask questions about the performance of each of the cognitive components in relation to the use of the system to try and anticipate where cognitive failures might occur and result in behavioural errors.

6.2 Deviations from expected behaviour

Thus far, we have described a cognitive view of error, employing causal guidewords such as 'lost', 'faulty', 'wrong', 'impossible' and so on. Perhaps more familiar is the behavioural approach, where errors are described in terms of deviations from some prescribed, or normal, course of action. A set of "keywords" captures classes of behavioural deviation and are derived in the main from the nuclear power industry, and employed in techniques such as HAZOP (Kletz, 1983):

- Omission – fail to carry out an action or the actions associated with a sub-goal
- Commission:
  - Incorrect – carry out the correct action or sub-goal, but do so incorrectly
  - Substitution – substitute an incorrect action or item of data for a correct one
  - Insertion – insert an extraneous action into the stream of behaviour
- Sequence – perform the right action or sub-goals unnecessarily
- Quantitative – carry out a correct action, but with some 'quantitative' error (too much / too little / too long / too short etc.).

6.3 Some problems

It is highly probable that such an enumerated set of terms is incomplete. Moreover, although the definitions may appear unambiguous, this is not always the case. Figure 7, adapted from (Hollnagel, 1998) illustrates, for example, the differing ways in which an action, required at a specific time, may be omitted:

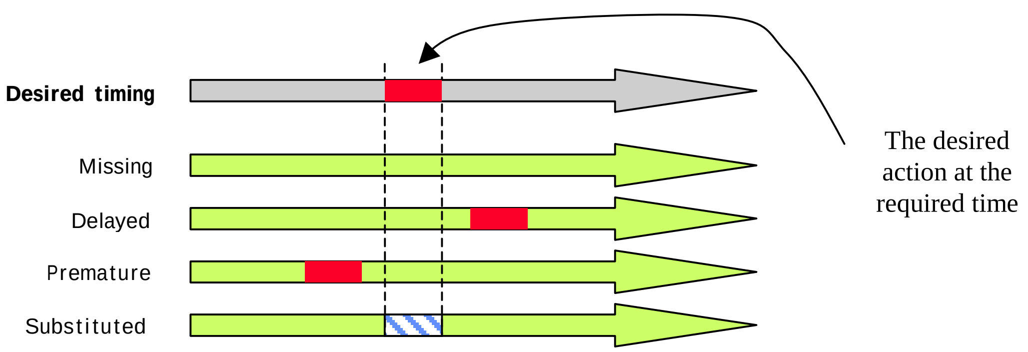


Figure 7 - Four types of "omission" error

This illustrates that an error of **commission** (substituted action, bottom arrow), i.e. performing an unplanned or unintended action, effectively *precludes* the required event taking place at the required time. Thus an error of commission logically implies an **omission** also, since one cannot make a commission error without also making an omission. Moreover, errors of commission are insufficiently constraining as a guide due to the large number of substitutions possible. For example, while entering data into a flight computer, a pilot could:

- Do something other than enter data;
- Enter the data into a different device;
- Enter, for example, a distance value instead of an altitude.

With these caveats in mind, we can now discuss the THEA error analysis which is framed around a set of questions based on the human information processing model, and employing some of the terms just described.

7 Error Analysis

We now draw together the previous strands to provide an analysis technique that will help us assess a system's vulnerability to erroneous interaction. The approach poses questions in a structured and systematic way, based on the failures suggested by the model of human information processing. In this way, it is easier to envisage ways in which things might go wrong, leading to a failure in cognitive processing. Once a potential problem is identified, it is possible to think about how that failure may be manifested in erroneous behaviour, as well as anticipating the ultimate effects on the state of the entire system.

7.1 The error identification process

Error identification in THEA involves the user asking pertinent questions about a scenario in order to reveal areas of interaction that may be problematic. We now detail a set of such questions used by THEA that assists the process. This is a preliminary step in the process of identifying possible causal factors, tracing their effects through to behavioural failures, and ultimately their impact on the task or system being controlled. Figure 8 illustrates such a process:

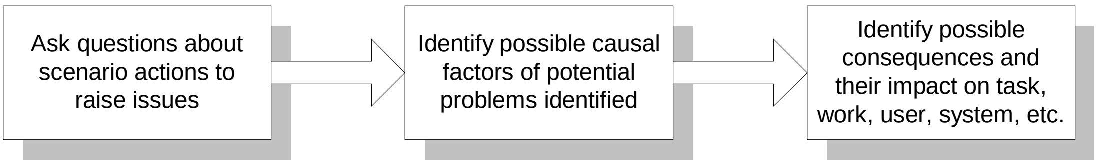


Figure 8 - Process for potential cognitive failure identification

Precisely how the analysis is carried out is largely dependent on the level of detail required. However, two recommended methods are:

1) Follow the goal hierarchical task structure from top to bottom, asking the set of questions about *each* goal or task, or
2) Select parts of the scenario where potential problems are anticipated, ask the question set for appropriate tasks, then conduct a detailed analysis of behavioural error and impact.

The first option clearly affords the most thorough analysis and is recommended for new designs. For complex designs, analysis will naturally be a lengthier process but is likely to uncover a greater number and range of concerns. In some instances, there will clearly be potential for a cognitive failure but with no obvious behavioural manifestations. A good example of this is where goals come into conflict. It is sometimes unclear what the behavioural implications of a conflict will be, although the problem is still potentially serious, especially if the goals involved are important or their resolution is complex. In such cases, the cognitive failure itself may be regarded as the "problem" to which a design solution may be required.

Analysis results can be recorded in a fairly ad hoc way, depending on the specific requirements of the project. However, it has proven useful in practice to record the results in a tabular format such as the one illustrated in Table 6.

Table 6 - A suggested format for recording error analysis results

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 and obtaining an undesirable result.	Consequences of the causal issue. These can take a number of forms: cognitive failures or behavioural errors whose likelihood may be increased; additional cognitive or behavioural work that may 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.

The ‘Question’ column represents a ‘checklist for cognitive analysis’, guiding the analyst in a structured manner through the questionnaire. The ‘Causal Issues’ column is provided as a means of recording an analyst’s thoughts with regard to factors that are likely to influence an operator’s predisposition to commit errors. The ‘Consequences’ column is used to record the possible impact(s) that the identified causal issue might have on the task itself, operator workload, system state, as well as any hazardous conditions that may result. Finally, “Design Issues” provides space for documenting ideas about how the design might be amended to manage or remove the identified problems.

7.2 Applying the cognitive error analysis (EA)

The questions in Table 7 are based on the failures that are possible in the execution-evaluation cycle model of human information processing, as previously discussed in Section 6.1. Note that, for clarity, a ‘causal issues’ column has been omitted since this is only relevant while completing the questionnaire (for example, see the completed questionnaire in Section 9).

Table 7 - THEA Error Analysis questionnaire for asking structured questions about a scenario

Questions	Consequences	Design Issues (where appropriate)
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. 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.	

P2. Are there well practiced 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 do exist and are familiar, then they might be followed inappropriately, not taking account of the peculiarities of the current context.	
P3. Are there plans or actions that are similar to one another? Are some used more often than others?	A similar plan may be confused for the intended one, resulting in the substitution of an entire task or sub-task.	
P4. Is there feedback to allow the user to determine that the task is proceeding successfully towards the goal, and according to plan (if there is one)?	Insufficient feedback may result in the user becoming confused about how the plan is progressing. Additional undesirable or inappropriate actions may be performed as a result of this lack of clear feedback.	
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 to the system resulting from user action clearly perceivable?	If there's no feedback that an action has been taken, the user may **repeat** actions, with potentially undesirable effects.	
I2. Are the effects of user actions perceivable immediately?	If feedback is delayed, the user may become confused about the system state, potentially leading to a **supplemental** (perhaps inappropriate) action being taken.	
I3. Are changes to the system arising from autonomous system action(s) clearly perceivable by the user?	A non-awareness of an autonomous system change, or a misconception in the user's mental model may result. This could lead to a misassessment of, for example, the current mode configuration, potentially resulting in inappropriate user actions.	

I4. Are the effects of any autonomous system action(s) perceivable immediately?	If feedback is delayed, the user may become confused about the system state, potentially leading to a **supplemental** (perhaps inappropriate) action being taken.	
I5. 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).	
I6. Can the user determine relevant information about the state of the system (from the total information provided?)	If not, the user will have to remember the information they require, thus making it prone to being lost or recalled **incorrectly**.	
I7. 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.	
I8. If interacting with a moded system, 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.	

8 Error analysis example

This section illustrates how an analysis might typically be carried out and is based on the case study detailed in Section 2.4. For clarity, the example in Table 8 does not examine particular (low-level) operator/system tasks, but instead demonstrates the general technique of questionnaire completion. By following the example, the technique of performing detailed domain-specific THEA analyses for individual projects should become apparent. The “Design Issues” column has been left intentionally blank since it is not really appropriate in this high-level ‘overview’ example. For a less complex, but more detailed lower-level worked example, see Section 9.

Table 8 - Error analysis example for the bird-strike scenario

Question	Causal issues	Consequences	Design issues
GOALS, TRIGGERING, INITIATION			
G1	1. Many goals are triggered fairly directly via the environment as well as the interface (e.g. presence of engine smoke and interface fire warning). 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	The presence of data within a dedicated warning display ‘evokes’ or ‘suggests’ that action is required to be taken	It is assumed that display will have sufficient ‘attention grabbers’ to alert crew.	
G3	Goals to Increase power and Engine 3 shutdown	Resolving the conflict	

	are in conflict (although it is inevitable in this scenario).	satisfactorily requires negotiation between PF and PNF. The time required for this negotiation may lead to a non-optimal (too late) decision.	
G4	Unknown (see EAs for lower-level tasks)	-	
PLANS			
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 engine 3 tasks later.	1. At the level of actions, plan following is well supported, but at the level of goals (e.g. Engine 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. Engine fire procedure is better practiced.	Actions from engine fire procedure may be done on engine 4. But this is a superset of engine failure actions.	
P4	Unknown (see EAs for lower-level tasks)	-	
PERFORMING 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 minimum manoeuvring 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.	May omit.	
PERCEPTION, INTERPRETATION & EVALUATION			
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	In this scenario most action effects are perceivable immediately.	-	
I3	Unknown (see EAs for lower-level tasks)	-	
I4	Unknown (see EAs for lower-level tasks)	-	
I5	In general no, but there are some requirements to monitor intervals of time between actions (second shot 30 seconds after the first one).	Distraction of other (high priority) tasks during waiting period may result in second shot being delayed/omitted.	
I6	Information relevant to the interaction tasks (as opposed to work tasks) can only be determined if user has checked off items etc.	-	
I7	Possibly, although this should be minimised if standard operating procedures are adhered to	Assume crew adherence to SOPs.	
I8	Unknown (see EAs for lower-level tasks)	-	

9 A fully-worked THEA example

In this second worked example, we conduct a full EA on a single sub-task involved in the ‘root’ goal of programming a domestic video recorder to record a weekly television programme. The scenario for this example is listed in Table 9. It is not the intention to address specific interface difficulties, but rather to provide an example which allows us to focus on the method itself through use of a domain with which most people are familiar.

9.1 Scenario details

Table 9 - Scenario for setting the date when making a domestic VCR weekly recording

SCENARIO NAME: Programming a video recorder to make a weekly recording
ROOT GOAL: Record weekly TV programme
SCENARIO SUB-GOAL: Setting the recording date
ANALYST(S) name(s) & Date: J. Smith / 14th March 2001
AGENTS A single user interfacing with a domestic video cassette recorder (VCR) via a remote-control unit (RCU).
RATIONALE The goal of programming this particular VCR is quite challenging. Successful programming is not certain.
SITUATION & ENVIRONMENT A domestic user wishes to make a recording of a television programme which occurs on a particular channel at the same time each week. The user is not very technologically aware and has not programmed this VCR previously. A reference handbook is not available, but there is no time pressure to set the machine – recording is not due to commence until tomorrow.
TASK CONTEXT The user must perform the correct tasks to set the VCR to record a television programme on three consecutive Monday evenings from 6pm-7pm on Channel 3 (see Figure 9). Today is Sunday.
SYSTEM CONTEXT The user has a remote control unit containing navigation keys used in conjunction with programming the VCR as well as normal VCR playback operation. The RCU has four scrolling buttons, indicating ‘left’, ‘right’, ‘up’, ‘down’. Other buttons relevant to programming are labelled ‘OK’, and ‘I’ (see Screenshot 1).
ACTIONS The user is required to enter a recording date into the VCR via the RCU using the buttons listed above. The actions appear in the order specified by the task decomposition see section 9.2.
EXCEPTIONAL CIRCUMSTANCES There are no exceptional circumstances.
ASSUMPTIONS

9.2 Task representation

A hierarchical task analysis (HTA) constructed for the scenario is shown in Figure 9, highlighting the sub-task ("Enter Date") that we are analysing. Note also the plan associated with the execution of the sub-tasks. Screenshot 1 shows the actual screen the user is presented with at this particular stage.

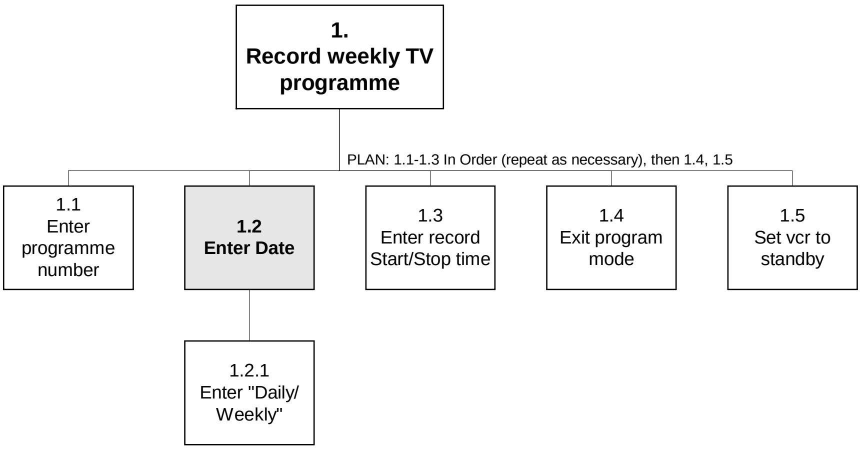
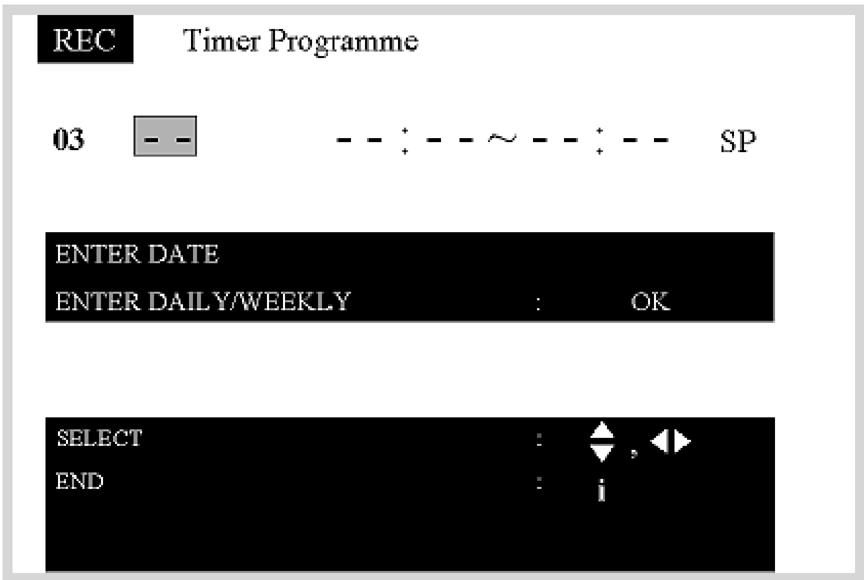


Figure 9 - Video recorder HTA

Screenshot 1 shows the display presented to the user for the "Enter Date" step in the programming sequence:



Screenshot 1 - VCR screenshot awaiting date input

Table 10 shows the completed error analysis questionnaire using the format shown in Table 5 (a template is provided in Appendix B – Blank error analysis template for personal use). It will be appreciated that this represents the analyst's subjective judgement since there are no right or wrong answers. Another analyst may well provide different answers. It has been found useful on occasion for more than one person to complete the questionnaire, especially if coming from different project areas. We have found, for example, that false assumptions or misunderstandings are often picked up by discussion between analysts, thereby increasing the validity of the answers.

9.3 Error Analysis questionnaire

Table 10 - A completed error analysis questionnaire for the "Enter Date" task

SCENARIO NAME: Programming a video recorder to make a weekly recording			
TASK BEING ANALYSED: Setting the recording date			
ANALYST(S) NAME(S) & DATE: J. Smith / 14th March 2001			
Question	**Causal issues**	**Consequences**	**Design issues (where appropriate)**
GOALS, TRIGGERING, INITIATION			
G1 *(Is the task triggered by stimuli in the interface, environment, or the task itself?)*	Yes. (The presence of an "Enter Date" prompt is likely to trigger the user to input the date at this point).	-	-
G2 *(Does the UI "evoke" or "suggest" goals?)*	N/A. (The UI does not, *per se*, strictly evoke or suggest the goal of entering the date).	-	-
G3 *(Do goals come into conflict?)*	There are no discernible goal conflicts.	-	-
G4 *(Can the goal be satisfied without all its sub-goals being achieved?)*	**NO.** The associated sub-goal on this page of setting the DAILY/WEEKLY function may be overlooked. Once the date is entered, pressing the right cursor key on the RCU will enter the next "ENTER HOUR" setting.	Failure to set the DAILY/WEEKLY option. Once the ENTER HOUR screen is entered, the DAILY/WEEKLY option is no longer available.	Suggest addition of an interlock so that the Daily/Weekly option cannot be bypassed.
PLANS			
P1 *(Can actions be selected in-situ, or is pre-planning required?)*	True. (Entering the date can be done 'on-the-fly'. No planning is required).	-	-
P2 *(Are there well practiced and pre-determined plans?)*	N/A. (A pre-determined plan, as such, does not exist, but the user should possess enough knowledge to know what to do at this step).	-	-
P3 *(Are there plans or actions that are similar? Are some used more often than others?)*	There are no similar or more frequently used plans or actions associated with this task.	-	-
P4	Yes. (As the user enters	Task is proceeding	(See A1).

(Is there feedback to allow the user to determine that task is proceeding successfully towards the goal, and according to plan?)	digits into the date field via the RCU, they are echoed back on-screen).	satisfactorily towards the goal of setting the date (although the date being entered is not necessarily correct).	
PERFORMING ACTIONS			
A1 *(Is there physical or mental difficulty in performing the task?)*	**YES.** The absence of any cues for how to enter the correct date format makes this task harder to perform.	The user may try to enter the year or month instead of the day. Additionally, the user may try to add a single figure date, instead of preceding the digit with a zero.	Have a explanatory text box under the field or, better still, default today's date in the date field.
A2 *(Are some actions made unavailable at certain times?)*	No. (The only actions required of the user is to enter two digits in the blank field).	-	-
A3 *(Is the correct action dependent on the current mode?)*	No. (The operator is operating in a single (programming) mode).	-	-
A4 *(Are additional actions required to make the right controls and information available at the right time?)*	**YES.** The date field is presented blank. If the user does not know the date for recording (or today's date), the user must know to press the 'down' cursor key on the RCU to make today's date visible.	The user may be unable to enter the date, or the date must be obtained from an external source. Also, if the user presses either the 'left' or 'right' cursor key, the 'ENTER DATE' screen is exited.	1. Default current date into date field; 2. Prevent user from exiting 'ENTER DATE' screen before an entry is made (e.g. software lock-in).
PERCEPTION, INTERPRETATION & EVALUATION			
I1 *(Are changes to the system resulting from user action clearly perceivable?)*	Yes. (Via on-screen changes to the date field).	-	-
I2 *(Are the effects of such user actions perceivable immediately?)*	Yes. (Digit echoing of RCU key presses is immediate).	-	-
I3 *(Are changes to the system resulting from autonomous system action(s) clearly perceivable?)*	N/A. (The VCR performs no autonomous actions).	-	-
I4 *(Are the effects of such autonomous system action(s) perceivable immediately?)*	N/A. (As I3).	-	-
I5 *(Does the task involve monitoring, vigilance, or spells of continuous attention?)*	No. (There is no monitoring or continuous attention requirements on the user).	-	-
I6	**NO.** User cannot	If user doesn't know	As A1.

(Can the user determine relevant information about the state of the system from the total information provided?)	determine current date without knowing about the 'down' cursor key. Also, if date of recording is known, user may not know about the need to enter two digits.	today's date, and only knows that, say, Wednesday, is when you want the recordings to commence, then user is stuck.	
I7 *(Is complex reasoning, calculation, or decision-making involved?)*	No.	-	-
I8 *(If the user is interfacing with a moded system, is the correct interpretation dependent on the current mode?)*	N/A.	(It is not considered likely that the date field will be confused with another entry field (e.g. 'Hour').	-

9.4 Examining the results and conclusions

When EAs have been performed for each goal/sub-goal in the hierarchical task structure, the questionnaires are examined and each potentially problematic area noted. This can be performed, say, by highlighting each 'negative' (potentially problematic) response on every completed questionnaire sheet (the grey cells in Table 10). Another useful exercise is to colour code each according to severity by, for example, using a green highlighter to mark a potential problem but of low consequence, and medium-/high-consequence ones with yellow and orange, say. When collated, these problems should be examined closely, plus any interrelationships and their possible implications noted and assessed. Interpreting the results should be undertaken with great care and design modifications considered as the preferred option for intractable difficulties. THEA and ProtoTHEA greatly facilitate the process of obtaining the appropriate results for examination.

For the video programming example, it is possible to identify certain problematic areas:

1. The goal of entering the date can be achieved without the sub-goal of setting the 'daily/weekly' function, which could result in only a *single* recording, instead of multiple recordings, being made;
2. The user could encounter problems in entering the date in the correct format, since there are no on-screen guides as to how this should be done, leading to confusion and potential disorientation;
3. The interface makes unwarranted assumptions about the user knowing specific remote control unit functions which are far from intuitive. The "Enter Date" screen can be unwittingly exited, or even 'ejecting' the user from the programming environment completely. At best, this will delay setting the recorder and, at worst, prevent the recorder from being correctly programmed, thereby losing the programme(s);
4. Essential data can only be obtained from the recorder with considerable difficulty and knowledge of remote control input 'code sequences'. Basic information is 'hidden' from the user when it needs to be defaulted on-screen.

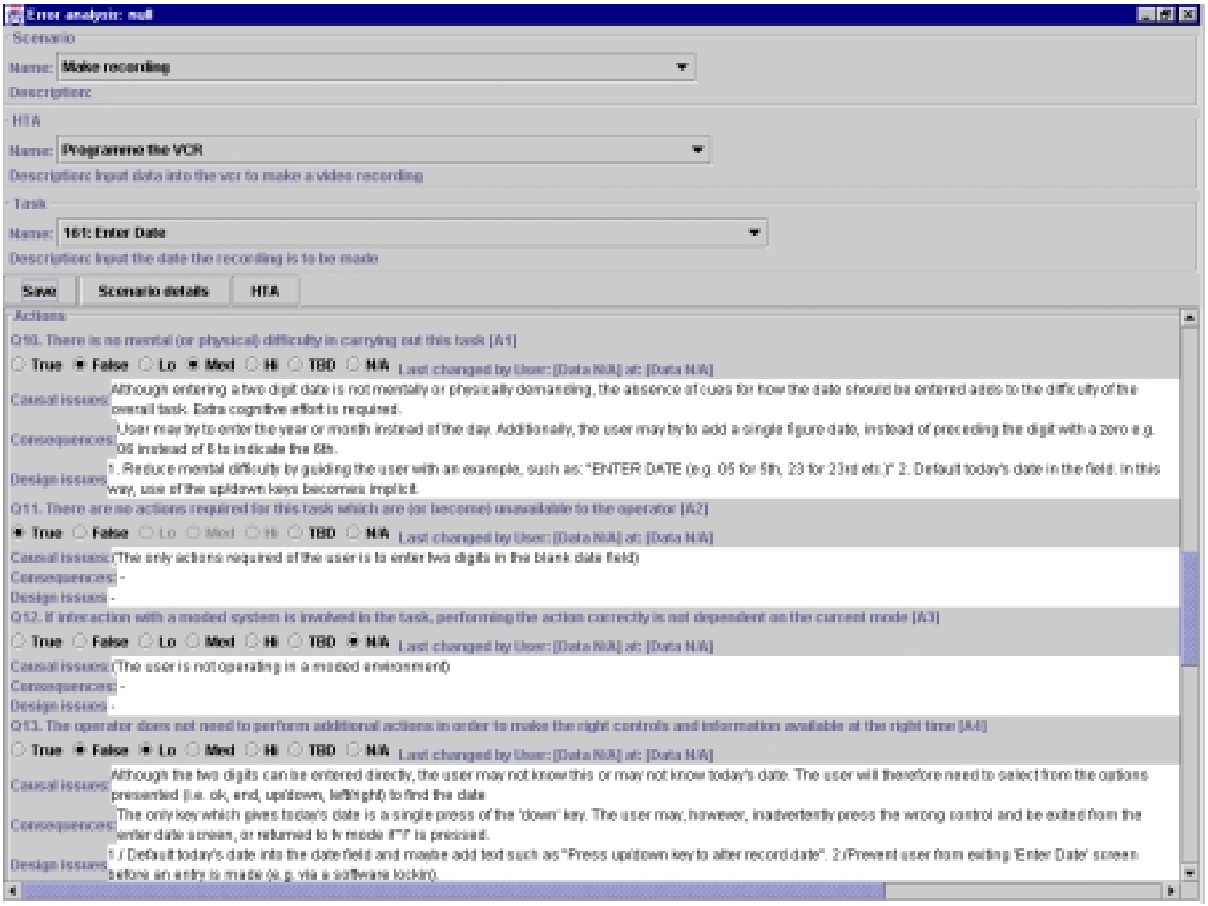
From such findings, user-centric design deficiencies can be established. If the design is still at an early stage of development, changes should be made. All, or part, of the error assessment process should then be repeated (see Figure 1).

ProtoTHEA tool support, as described briefly in Appendix A – Tool support with ProtoTHEA., works in exactly the same way as the 'manual' method described above except that an output diagram, known as a "Problem state profile chart", is provided, summarising and weighting the severity of each potential concern identified.

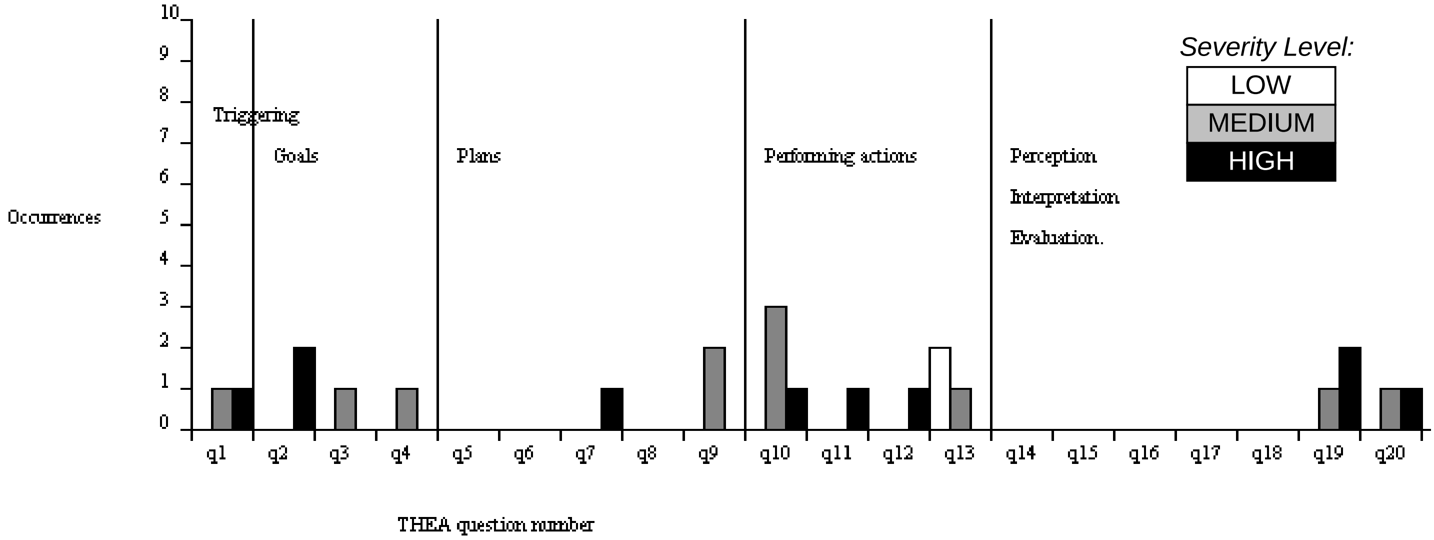
10 Appendix A – Tool support with ProtoTHEA.

When conducting larger and more complex case studies, a need was identified for tool support to assist with error analysis. This resulted in the development (Johnson, 1999) of "ProtoTHEA" , a prototype tool where, in addition to error analysis details, scenario and HTA information for each project is entered and stored via a graphical user interface, with all data being stored in a user-transparent database. For each scenario, an output in the form of a 'failure state profile' is automatically created (see Screenshot 3). Such an output is intended to highlight areas of the design which the error analyses have identified as potentially problematic. The tool also tracks analysis (and any analyst) changes made during any design review and update sessions. This is considered vital for traceability purposes, especially for large scale projects where personnel may change throughout the design lifecycle.

Screenshot 2 shows a partial error analysis for the video recorder case study discussed in Section 9. Question 10 checks that "*There is no mental or physical difficulty in carrying out this task*". The analyst can answer 'True', 'False (adding whether it is considered to be Low, Medium, or High severity)', 'TBD' (to be decided, if no decision has been reached on this question), or 'N/A' if the question is not applicable for the current task. Space beneath each question allows for analysts' comments to be inserted as before i.e. 'Causal Issues', 'Consequences' and 'Design Issues'. This is strongly recommended.



Screenshot 2 - Extract of a completed ProtoTHEA Error Analysis questionnaire



Screenshot 3 - A typical ProtoTHEA 'Problem State Profile' chart output[3]

Outputs such as this afford a quick overview of potential problems, often by highlighting problem 'clumps'. In the screenshot, triggering and goals, plus certain actions, need further examination. Two questions concerning perception also need to be investigated. As mentioned earlier, the questionnaire should be revisited and design changes considered where necessary. The diagram may also help address, and raise issues, concerning more generic design concerns e.g. "Our moding concept may need to be re-appraised".

[3] Q1 - Q20 are exactly the same as G1-G4, P1-P4 etc. in the manual approach. Using consecutive numbers in ProtoTHEA facilitates database data manipulation.

11 Appendix B – Blank error analysis template

SCENARIO NAME:
GOAL/SUB-GOAL BEING ANALYSED:
ANALYST NAME(S) + DATE:

THEA EA Question	Causal Issues	Consequences	Design Issues (where appropriate)
GOALS, TRIGGERING & INITIATION			
G1 *(Is the task triggered by stimuli in the interface, the environment, or by the nature of the task itself?)*			
G2 *(Does the user interface "evoke" or "suggest" goals?)*			
G3 *(Do goals come into conflict?)*			
G4 *(Can a goal be achieved without all its 'sub-goals' being correctly achieved?)*			

PLANS			
P1 *(Can actions be selected in situ, or is pre-planning required?)*			
P2 *(Are there well practiced and pre-determined plans?)*			
P3 *(Are there plans or actions that are similar to one another? Are some used more often than others?)*			
P4 *(Can the user determine that the task is proceeding successfully towards the goal, and according to plan (if there is one)?)*			

ACTIONS			
A1 *(Is there physical or mental difficulty in executing the actions?)*			
A2 *(Are some actions made unavailable at certain times?)*			
A3 *(Is the correct action dependent on the current mode?)*			
A4 *(Are additional actions required to make the right controls and information available at the right time?)*			

PERCEPTION, INTERPRETATION & EVALUATION			
I1 *(Are changes to the system resulting from user action clearly perceivable?)*			
I2 *(Are the effects of such user actions perceivable immediately?)*			
I3 *(Are changes to the system resulting from autonomous system action(s) clearly perceivable by the user?)*			
I4 *(Are the effects of such autonomous system action(s) perceivable immediately?)*			
I5 *(Does the task involve monitoring, vigilance, or spells of continuous attention?)*			
I6 *(Can the user determine relevant information about the state of the system from the total information provided?)*			
I7 *(Is complex reasoning, calculation, or decision-making involved?)*			

I8 *(If the user is interfacing with a moded system, is the correct interpretation dependent on the current mode?)*			

12 Appendix C – Examples of Error Analysis questions

G1: *"Is the task triggered by stimuli in the interface, the environment, or by the nature of the task itself?"*

Instrument triggering: A dial needle moves into a red 'danger zone' and a bell sounds to alert the user to do something;

Environment triggering: Ice forming on an aircraft windshield elicits a response from the pilot to activate windshield de-icing.

G2: *"Does the user interface "evoke" or "suggest" goals?"*

An aircraft primary flight display shows waypoints (i.e. pre-determined goals) graphically, as well as current progress towards those goals.

G3: *"Do goals come into conflict?"*

A pilot of an aircraft flying at low altitude over the sea may suffer an engine fire (perhaps after suffering a bird-strike), and is forced to choose between closing the engine down and extinguishing the fire, and obtaining more thrust to remain above the sea.

G4: *"Can a goal be achieved without all its 'sub-goals' being correctly achieved?"*

<u>True</u>: Photocopying a sheet of paper ('the goal') can be achieved without first removing the original copy (and possibly the photocopying card as well, for non-cash machines);

<u>False</u>: A cash teller machine will not allow removal of cash ('the goal') until the user's bank card has first been removed ('the sub-goal').

P1: *"Can actions be selected 'in-situ', or is pre-planning required?"*

Walking down to a local store can be accomplished without first deciding which route to take i.e. it can be performed 'in-situ' or 'on the fly'. However, one day you hear that the normal route is impassable. Some form of advance planning will probably now be required.

P2: *"Are there well-practiced and pre-determined plans?"*

A pilot uses a 'normal' checklist during each phase of flight, every flight, to ensure that actions which must be completed for safe flight are actually performed. Similar use of an emergency checklist during non-normal conditions ensures appropriate actions are taken to rectify the abnormality, as well as helping to reduce mental workload on the pilot.

P3: *"Are there plans or actions that are similar to one another? Are some performed more often than others?"*

<u>Plan similarity</u>: A maintenance engineer is using a checklist to replace a *specialist* nut from a bolt that has corroded, but is erroneously using the checklist for nut removal from a *normal* bolt. Both checklists are very similar to each other, except that the specialist nut replacement checklist contains some important extra actions that must be carried out.

<u>Plan frequency:</u> A car driver uses their vehicle to drive to and from work each day. On one occasion, however, the driver must alter slightly the return journey to pick up some groceries. It is quite possible that the revised plan will be 'overwritten' by the more frequently conducted regular journey home and the groceries will be forgotten ('the task omitted').

P4: *"Is there feedback to allow the user to determine that the task is proceeding successfully towards the goal, and according to plan (if there is one)?"*

The global positioning system (GPS) on an aircraft flight deck alerts the pilot, both visually and audibly, that a particular intended waypoint has been reached. The task of navigating towards some ultimate waypoint is proceeding successfully and according to plan (via pre-set waypoints towards the final destination).

A1: *"Is there any physical or mental difficulty in executing the actions?"*

A pilot is required to perform a 'miles' to 'kilometres' conversion during a high workload phase of flight.

A2: ***"Are some actions made unavailable to the user at certain times?"***

A computer user creating a document in a word processor, clicks with the mouse on a drop-down menu to access the 'Print' command. However, this command is 'greyed out' and is unavailable to the user. Subsequent investigation reveals that the printer had previously been removed for servicing and had not been reconnected. After plugging in and switching on, the same drop-down menu now indicates that the 'Print' command is available for use (i.e. it is not greyed-out).

A3: ***"Is the current action dependent on the current mode?"***

A typical domestic clock-radio alarm requires the user to set each function (time/clock/alarm) separately via specific button presses. Setting a radio channel, for example, requires the user to be in 'radio' mode – if not, the user might unwittingly be altering the clock time.

A4: ***"Are additional actions required to make the right controls and information available at the right time?"***

A mining engineer about to detonate an explosive charge must first insert and rotate an arming key in the firing control interface. When done, a lamp illuminates indicating that the fire control button is 'available' i.e. live.

I1: ***"Are changes to the system resulting from user action clearly perceivable?"***

True: Changing television channel
False: Fine tuning a television set for strongest signal/least noise

I2: ***"Are the effects of such user actions perceivable immediately?"***

True: Turning up the volume control on a radio
False: Turning up a central heating thermostat on a cold day. The house will not feel warmer until after a considerable delay.

I3: ***"Are changes to the system resulting from autonomous system action(s) clearly perceivable by the user?"***

A computer user receives an email message and is notified of its arrival by a screen flash and a brief sound.

I4: ***"Are the effects of such autonomous system action(s) perceivable immediately?"***

A domestic central heating system switches itself on at a pre-programmed time. A change in the ambient temperature will not, however, be noticed until some time later.

I5: ***"Does the task involve monitoring, vigilance, or spells of continuous attention?"***

Measuring out a specific weight of a recipe ingredient requires continuous monitoring of the dial on the weighing scales. If attention is distracted from this task, excess ingredient may be inadvertently added which may go unnoticed and/or cause an overflow from the measuring bowl.

I6: ***"Can the user determine relevant system information about the state of the system from the total information provided?"***

A motorist determines from the dashboard fuel gauge that a refuelling stop is required as the needle approaches 'Empty' and a low fuel warning lamp illuminates.

I7: ***"Is complex reasoning, calculation, or decision-making required?"***

A motorist driving in a foreign country using a different distance and fuel measuring system must perform a conversion calculation in order to judge distances and to put an appropriate quantity of fuel in the tank. The fuel grades may also be labelled in a manner unfamiliar to the motorist, and careful decision-making must be performed in order not to put the incorrect fuel into the vehicle with potential for damaging the engine.

I8: ***"If the user is interfacing with a moded system, is the correct interpretation dependent on the current mode?"***

Input moding: Pressing the 'Play' button on a domestic hi-fi system might activate the CD player instead of the desired cassette deck. Pressing 'Play' will activate only the device which is currently selected on the front panel i.e. it is mode dependent;

<u>Output moding:</u> The same hi-fi unit may also be indicating 00:37:22 in the display window, with the rightmost (least significant) digits incrementing in second intervals. The user may assume that the CD has been playing for 37m 22s when in fact the display is in ‘Clock’ mode, thus indicating that the time is approaching twenty to one in the morning.

13 Appendix D – Non-scenario-based evaluation approaches

In this report we have shown how scenarios can help us anticipate ways in which a system under design may be used. Here we acknowledge that there are many other, more general, aspects of good interface design, but which fall outside the scope of this report. However, we briefly discuss two of the most important when assessing a design:

1. The ways in which superficial aspects of an interface may confuse a user;
2. The effects of restricting a user's authority.

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. A number of questions about a design can be asked to help expose the potential for problems. If the answer to any of the questions is "yes", then there may be a problem with the interface under scrutiny.

- **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 awkward command sequences, manipulations of data, or perceptual/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?

Authority limiting

Another highly relevant aspect of a system's behaviour when considering interaction error is the way in which constraints are imposed on what an operator is able to do to the system. Limiting a user's authority to a "safe" or "acceptable" influence is often incorporated within a design to prevent or reduce the likelihood of particular errors. For example:

- **Lock-ins** – prevent actions from being omitted from a sequence, or keeping an operation active, preventing someone from terminating it prematurely (example: cannot remove money from a cash point machine without first removing bank card).

- **Lock-outs** – prevent events from occurring (example: a barrier to the basement of a building from the ground floor, so in an emergency people are prevented from attempting to exit at the wrong level).

- **Interlocks –** prevent certain sequences of actions being carried out or certain states from being reached i.e. they force operations to take place in proper sequence (example: a microwave oven door which disconnects power if opened during operation. The pin on a fire extinguisher to prevent accidental use of the device).

- **Guards** – make certain high-consequence actions harder to perform or make them involve a number of sub-actions (example: physically guarding important switches, or a ‘confirm’ box on making a ‘file deletion’ selection.

- **Protections** – allow the human to carry out actions but limit the effect they can have on the controlled process (example: an aircraft flight control system can provide protection against stalling, over speed, and so on).

Bibliography

Dix, A., Finlay, J., Abowd, G. and Beale, R. (1998) *Human-Computer Interaction*, Prentice Hall.

Fields, B., Harrison, M. and Wright, P. (1997) THEA: Human Error Analysis for Requirements Definition. Technical Report (YCS294), University of York.

Hollnagel, E. (1998) *Cognitive Reliability and Error Analysis Method CREAM*, Elsevier Science.

Johnson, P. (1999) A Tool for Human Error Assessment. M.Sc. Project. *Department of Computer Science*, University of York.

Kirwan, B. (1994) *A guide to practical human reliability assessment.* Taylor and Francis.

Kletz, T. A. (1983) *Hazop and Hazan - Notes on the Identification and Assessment of Hazards.* Institution of Chemical Engineers.

Kyng, M. (1995) Creating Contexts for Design. In *Scenario-Based Design: Envisioning Work and Technology in System Development* (Ed, Carroll, J. M.) Wiley, pp. 85-107.

Loer, K., Harrison, M. and Wright, P. (1999) *Checking Mode Complexity and Other Properties in Interactive Systems.* DCSC Technical Report. University of York.

Norman, D. A. (1988) *The psychology of everyday things*, Basic Books.

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