THEA – A Technique for Human Error Assessment Early in Design

Steven Pocock, Peter Wright, Michael Harrison[*]
Department of Computer Science
University of York
Heslington, York. YO10 5DD, UK

SUMMARY

Human activity constitutes a major source of vulnerability to the integrity of interactive systems. Wherever human actions are either inappropriate, incorrect, or erroneous, there will be implications for design. This is especially true in high risk endeavours such as commercial air and marine transportation, power production, medical care and space flight. The aim should therefore always be to design an interactive system as resilient to human erroneous actions as possible, and to achieve this as early as possible in the design phase. We present in this paper a formative error assessment technique contributing to the achievement of this goal, known as the Technique for Human Error Assessment (THEA). The method has been applied to several real-world case studies and has demonstrated its suitability in evaluating a design for its vulnerability to human interaction failures which may become problematic once the design becomes operational.

Keywords

THEA, scenario, cognitive failure, error analysis

INTRODUCTION

It has been estimated [2] that approximately 60-90% of all system failures are the direct consequence of human erroneous actions. The concern for safe and reliable performance has understandably been especially high in the nuclear power industry where techniques such as Probabilistic Safety Assessment (PSA) and Human Reliability Analysis (HRA) have been extensively employed. Other methods for assessing the impact of erroneous human actions on interactive systems have since appeared – some qualitative, others quantitative, but it is not intended in this report to review such methods. A brief discussion of some of these can be found in, for example, [7] [8] [2]. The THEA method described in this paper, has its roots in the class of methods of HRA and is designed to inform human-machine interface (HMI) design at an early stage of development.

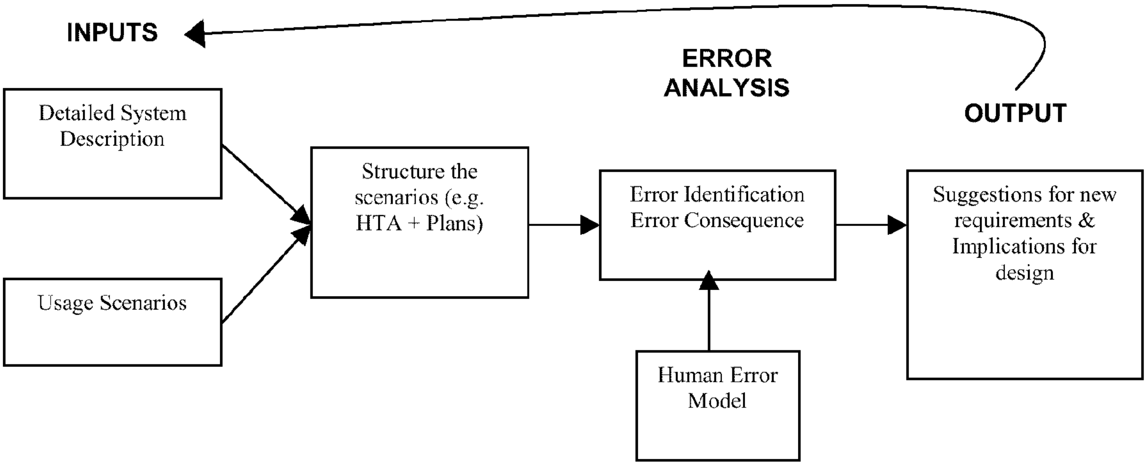

INPUTS
ERROR ANALYSIS
OUTPUT
Detailed System Description
Usage Scenarios
Structure the scenarios (e.g. HTA + Plans)
Error Identification Error Consequence
Human Error Model
Suggestions for new requirements & Implications for design

Figure 1: The THEA process

[*] *Current address: DERA, Centre for Human Sciences, Farnborough, Hants GU14 0LX, UK*

Paper presented at the RTO HFM Workshop on "The Human Factor in System Reliability – Is Human Performance Predictable?", held in Siena, Italy, 1-2 December 1999, and published in RTO MP-032.

THEA possesses some similarities with formative evaluation techniques such as Cognitive Walkthrough [9]. In contrast however, THEA aims to consider not only problems with information presentation and feedback but also problems with the planning and execution of actions. THEA also takes a hierarchical view of goals and actions in addition to the sequential perspective of Cognitive Walkthrough. THEA is a strongly suggestive method, guiding the analyst in a structured way to consider areas of a design for potential interaction difficulties. Other methods, such as the *human error identification in systems tool* (HEIST) described in [5], possess similar goals to THEA, except that THEA achieves them with considerably less exertion – eighteen error analysis questions as opposed to 113, which is perhaps why the latter approach has remained largely theoretical. It would certainly be impractical to use without tool support, whereas THEA has the capability of conducting sizeable analyses by means of a prototype tool called ProtoTHEA.

The basic philosophy of THEA views errors as contextualised phenomena influenced by, for example, performance shaping factors. Thus for any method to effectively assess a design for vulnerability to error, it must take account of context. THEA explicitly takes contextual and cultural issues into consideration by means of usage scenarios. In this way it is hoped to elicit the way work is *actually* practiced and not simply how designers *envisage* it as being practiced.

We commence with an overview of THEA, followed by a case study to illustrate the technique.

THEA

The main aim of THEA is to use systematic methods of asking questions and exploring interactive system designs based on how a device functions in a scenario. The purpose of doing this is to provide a systematic and structured means of critiquing a design and developing further requirements [1]. In this way, it is hoped to assist system designers anticipate human interaction failures which may become problematic once a design becomes operational. The technique is intended primarily for use early in the development lifecycle whilst functionality is emerging, and begins with a formal description of the work under analysis. This is achieved by combining two primary inputs consisting of a detailed description of the design under consideration – preferably with domain expert input – and a number of usage scenarios. These inputs, together with the remainder of the THEA process, are illustrated in Figure 1.

Scenarios

THEA views performance failure as an attribute of "cognition in the world" [4], that is to say, of the context or the circumstances which play a fundamental role in its methodology. Applying a communications analogy (op.cit.), performance conditions – or context – may be thought of as the 'signal', with erroneous human actions as 'noise' superimposed on it. Too little signal and the communication becomes unintelligible. Thus by analogy, with insufficient context, performance failure becomes less meaningful. THEA analyses attempt, through use of detailed scenarios, to capture those complex conditions which result in the human behaving in an unanticipated and unintended manner.

Scenarios should thus comprise not only *actions* which take place in a given situation, but also *contextual factors* which surround the action, allow it to happen, and provide opportunities for "error". To represent the context as comprehensively as possible, a scenario template in [1] incorporates the following information:

1. Agents
   - The human agents involved and their organisation
   - The roles played by the humans, plus their goals and responsibilities
2. Rationale
   - Why is the scenario interesting?
3. Situation and Environment
   - The physical situation in which the scenario takes place
   - External and environmental triggers, problems and events that occur in this scenario

4. Task Context
   - What tasks are carried out?
   - What formal procedures exist, and are they followed as prescribed?
5. System Context
   - What devices and technology are involved? What usability problems might they possess?
   - What effect can users have?
6. Action
   - How are the tasks carried out in context?
   - How do the activities overlap?
   - Which goals do actions correspond to?
7. Exceptional circumstances
   - How might the scenario evolve differently?
8. Assumptions
   - What, if any, assumptions have been made?

Principal sources for scenario elicitation include:

- Experience with earlier versions of the system. 'Top-down' designs are relatively infrequent and previous versions usually have associated reports highlighting problem areas;
- Incident and accident reports;
- Frequent conditions and normal operation;
- Where technology changes. This is the principal source for the case study presented in this paper;
- Where concepts change. For example, changing from conventional air traffic control to Datalink.

Finally, we want to know how many scenarios will be required to capture the usage context in sufficient detail. The answer is really reliant upon expert judgement as to when a 'good enough' coverage has been achieved, and for this reason it is highly desirable to have at least one domain expert involved in the scenario construction process.

Goal Decomposition

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. It is hierarchical because task goals are broken down into a structure of sub-goals which must first be achieved before the top level goal can be satisfied. In this way we can describe operators' tasks in terms of the goals and sub-goals to be achieved and the actions used to achieve these goals. Plans are appended to each task to describe the flow of control through the task and detailing *how* the sub-goals and actions within a task are combined to satisfy the higher level goal.

Task descriptions, while good at describing what a user has to do and know, is less adept at describing how an interface might respond to a user's inputs. THEA presumes that some notion of causality can be used to explore the interaction between for example, a display and other perceptual cues, operator memory requirements, and other aspects of the design. A set of behavioural analysis guidewords (omission, commission, and so on) is employed, based on a control model of operator-system interaction [6]. These can trigger questions about the extent to which, for example, a display is able to support goals and plans, or to consider how apparent it would be for an operator to perform an appropriate action. We believe this affords a means of linking task and system descriptions more directly, and forms the basis of the THEA error analysis phase.

Error Analysis

The foregoing steps identify a number of factors facilitating an understanding of the context in which human actions – and therefore erroneous actions – take place. We are now in a position to draw these strands together in the analysis phase which helps identify where HMI error may be problematic.

The analysis adopts a structured questionnaire-, or checklist-, style approach, referred to in [1] as the "Cognitive Error Analysis". This is based on failures (Table 1) that are possible in Norman's execution-evaluation cycle model of human information processing [6].

Table 1: Examples of cognitive failure

Stage	Cognitive failure
Goals	Lost/Unachievable/Conflicting No triggering/activation Triggering/activation at wrong time, or wrong goal activated
Plans	Faulty/Wrong/Impossible
Actions	Slip/Lapse
Perception/ Interpretation	Failure to perceive correctly Misinterpretation

The error analysis poses questions about the scenario to reveal areas of design where cognitive failures may occur, and assess their possible impact on the task or system being controlled. A simple example might be the high level goal of photocopying a sheet of paper. One of the THEA analysis questions asks whether the goal can be accomplished without all its sub-goals being correctly achieved. The analyst would typically answer (in the case of most photocopiers) "yes" since it is entirely possible to walk away with your copy but leave the original document and/or copier card in the machine. The sub-goal has thus been lost and a 'post-completion' error has occurred. A full list of the THEA error analysis questions can be found in Appendix A.

There will be occasions when no obvious behavioural manifestations are evident. For example, if an operator is presented with conflicting goals, this may itself be a 'manifestation' of the problem which, if serious enough, may require a design solution to be found.

Exactly how the analysis is carried out is largely a matter of choice, but the two envisaged methods are:

1. Follow the goal hierarchical structure from top to bottom asking each question about each goal or action;
2. Select parts of the scenario where potential problems are anticipated, then conduct a detailed analysis of behavioural error and impact where appropriate.

Clearly the first option is the most thorough and is recommended for new designs. Understandably it is probably going to be lengthy and time consuming but also likely to uncover a greater number and range of concerns.

Recording the results

Whichever approach is adopted, the analysis results may be recorded according to project requirements. We have found, however, that a tabular format provides a practical way of presenting the information. **Table 2** shows a typical arrangement, while table 4 provides an example:

Table 2: Tabular format for recording EA results

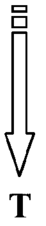
Question	Causal Issues	Consequences	Design Issues
Question identifier as an aid to traceability	Issues raised by analyst	Consequences of the causal issue	Notes, suggestions, comments, re-design ideas

From our own case study work, sometimes involving large and complicated scenarios, we identified a need for tool support to assist the analyst with the entry, handling, and storage of information associated with a project. This resulted in the development of ProtoTHEA, a prototype tool where the scenario, HTA, and error analysis details of a particular project can be entered via a graphical user interface. All information is held in a database and an output, in the form of 'failure state profile' charts (adapted from [8]), is automatically obtained for each scenario. Appendix B shows a typical ProtoTHEA HTA and error analysis example, as well as a failure state profile chart.

APPLICATION OF THEA – CASE STUDY

We now illustrate a practical application of THEA by means of a case study, based on information collected from flight crew, involving a change of technology on the flight deck of a fisheries reconnaissance aircraft. A major change between the old and the new flight decks concerns the crew complement being reduced from three people to two, the flight engineer being replaced by computerised technology. The scenario involves a situation where the activities of the flight engineer would, on the old flight deck, be particularly significant. We deal with emergency conditions rather than normal operation, but since the tasks in themselves are fairly straightforward and do not involve much decision making, the crew activities involve more knowledge intensive activities such as fault diagnosis.

Table 3: Scenario timeline showing actions – some conflicting – performed by each agent



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

Situation and environment

The starting condition involves a four-engine fisheries patrol aircraft at low level over water, photographing a fishing vessel. To conserve fuel, the aircraft is flying on engines 2,3,4 only. Engine 1 (leftmost) has been closed down for fuel economy reasons. The aircraft suffers a massive bird strike on the right side. As a result of bird ingestion to engines 3 and 4, both 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 warning or cautions being signalled after a short delay.

Actions in context

As we discussed earlier, one of the principal components of a scenario is a description of the actions which take place. An HTA may be employed, but it is not always necessary. If, for example, interaction with the system of interest is relatively simple, then it is probably sufficient to simply identify the goals users have, and write down a list of the actions necessary to achieve the goals. If the interaction is more complex, then a more formal approach for capturing tasks and goals, such as HTA, may be needed. For this scenario, we adopt the former approach since it is not the intention here to produce a fully worked example, rather to give a flavour of how the technique may be used.

In Table 3 we show some of the crew and 'system' actions in the early stage of the scenario, with time flowing downwards. What is interesting is that one can observe both pilots conducting possibly contradictory actions at the same time – the PF is attempting to restart engine 1 to produce more thrust, while the PNF is shutting down the faulty engines i.e. *reducing* thrust. However, what this diagram does *not* show are links between actions and the surrounding context, which is a main reason for thinking about scenarios in the first place. To accommodate this, Table 3 may be modified to include the goals – derived from the task analysis – to which they are directed. Figure 2 shows a goal structured action sequence for our scenario with time now represented qualitatively along the horizontal axis. The same actions as before are shown but, in addition, the goals that drive the interaction – as well as triggers that bring the goals into being – can be seen. Presenting scenario actions in this way illustrates a number of features not immediately evident in, for example, a traditional HTA. In particular, Figure 2 shows which goals and tasks become active, and active concurrently in the scenario, as well as which actions are related by being directed towards the same goals. These are not present in the simple event listing of Table 3 which makes no mention of goals.

Analysis example

An illustration of how the analysis is conducted is shown in Table 4. We have selected only two of the questions from the full cognitive error analysis question list (see Appendix A) which are particularly pertinent to this scenario, namely:

G1 – The mechanisms which trigger or activate goals, and
G3 – The potential for conflicting goals.

Asking question G1 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 a warning), whereas others are less directly triggered and may be more prone to being omitted (e.g. "Engine 3 cleanup"). A full version of the analysis is provided in [1].

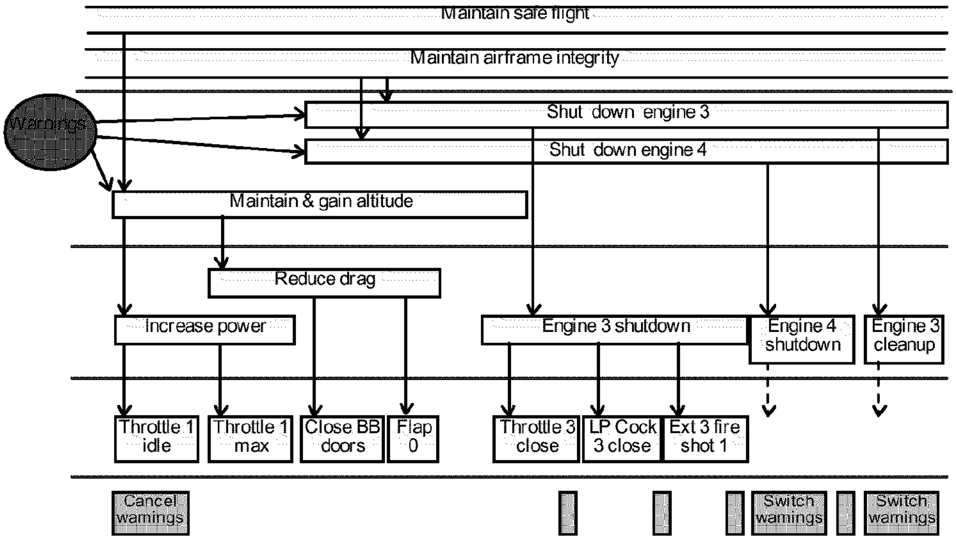


Figure 2: Hierarchical goal structuring of scenario actions

Table 4: Example application of error questionnaire

Question	Causal Issues	Consequences
G1 (Triggers, task initiation)	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 are 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 is also possible that "Engine 4 shutdown" or "Engine 3 cleanup" might be omitted or delayed.
G3 (Goal conflicts)	Goals to increase power and Engine 3 shutdown are in conflict (although this is inevitable)	Resolving the conflict satisfactorily requires negotiation between PF & PNF. The time required for this may lead to a non-optimal (too late) decision.

When performing a full analysis, causal issues raised producing noteworthy or problematic consequences are documented in the 'consequences' column. Entries for certain questions might be left blank, indicating that the question did not appear to reveal any interesting insights. A third column could also be added entitled "Design Suggestions". Thus we might add to G3 "Attempt to design out conflicts or give participants the resources to resolve them", and so on.

Finally, it is worth mentioning that certain 'keywords' (omission, commission, etc.) will not make sense in the context of every scenario. For example, a 'repetition' error is not-applicable to an aircraft's take-off sequence. In other cases, physical constraints may make it impossible, or it would be hard to imagine how such deviations might occur.

THEA & QUANTIFICATION

The primary output of THEA is a description of a number of problem areas associated with a design and its operation which may be the cause of interaction errors. These are intended to assist designers reason about errors at the early stages of a design before it becomes impractical or prohibitively expensive to effect a longer term design change or implement shorter term procedural 'fixes' or limitations.

Unlike some hazard identification methods such as hazard and operability studies (HAZOP), THEA does not directly identify hazards per se but instead addresses the causal factors which contribute to them. That is, it does not provide quantitative estimates of the likelihood of human erroneous actions. This is not to say that the method proscribes the use of supplemental quantification where useful or necessary. For example, THEA has been supplemented in certain case studies by the Human Error Assessment and Reduction Technique (HEART) [10]. This is a task-based approach utilising a database of error probabilities, and relying on the application of a simple algebraic formula to a chosen generic task and weighted error producing conditions (EPCs). It has demonstrated its usefulness in supporting THEAs qualitative output by allowing us, where a number leads to a concern, to ask:

- Have we chosen the wrong generic task?
- Have we chosen inappropriate EPCs?
- Have we weighted the EPCs disproportionately?

In this way, our assumptions, both qualitative and quantitative, may be reflected upon and revised if necessary. The advantage of a supplemental method such as HEART is that it is readily understandable by all interested parties and is a way of supporting dialogue about human reliability estimates.

While numbers may be useful, it is important to be clear how they are intended to be used. We must also be quite clear what they represent and to whom. For example, the 'traditional' engineering view regards numbers as representing real values of probabilities which may be combined and manipulated arithmetically. In our experience, numbers represent broad categories of risk and serve as 'tokens' for the negotiation of concerns ("Do we have a problem?" or "I think your estimate for this error is unrealistic"). That is to say, numbers should not be treated as objective truths but rather as starting points for discussion. Superficially, qualitative and quantitative predictions are different outcomes, but it will be appreciated that they are actually opposite sides of the same coin. As Hollnagel [3] points out:

> *"Quantification can only be done for something that has been clearly identified and described, and this description must necessarily be qualitative. Quantities must be quantities of something, and that something must be previously described."* (p.80)

Whilst it may be argued that a quantitative approach is necessary to support and satisfy conditions of, for example, a Probabilistic Safety Assessment (PSA) or a specific customer requirement ("No single failure shall have a catastrophic or critical hazardous consequence in every 10^9 hours"), it is uncertain whether, or to what extent, such an approach actually matches reality. All quantitative methods are ultimately based on a qualitative description and some underlying model. It follows that if any of the descriptive steps are lacking, the outcome of any numerical analysis will necessarily be incomplete no matter how refined the quantification process.

DISCUSSION

This paper has described a formative error analysis technique, THEA, for analysing system vulnerability to erroneous human actions. One of the most important antecedents of the THEA error analysis process is gaining an understanding of how the system being examined will be used in practice. We formulate 'usage scenarios' to furnish us with context of use – the circumstances or conditions under which an event occurs – to elicit how work will *actually* be performed as opposed to how it is *envisaged* it will be performed.

It is highly desirable to carry out an analysis early in the design process before adverse consequences are encountered at 'the sharp end'. THEA anticipates, through design critique, interaction failures which may become problematic once a design is operational. In such a way it can assist in developing further requirements before a design becomes 'rigid' and excessively difficult or expensive to modify. We differentiate between cause and consequence since incorrect operator actions and assessments are treated as the *starting* point for analysis rather than the conclusion – they are recognised as *symptoms* rather than causes. In this predictive role, causes are the initiating events and manifestations are the possible outcomes. Of course, THEA works equally well for retrospective analyses of extant designs. A recent case study employed the technique to appraise a system where specific erroneous operator actions would result in serious consequences. THEA highlighted system design issues contributing to such performance as well as providing an assessment of possible consequences. Our results supported the clients' numerical analysis thus affording a more confident design assessment. In addition, the case study facilitated convergence of practitioners and human factors personnel through the exchange of ideas and techniques. This helped overcome what Hollnagel refers to in [3] as "the *conceptual impuissance or abstruseness*".

We have found from experience that, although no special expertise is required to carry out the error analysis procedure, input to the process by domain experts significantly expedites its completion. Additionally, tool support offered by ProtoTHEA has demonstrated an ability to manage large and complex case studies. Whether the 'traditional' or tool-assisted approach is employed, the emphasis of THEA is on functionality and practicality, both ably demonstrated in recent case study work.

REFERENCES

1. Fields, B., Harrison, M., & Wright, P. (1997) THEA: Human Error Analysis for Requirements Definition. YCS 294 (1997)
2. Hollnagel, E. (1993) *Human Reliability Analysis: Context and Control.* Academic Press Limited.
3. Hollnagel, E. (1998) *Cognitive Reliability and Error Analysis Method CREAM.* Elsevier Science Ltd.
4. Hollnagel, E. *The human factor in system reliability: Is human performance predictable?*, in *Anticipating failure: What should we make predictions about?* 1999, NATO RTO HFM Workshop: Certosa di Pontignano, Siena, Italy.
5. Kirwan, B. (1994) *A guide to practical human reliability assessment.* Taylor and Francis.
6. Norman, D.A. (1988) *The psychology of everyday things.* Basic Books.
7. Reason, J. (1990) *Human error.* Cambridge.
8. Reason, J. (1997) *Managing the Risks of Organizational Accidents.* Ashgate Publishing Ltd.
9. Wharton, C., et al. *The Cognitive Walkthrough method: A practitioner's guide*, in *Usability Inspection Methods*, R.L. Mack and J. Nielsen, Editors. 1994, John Wiley and Sons, Inc. p. 105-140.
10. Williams, J. (1986) HEART - A proposed method for assessing and reducing human error. *Proceedings of 9th Advances in Reliability Technology Symposium.* (1986, University of Bradford).

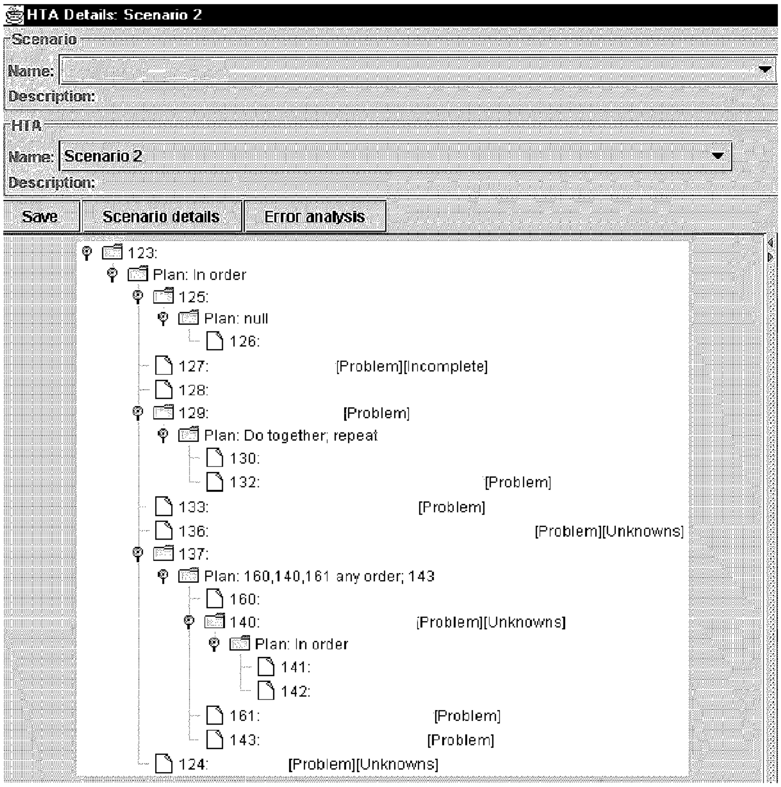
APPENDIX A – THEA error analysis questions

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.	

APPENDIX B – ProtoTHEA example: HTA and error analysis extract

The diagrams below show typical extracts from the ProtoTHEA tool. For the HTA in Screenshot 1, specific tasks have not been labelled for clarity, permitting illustration of feedback to user as to the status of each task. This enhances traceability and completeness. The error analysis extract in Screenshot 2 shows a typical screen presented to an analyst, demonstrating the questionnaire nature of the process. All respondent data is stored automatically, and the resultant failure state profile chart for each scenario is shown in Screenshot3.

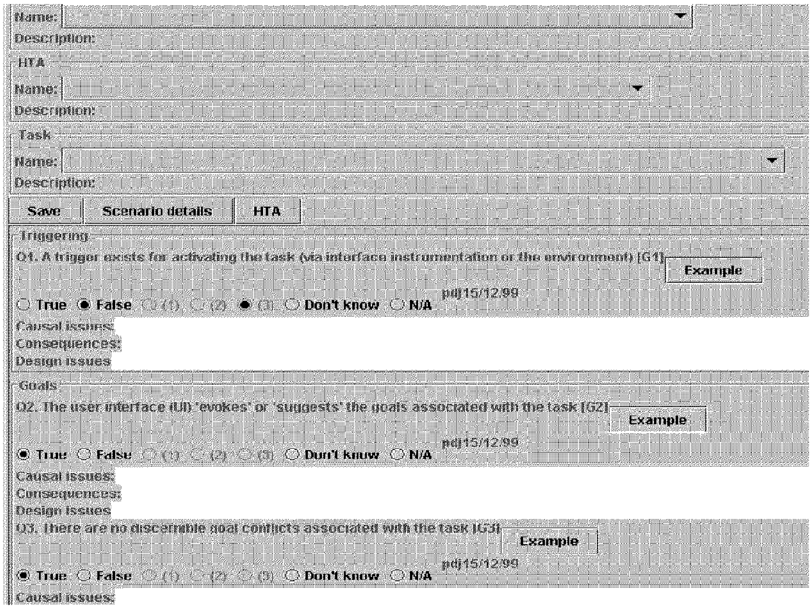


Name: Scenario 2

Description:

```
123:
  Plan: In order
    125:
      Plan: null
        126:
    127:                    [Problem][Incomplete]
    128:
    129:                    [Problem]
      Plan: Do together; repeat
        130:
        132:                                  [Problem]
    133:                              [Problem]
    136:                                        [Problem][Unknowns]
    137:
      Plan: 160,140,161 any order; 143
        160:
        140:                          [Problem][Unknowns]
          Plan: In order
            141:
            142:
        161:                          [Problem]
        143:                          [Problem]
    124:              [Problem][Unknowns]
```

Screenshot 1 – Hierarchical Task Analysis (HTA) extract



Q1. A trigger exists for activating the task (via interface instrumentation or the environment) [G1]

pdj15/12/99

○ True ● False ○ (1) ○ (2) ● (3) ○ Don't know ○ N/A

Causal issues:

Consequences:

Design issues

Q2. The user interface (UI) 'evokes' or 'suggests' the goals associated with the task [G2]

pdj15/12/99

● True ○ False ○ (1) ○ (2) ○ (3) ○ Don't know ○ N/A

Causal issues:

Consequences:

Design issues

Q3. There are no discernible goal conflicts associated with the task [G3]

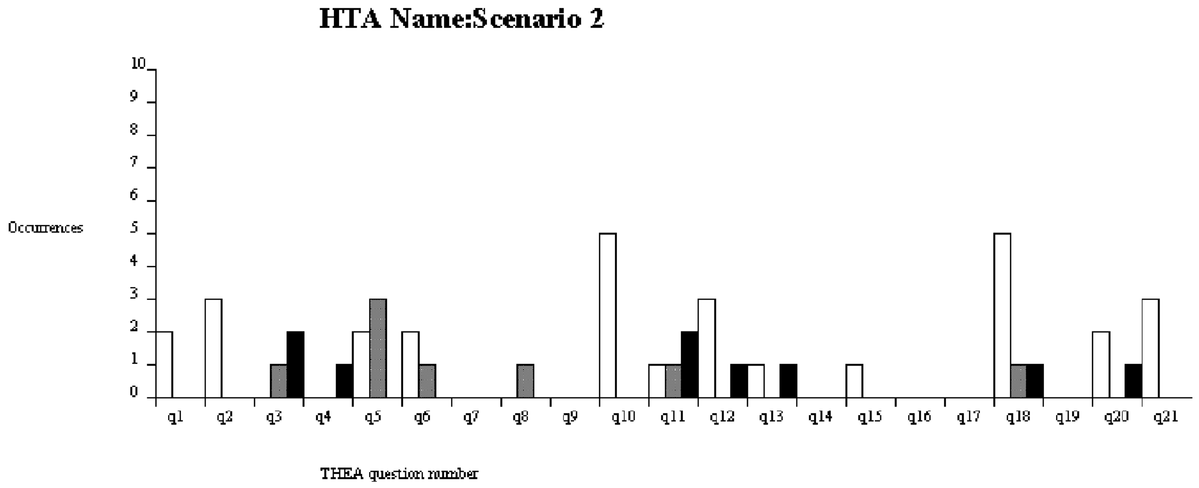
pdj15/12/99

● True ○ False ○ (1) ○ (2) ○ (3) ○ Don't know ○ N/A

Causal issues:

Screenshot 2 - Error Analysis extract

APPENDIX B (continued)



Screenshot 3 – Failure state profile chart for Scenario