

Task Analysis for Error Identification.

Chris Baber¹ and Neville Stanton²
The University of Birmingham and Brunel University

¹Electrical, Electronic and Computing Engineering, The University of Birmingham,
Birmingham. B15 2TT. *Email: c.baber@bham.ac.uk*

²Dept. of Design, Brunel University, Runnymede Campus, Englefield Green, Egham,
Surrey, TW20 0JZ. *Email: neville.stanton@brunel.ac.uk*

Human error is a significant contributor to product failure. However, it is uncommon for designers to explicitly consider the potential for human error in the design of products. In this chapter, it is proposed that ‘human error’ arises as a consequence of the interaction between user and product, and that modeling this interaction can allow insight into possible error paths. Using a simple representation of product functioning, based on state-space diagrams, Task Analysis for Error Identification indicates paths between states that are open to the user but which do not support the achievement of the user’s goal; such paths are considered to be erroneous. From this perspective, one of the aims of product design is to minimize paths to error.

Keywords: task analysis, human error, state-space diagrams, product design, product evaluation.

Introduction

Product evaluation often involves user trials. However, such trials require a prototype to be available. Admittedly, the prototypes need not be fully functioning products, but there may well be a need to have made most of the major design decisions prior to building such prototypes. This means that prototypes might already reflect significant design decisions and assumptions concerning *how* the user will interact with the product. An alternative approach is to perform analytic evaluation of product concepts. In other words, to develop predictive models of user performance, e.g., in terms of the time it takes to perform a sequence of actions or in terms of possible user errors, and to use these models as a means of evaluating design ideas. In this manner, designs can be compared and evaluated prior to committing to a specific concept. We term such an approach ‘analytical prototyping’ (Baber and Stanton, 2001).

Traditionally, human-computer interaction (HCI) has focused on models of user performance that are concerned either with performance time or with information-processing, as in GOMS (Card et al., 1983). While such approaches have proved useful, they tend to concentrate on ‘expert’, i.e., error-free, performance. Furthermore, the focus of such models has tended to be on the activity that the user performs on the product, rather than on the *interaction* between user and product. In other words, many of these models treat the product as a neutral object of user action, and do not consider the effects of changing product state on either user activity or user decision-making. Consequently, we set ourselves the challenge of developing a method that could describe user-product interaction, could focus on user error and could support analytical prototyping. Before presenting a worked example of the method in action, the chapter begins with a discussion of the assumptions underlying the method. These assumptions form part of a theory of human action, although as the reader will see, the theory remains very much “under development”.

Assumptions

The method that we have developed is based on a theoretical underpinning that reflects, we feel, current thinking about human-computer interaction. Significantly, we are struck by the commonly reported occurrence of people failing to read instructions or manuals prior to attempting to use a product. To us, this feels as if people are approaching the products ‘as if’ they were experts, i.e., as if they had a complete and appropriate repertoire of routines to use on the product. Thus, on receiving a new digital camera as a present, the prospective user might take the camera out its packaging, check that the batteries are inserted and then immediately try to take a picture; only when the interaction breaks down will our fictitious user turn to the user manual.¹ This implies that people either possess all the relevant knowledge needed to use the product (which might seem a little odd given that this is their first encounter with the specific product), or that they possess sufficient knowledge to imbue confidence in their actions when using the product. The first assumption is that (paradoxically, perhaps) users behave like experts; in other words, the user approaches the product with a repertoire of appropriate actions that, if successful, will make the product work. Only when the user does not feel that they possess this repertoire (whether through lack of experience or confidence), do they seek help from another source, such as a manual or a ‘local expert’. The question, therefore, is what happens in the user-product interaction to support such behaviour. In any interaction, a product can be assumed to present a set of functions to the user (via its features, its display, its controls etc.), in the form of a *system image*. The user conceptualizes a *representation* of how the product works, possibly in terms of pairing features with functions, e.g., the user might assume that pressing a specific button would turn the product on, or possibly in the form of *routines* based on previous experience. When there is a mismatch between the user’s representation and the system image, then the user could make errors or become frustrated.

Taking these notions of system image, representation and routines has led to development of the notion of rewritable routines. In order to present this notion, it is necessary to take a slight detour into the realm of *mental models*. It is well-known that there are many different and competing views on the nature of mental models. For example, Johnson-Laird (1980) presents mental models as temporary frameworks developed during the task of reading (or other tasks involving information extraction) in order to allow the reader to maintain the ‘gist’ of a story and to develop hypotheses of plot development or character motivation and action. In this form, the ‘model’ is developed on an ad-hoc basis and requires the recruitment of knowledge from long-term memory and its combination with information derived from the text. Consequently, the mental model represents a specific state of affairs in the text (or the world). Alternatively, the notion of mental models as representations of the workings of products appears to be popular in the HCI literature. This view can be traced at least as far back Craik’s (1947), on tracking control, which suggested that an operator needs to represent the system they are controlling in order to anticipate its dynamics. Researchers might accept that mental models allow users to make inferences about the system with which they are interacting, and that a mental model provides a ‘problem space’ (in Newell and Simon’s, 1972 terms). However, it is generally accepted that such mental models are imprecise, incomplete and inconsistent. Consequently, the problem space might itself be messy and problematic. O’Malley and Draper (1992) have suggested that, rather than the mental model containing a representation of the product, what is required is some means of filling the gaps in

¹ This, of course, presents one view of how people interact with new products. An alternative view is that the prospective user reads the manual from cover to cover, fully digesting any instructions and guidelines, in order to acquire sufficient knowledge to use the product, and only then begins to take pictures. The reality probably lies between these extremes. However, it is suggested that both versions present the ‘user as expert’ (one by dint of experience, the other by dint of reading).

the system image and interpreting any information provided by the product. The proposals of O'Malley and Draper (1992) have more than a passing resemblance to the suggestions of Johnson-Laird (1980), and we take this as the starting point for our concept of rewritable routines.

To conclude this brief discussion, users' rely on highly fragmented knowledge of the product, rely on a variety of metaphors to contrast the product with other products and can be heavily influenced by the system image. This suggests that 'mental models' need not be complete, coherent internal representations of a product (i.e., not a 'model' in any physical sense) nor that 'mental models' are sufficiently coherent to predict the consequences of actions (i.e., not a 'model' in a mathematical sense).

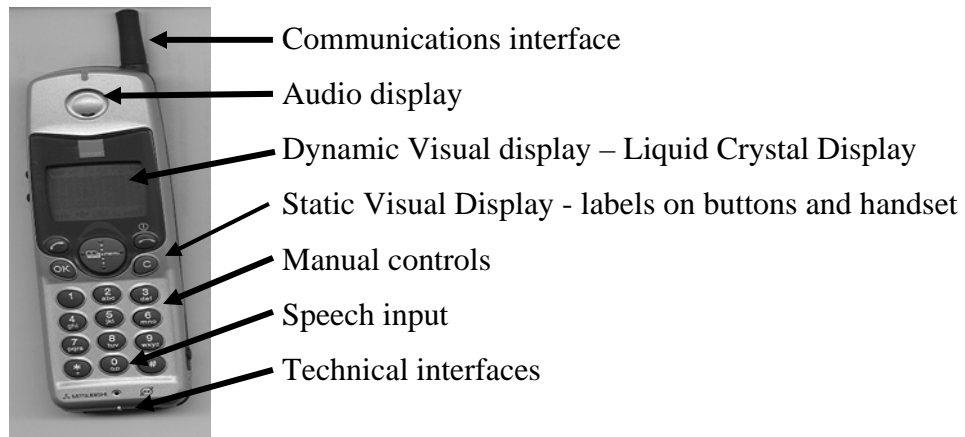


Figure 1: System Image of a Mitsubishi mt401 mobile telephone

By way of introduction to the notion of rewritable routines, consider the mobile telephone shown in figure 1. The System Image presents several types of information to user, in a variety of formats. Some information is permanently accessible, e.g., the static visual display, the technical and communications interfaces, while other information requires user action to be accessed, i.e., the dynamic visual display and audio display. User actions can be performed using manual controls or speech.

From the previous discussion, it is proposed that the 'system image' *implies* a set of routines that the person can use, and that the selected routine will depend on the user's goal and on the user's previous experience. In order to explore this point, assume that a person who has not previously encountered this model of mobile telephone is set the simple goal of turning it on. This will require the user to identify a button or key that can be expected to turn on the telephone. The user might examine the side or top of the telephone to look for an 'on' button, the user might see the 'green, lifted-handset' icon and assume this means make a call (and by analogy, turn on), the user might recognise the $\textcircled{1}$ icon as an ISO symbol for 'on', the user might assume that 'OK' meant turn on or could press any of the other keys on the basis of some other assumption. The point to be made is that, for the first-time user there are multiple possible routines that can be activated in the goal of turning on the telephone. The selection of a routine will be influenced by prior experience and by the system image. One could imagine that a given user might assume that the 'green, lifted handset' indicates the initial action that one makes using a 'conventional' telephone. In other words, the user brings a 'routine' to the use of the telephone that incorporates the following sequence of tasks: "pick up handset, dial number, hold conversation, replace handset".

The 'system image' provides cues to the first step in this sequence of tasks, i.e., 'pick up handset'. Of course, this is the wrong action and the user should press the key labelled with a 'red, lowered handset', i.e., on this model of mobile telephone, one begins by 'hanging up' (by way of explanation, notice the 'on' symbol above this key). The users choice of action will be based on the appearance of the telephone (its system image) and the user's prior experience, which leads them to assign meanings to the keys, icons and other objects in the system image.

We assume that people require very little knowledge of the internal workings of a product in order to use it. Thus, one does not need to know how a car engine works in order to drive the car. We also assume that all interaction with products will be goal-based and purposeful, i.e., that people have a reason for using the product and that they seek to match their goals with the products' functions. We further assume that interaction between user and product proceeds through a series of states, i.e., as hypothesized by Card et al. (1983). At each state, the user interrogates the system image for a correspondence between goal and function. Ideally, the goal and function would match and the action would be obvious. In the ergonomics literature, such matching is known as 'stimulus-response compatibility', e.g., turning a control knob clockwise causes a pointer to move to the right on a linear scale. People appear to have well developed stereotypes for some forms of stimulus-response compatibility and to base their actions on these stereotypes. Thus, at one level of behaviour users can simply match the system image with a stereotyped response. Such stereotyped responses can be thought of as *Global Prototypical Routines*. Errors can arise when users mistakenly match an object in the system image with their goal, or when a strong stereotype over-rides the correct action. For example, in work on ticket-vending machines (Baber and Stanton, 1996), a common error related to users attempting to insert their money as the first action (this 'error' is further supported by the prevalence of labels on these machines informing users 'Do not put your money in first'). It is proposed that this error is the result of a conflict in two Global Prototypical Routines. In one routine, users follow a 'Vending machine routine' (i.e., insert money; make selection; retrieve item), and in another routine users follow a 'Ticket-kiosk routine' (i.e., request ticket; pay money; collect ticket). It would appear that the designers had assumed that latter routine, when users often follow the former.

One rarely achieves the overall goal with a single action, and users need to keep track of their position in a sequence of goals (as proposed by GOMS of Card et al., 1983). Further, there may be occasions when there does not appear to be a clear match between goal and function. At this stage, users need to engage in problem-solving, and to infer the appropriate action on the basis of the system image. Rather than carrying a representation of the product throughout the interaction, users only require information when they are unsure of the appropriate action. The 'routine' represents a set of actions (or a single action) that is deemed appropriate for a given state. These can be thought of as *State-Specific Routines*. Interpretation of the system image in terms of the current goal state might draw on knowledge related to other products, i.e., through analogy or metaphor, in order to infer an appropriate action. Once the action has been performed, then the knowledge is no longer required. In this way, the routines are 'rewritable' in that they can be overwritten by subsequent information. Thus, users might invoke one or two pieces of information from long-term memory (using metaphor or analogy) in order to determine an appropriate action. However, in order to minimize working memory load, the users will rarely need to maintain this information throughout the interaction, and will concentrate on monitoring their progress towards the goal.

It is assumed that the majority of routines that one uses will be local. This is because movement through states in human-machine interaction is punctuated by brief periods in which the machine responds to user actions. From this perspective, users will employ multiple routines to determine

relevant action as and when required. This suggests that users neither need nor use mental models of the product being used. Rather, users seek to draw on previous experience to infer actions only when necessary. This further suggests that basing a design on a single metaphor (or even on a collection of metaphors) need not be useful; users will have different experiences and so need not appreciate the relevance of the supplied metaphor. Consequently, a general design proposal is that the 'exits' from each state in a transaction need to be clearly marked, need to be clearly related to potential user goals and need to be kept to a minimum (so as not to overload or challenge problem-solving abilities). In this way, the description would seek to consider knowledge-in-the-world, knowledge-in-the-user's-head, and knowledge-in-the-context.

While the ideas presented in this section are by no means radical, they have led us to propose that there is a need to represent the interaction between user and product in a manner that makes it easy to consider these ideas during initial design activity. Furthermore, we want our approach to produce quantitative data, i.e., time and error that will support early evaluation of products in terms of 'user performance'.

Task Analysis for Error Identification

The basis of Task Analysis for Error Identification (TAFEI) is the assumption that user-product interaction proceeds through a goal-oriented series of *states*, i.e., that each user action modifies that state of the product until the user has reached a specific goal. This means that the interaction can be easily represented in terms of a simple finite-state machine. However, it is important to note that the progression from state to state is dependent on the user's goal. This reduces problems of combinatorial explosion that are often associated with finite-state descriptions. This will become clear in the worked example below. At each state, the user needs to select an appropriate action in order to progress towards the goal. However, it is assumed that more than one action will be possible in each state. Thus, selection will depend upon routines (later developments in the method discussion in the Discussion section show how these notions are being incorporated into TAFEI). It is assumed that there are a small number of 'legal' actions, i.e., actions that will progress the user to the goal (typically in the region of 1 or 2) and that all other possible actions are 'illegal', in the sense that they will not take the user to the goal.

Procedurally, TAFEI is comprised of three main stages. Firstly, we describe the human side of the interaction; we tend to employ an Hierarchical Task Analysis (HTA), but one could employ any technique to describe human activity. However, HTA suits our purposes for the following reasons: i. it is related to specific tasks; ii. it is directed at a specific goal; iii. it allows consideration of task sequences (through 'plans'). Secondly, State-Space Diagrams (SSDs) are constructed to represent the behaviour of the artifact. Plans from the HTA are mapped onto the SSD to form the TAFEI diagram. Finally, a transition matrix is devised to display state transitions during device use. TAFEI aims to assist the design of artifacts by illustrating when a state transition is possible but undesirable (i.e., illegal). Making all illegal transitions impossible should facilitate the cooperative endeavour of device use.

The first step in a TAFEI analysis is to obtain an appropriate HTA for the device. As TAFEI is best applied to scenario analyses, it is wise to consider just one specific goal, as described by the HTA (e.g., a specific, closed-loop task of interest) rather than the whole design. Once this goal has been selected, the analysis proceeds to constructing State-Space Diagrams (SSDs) for device operation.

A SSD essentially consists of a series of states that the device passes from a starting state to the goal state. For each series of states, there will be a current state, and a set of possible exits to other states. At a basic level, the current state might be “off”, with the exit condition “switch on” taking the device to the state “on”. Thus, when the device is “off” it is ‘waiting for...’ an action (or set of actions) that will take it to the state “on”. It is important to have, on completing the SSD, an exhaustive set of states for the device under analysis *in terms of the specific goal being considered*. Numbered plans from the HTA are then mapped onto the SSD, indicating which human actions take the device from one state to another. Thus the plans are mapped onto the state transitions (if a transition is activated by the machine, this is also indicated on the SSD, using the letter ‘M’ on the TAFEI diagram). This results in a TAFEI diagram.

As mentioned above, in each state there will be a set of actions available to the user; we describe these as actions that the product is ‘waiting for’. Thus, when you insert your card into an automatic-teller machine (‘cashpoint’), the machine will be ‘waiting for’ <enter PIN>. When the PIN is correctly entered, the machine will be ‘waiting for’ <select cash, select cash + receipt, select balance, etc.>; selecting any one of these options will move the product to a new state. If the user selects an option that does not fit with the overall goal (e.g., if the user selects <balance> when their goal is <cash>) then the action is ‘illegal’. In other words, illegal actions lead the user away from their goal. Clearly, the definition of an ‘illegal’ action is dependent on the users’ goals, rather than on any characteristic of the product. The TAFEI approach seeks to identify illegal actions in pursuit of specific goals. Having identified illegal actions, the next step is to propose ways of redesigning the product in order to minimise such actions.

The most important part of the analysis (from the point of view of improving usability) is the transition matrix. All possible states are entered as headers on a matrix. The cells represent state transitions (e.g., the cell at row 1, column 2 represents the transition between state 1 and state 2), and are then filled in one of three ways. If a transition is deemed impossible (i.e., you cannot go from state X to state Y) “-” is entered into the cell. If a transition is deemed both possible and desirable (i.e., it progresses the user towards the goal state - a correct action), this is a legal transition and “L” is entered into the cell. If, however, a transition is both possible but undesirable (a deviation from the intended path - an error), this is termed illegal and the cell is filled with an “I”. The idea behind TAFEI is that usability may be improved by making all illegal transitions (errors) impossible, thereby limiting the user to only performing desirable actions. It is up to the analyst to conceive of design solutions to achieve this.

Examples of applications of TAFEI include prediction of errors in using kettles (Baber and Stanton, 1994; Stanton and Baber, 1998), comparison of word processing packages (Stanton and Baber, 1996; Baber and Stanton, 1999), withdrawing cash from automatic teller machines (Burford, 1993), medical applications (Baber and Stanton, 1999; Yamaoka and Baber, 2000), recording on tape-to-tape machines (Baber and Stanton, 1994), programming a menu on cookers (Crawford, Taylor and Po, 2000), programming video-cassette recorders (Baber and Stanton, 1994; Stanton and Baber, 1998), operating radio-cassette machines (Stanton and Young, 1999), recalling a phone number on mobile phones (Baber and Stanton, 2001), buying a rail ticket on the ticket machines on the London Underground (Baber and Stanton, 1996), and operating high-voltage switchgear in substations (Glendon and McKenna, 1994).

All of these examples of applying of TAFEI share common features, which define the operational parameters of the technique. First, they are all applied to purposeful use of the device, drawn from the goals of a tasks analysis. Second, each of the devices offers a clear and logical sequence of activity. The tasks are discrete, step-by-step, rather than continuous and concurrent. Third, there are clear system boundaries between the device and human elements. Given these

requirements, it is no wonder that most of the analyses have tended toward single user, single device systems. Theoretically it should be possible to take a nested systems approach, to analyse systems of greater complexity by addressing different levels and different scenarios. Some movement in this direction has begun with the analysis of operating high-voltage switchgear in substations (Glendon and McKenna, 1994). Before this is taken any further, it is sensible to assess the performance of the TAFEI technique on relatively simple systems first.

Worked Example

In this section, a worked example will be presented to illustrate the method. The product selected for this example is a digital watch. While the number of states that the watch can enter is limited, the product is sufficient to explain how the method is applied.

We begin by considering the system image of the product. The system image presents the controls, display, labels and instructions that the product presents to the user. Figure two presents a simple, labeled schematic of the digital watch. It is often useful to begin the analysis with a simple sketch to help focus on the key features of the product. This step can be applied even if the product is at the concept stage, i.e., in the absence of a working product.

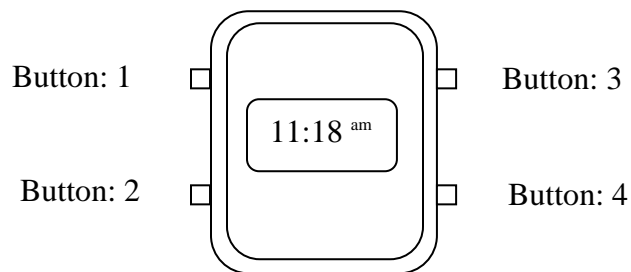


Figure two: Simple, labeled Schematic of Product

The next stage is to determine specific user goals. Definition of a goal is essential to both focus the analysis and also to constrain the number of states that need to be considered, and the path through the states. The user goal also defines the concept of illegal action. In this example, two user goals will be presented. In general, we would advise on using a set of between 2 to 5 user goals that are representative of typical product use, and which reflect different functions.

USER_GOAL₁: Change date to March 1st

USER_GOAL₂: Start stopwatch

Each user goal can be decomposed in two ways: (i.) in terms of the states through which the product passes in order to realize the goal; (ii.) in terms of the user actions required to achieve the goal. It is important to recognise these as two separate processes, rather than to attempt to combine them into one process. In this paper, we will focus on just one of the goals.

State-space Diagram

Figure three shows the states that our watch will pass through in order to support the actions required to change the date. By way of explanation, the watch needs to be in the Mode selection state (1) before one can change any setting. The watch allows one parameter to be changed, e.g., month or day or hour or minute. When a change has been made, the watch returns to the Mode Selection state (1) and then can either accept a new setting or can return to the initial state (0).

So, to change the 'date', the watch's states will be: 0 (initial) > 1 (mode selection) > 2 (change month) > 1 (mode selection) > 2 (change day) > 1 (mode selection) > 0 (initial)

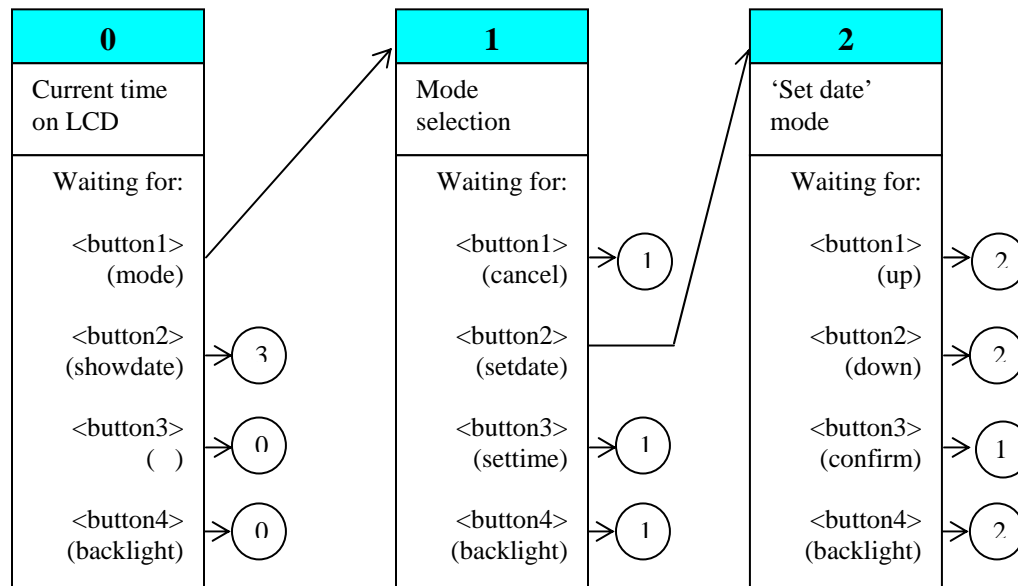


Figure Three: State-space diagram for USER_GOAL₁: Change date to March 1st

From figure three, it is apparent that the watch exhibits non-linear task sequencing; in other words, many of the user actions will result in the watch either remaining in its current state, or the watch returning to a previous state. One consequence of such operation, we feel, is that the user does not have a sense of a task progressing towards a goal, but rather feels that actions are random and poorly structured. Note also, that the definition of each of the buttons (with the exception of 4) change with each state of the watch. Such inconsistency might be misleading and confusing to users, particularly in the absence of clear labeling or the instruction manual.

The numbering of the SSD is specific to the goal being described; in other words, if a different sequence of states, in pursuit of a different goal, is constructed, then the state <Mode Selection> might be assigned a different number. Note also that the numbering of the 'waiting for...' states first uses the numbers in the diagram, and then increments numbers of 'undrawn' states. Thus, the last state in figure three is numbered 2, but the first 'waiting for...' state for state0 is 'press button 2<showdate>', which is not part of the task sequence for this goal and is assigned to be state3.

Notice also that the 'waiting for...' states all involve the use of four buttons, but that the function of the buttons changes according to the mode of the watch. The SSD indicates the resulting function for each button, dependent on the watch's current states.

User Actions

Having established the manner in which the watch will change states as the user pursues a specific goal, the next stage is to consider the actions that the user will perform. One could construct a Hierarchical Task Analysis (HTA) for this process, or one could simply reflect on the sequence in which buttons are being pressed. We prefer to use the HTA notation for two reasons:

(i.) it is an agreed standard form of analysis, at least amongst ergonomists; (ii.) it allows consideration of factors outside the simple button-pressing sequence. Figure three shows an HTA for the goal of changing the date.

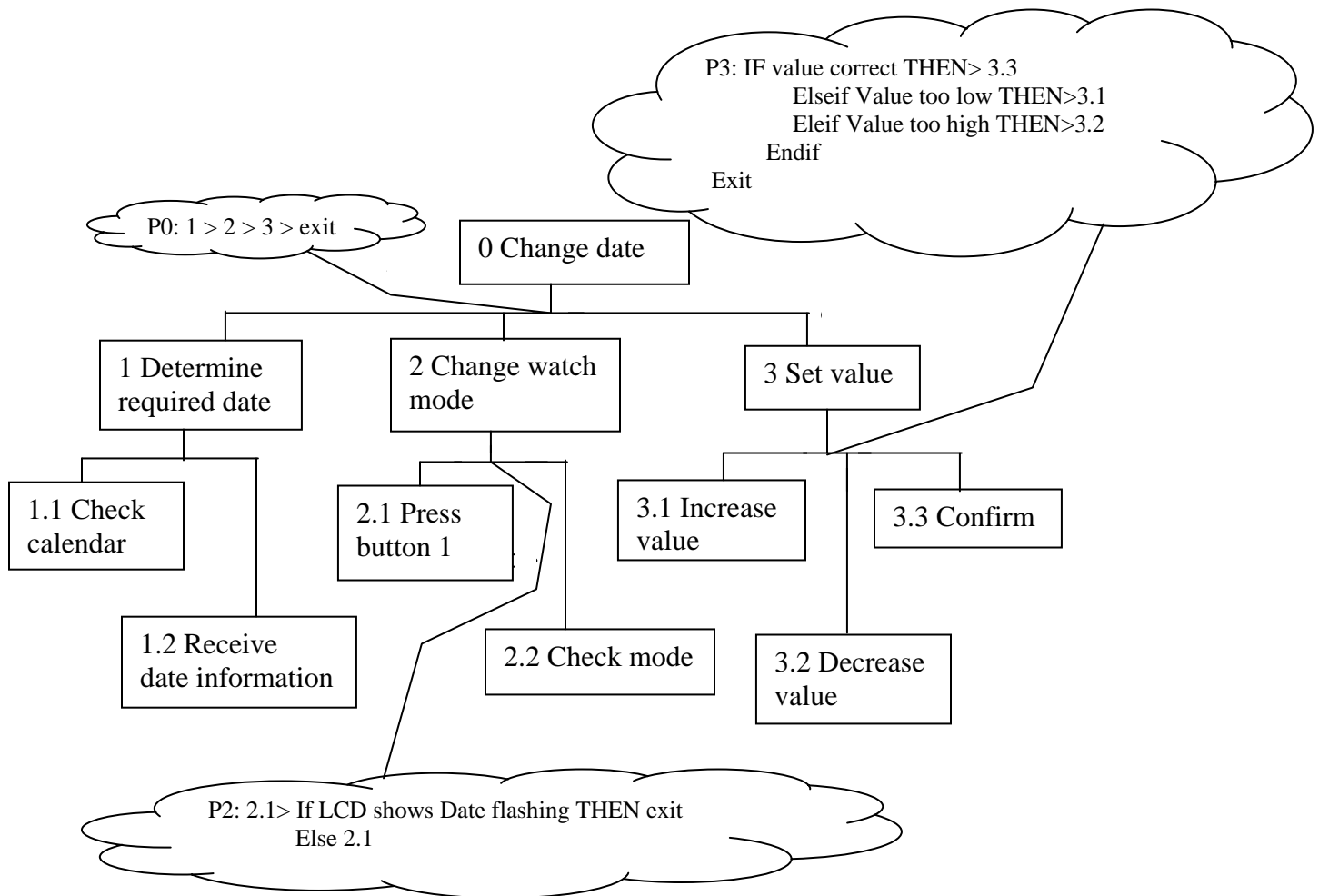


Figure 3: HTA for USER_GOAL₁: Change date to March 1st

TAFEI diagram

The plans and actions on the HTA can now be added to figure two. The numbers for plans or actions are simply written on the errors that correspond to the appropriate user activity. Thus, the arrow from State0 to State1 can be labeled “p2”, and the arrow from State1 to State2 can be labeled “p3”.

In this manner, we have a simple representation linking user actions to product states. Labelling the transitions in terms of the HTA provides a convenient means of indexing the two diagrams. Furthermore, at this stage, it is possible to use TAFEI for creating scenarios. For instance, the design team could consider what user action would lead the user to fail to perform ‘p2’, e.g., by pressing any of the other buttons. The answer might simply be a slip, i.e., the user intending to press button 1 but inadvertently pressing one of the other buttons, or it might be a mistake, i.e.,

the user intending to press one of the other buttons due to using an inappropriate routine or misinterpreting the system image.

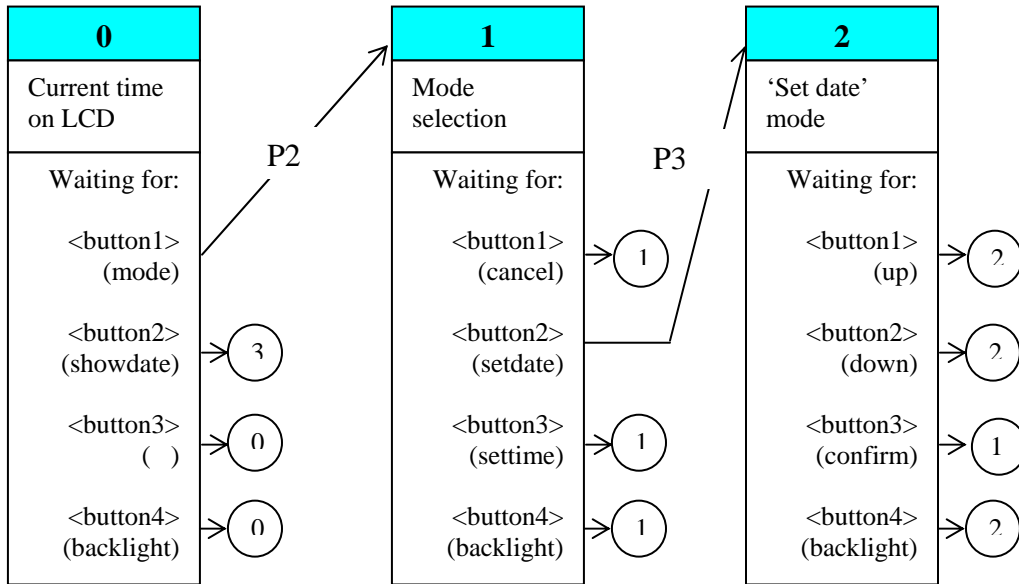


Figure four: TAFEI diagram

Transition Matrix

The identification of user error is initially performed using a transition matrix. We begin by asking whether it is possible to move from a given state to another state. It is important to realize that we are not considering *why* a user might make such a transition, only whether there is anything in the design to *prevent* such a transition. Given that all of the buttons on our watch can be pressed at all times, then the transition matrix for this product will be as shown in figure four.

From states	To States			
	0	1	2	3
0	I	L	I	I
1	L	I	L	I
2	I	L	L	I
3	I	I	I	I

Figure Five: Transition matrix for USER_GOAL₁: Change date to March 1st

In figure five, moving from state0 to state1 is the only course of action that will result in the achievement of the goal. However, the design of the watch allows the user to press any other button, i.e., effectively to move to one of the other states, and these actions will not lead to the goal; hence, they are illegal.

From figure five, it is clear that (even in so simple a product) there is potential for user error, with some eleven illegal actions (compared with five legal actions). The issue for the design team is

how to redesign the watch in order to reduce the number of illegal actions. While this chapter is not the place to debate ways in which the design team might approach this task, the reader can imagine discussions involving changing of labeling, clear indication of mode, highlighting of 'available' keys etc.

Validation

In order to demonstrate that the technique is usable, we have conducted a series of experiments into the reliability and validity of TAFEI. Empirical evidence of a method worth should be one of the first requirements for acceptance of the approach by the ergonomics and human factors community. Stanton and Stevenage (1998) suggest that ergonomics should adopt similar criteria to the standards set by the psychometric community, i.e. research evidence of reliability and validity before the method is widely used. It may be that the ergonomics community is largely unaware of the lack of data (Stanton and Young, 1996) or assumes that the methods provide their own validity (Stanton and Young, 1999).

In an earlier study, Baber and Stanton (1996) aimed to provide a more rigorous test of the predictive validity of TAFEI. Predictive validity was tested by comparing the errors identified by an expert analyst with those observed during 300 transactions with a ticket machine on the London Underground. Baber and Stanton (1996) suggest that TAFEI provides an acceptable level of sensitivity based on the data from two expert analysts ($r = 0.8$). A major claim for heuristics approaches is their speed of use. Study one suggests that, given equal, time an heuristic approach is less productive than the structured HEI approach of TAFEI. HEI asks analysts to consider human activity (rather than device characteristics). This means that we are asking our assessors to consider how potential users might experience problems with the device (rather than focusing on device specifics).

Participants also showed performance improvement over the first two experiments, increasing the hits and reducing the misses without compromising the false alarms. Both reliability and validity achieved acceptable levels, and the data compare well with previous studies (Stanton and Stevenage, 1998). The signal detection paradigm is a useful way of coding error prediction data, and the present study reinforces this approach. One might argue that a 'hit' rate of 5.4 out of 11 (49%) is relatively poor, but as argued above, any technique that can predict human error reliably can prove useful in product design and evaluation. Furthermore, Baber and Stanton (1996) have shown that expert performance with TAFEI reaches 90%.

Conclusions

TAFEI has been designed as a method with a specific focus. The aim is to support designers in developing products with minimal potential for user error. It is proposed that, by explicitly, exploring potential for error in the early stages of design, one can reduce such problems in later stages and that user trials can focus on other aspects of use. The method pairs a simple description of device operation (using state-space diagrams) with a description of user activity, in order to propose likely illegal operations.

We feel that TAFEI is of benefit for two reasons: i. It forces that analyst to consider human activity, and to consider problems in the light of this activity; ii. it forces the focus of attention away from device features. This means that it is possible to suggest radical revisions to a design (in order to reduce predicted errors), rather than seeking to modify specific features. Finally, TAFEI does not need real users performing real tasks with real products; thus, the techniques are

applicable to very early stages of design (see Baber and Stanton, 1999 for a discussion of how this approach can be applied as a design tool).

Currently, the method is being extended to provide quantitative predictions of user performance (Baber and Stanton, 2001). Thus, each user action can be assigned a time, e.g., using the times proposed by Card et al. (1983). This means that one can suggest transaction times for interacting with the product. Such an approach provides similar data to keystroke-level models, discussed above. However, assuming that each state offers more than one exit, i.e., more than 'waiting for...' component, there is the possibility that interactions can branch between states. Following this observation, we have been using MicroSAINT to model user interactions. In MicroSAINT, transitions between states are described in terms of time and probability. Thus, for figure four, we might assume a time of 200ms to press button1 and a probability of 0.8 to press button1 (with 0.1 for button2 and 0.05 each for buttons3 and 4). Obviously a full set of probabilities are required prior to developing the method further, and we are currently working with the HEART method proposed by Williams (1982). The point is that, from figure 4, we can imagine a situation in which a user erroneously presses button 3, is taken back to state 0 and then needs to recognise the mistake before pressing button 1. In this manner, it is possible to develop quite sophisticated models of user activity. Of course, such models do not explain why the user would do this, which is why we are also developing the rewritable routines theory.

While the notion of Global Prototypical and State specific routines appears plausible, there is a need to both validate these concepts and also to propose an architecture for cognition that supports them. This forms part of an ongoing project. For instance, figure six shows the results of an initial study into people's response to the system images of a compact disc (CD) player. In the study, nine people were given blank templates of a CD player. They were also given a set of ten functions. Their task was to indicate which of the rectangles on the system images they thought was associated with the functions. If response to a blank system image was solely guesswork, then the patterns of response would be random, i.e., we would not anticipate any clear majority responses. If, on the other hand, people had Global Prototypical views of how products were laid out, then one might anticipate quite strong majority responses.

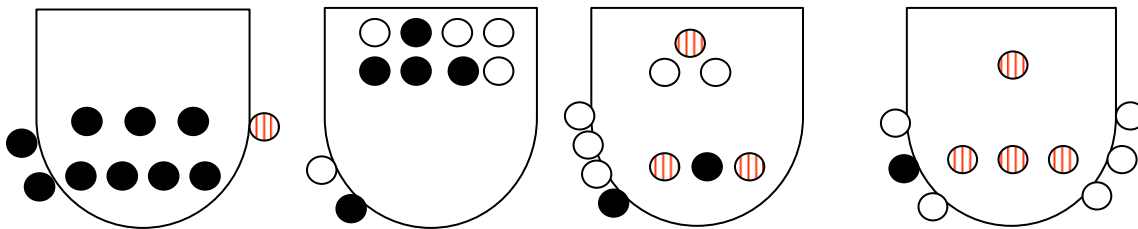


Figure six: Responses to CD player task

In figure six, the filled (black) circles indicate majority responses, i.e., 7 or more people agreeing. The shaded circles indicate 5-7 people agreeing, and the blank circles indicate <5 people agreeing. The suggestion is that some system images (i.e., the two on the left) evoke stronger Global Prototypical Routines than others. Consequently, when faced with a new product, people will look for specific controls in the area that they expect to find the controls. A point to note is that one black circle appears on all of the layouts, and this corresponds to respondents' expectations for the volume control. This study has been repeated with other participants and for other types of product, e.g., mobile telephone and personal digital assistant, with similar patterns

of results. This suggests that people do hold expectations about product layout, that (as with notions of display-control compatibility, e.g., Sanders and McCormick, 1992) people will anticipate that certain features will be in certain locations, and that certain functions will be operated from certain locations. Given people's tendency to behave 'as if' they were experts, then many of the slips arising from interaction with products simply occur because the user is expecting to operate a control that is not in its 'expected' place. TAFEI provides a method that allows designers to explore design concepts from the perspective of a theory of human-product interaction.

References

- Baber, C. and Stanton, N.A., 1994, Task analysis for error identification, *Ergonomics* 37 1923-1942.
- Baber, C. and Stanton, N.A., 1996, Human error identification techniques applied to public technology: predictions compared with observed use, *Applied Ergonomics*, 27 119-131.
- Baber, C. and Stanton, N. (2001) Analytical prototyping of personal technologies: using predictions of time and error to evaluate user interfaces, In M. Hirose (ed) *Interact'01* Amsterdam: IOS Press, 585-592.
- Baber, C. and Stanton, N. (1999) Analytical prototyping, In J.M. Noyes and M. Cook (eds.) *Interface Technology*, Baldock: Research Studies Press. 175-194.
- Card, S.K., Moran, T.P. and Newell, A., 1983, *The Psychology of Human-Computer Interaction*, Hillsdale, NJ: LEA.
- Craik, K.J.W., 1947, The theory of the human operator in control systems, part I, *British Journal of Psychology*, 38, 56-61.
- Crawford, J.O., Taylor, C. and Po, N.L.W., 2001, A case study of on-screen prototypes and usability evaluation of electronic timers and food menu systems, *International Journal of Human Computer Studies*, 13, 187-202.
- Glendon, A. I. & McKenna, E. F. (1994) *Human Safety and Risk Management*. London: Chapman & Hall.
- Johnson-Laird, P.N., 1983, *Mental Models*, Cambridge: Cambridge University Press
- Newell, A. and Simon, H.A., 1972, *Human Information Processing*, Englewood Cliffs, NJ: Prentice Hall.
- O'Malley, C. and Draper, S., 1992, Representation and interaction: are mental models all in the mind?, In Y. Rogers, A. Rutherford and P.A. Bibby (eds), *Models in the Mind: theory, perspective and application*, London: Academic Press, 73-91.
- Sanders, M.S. and McCormick, E.J., 1992, *Human Factors in Engineering and Design*, New York: McGraw-Hill.
- Stanton, N.A. and Baber, C., 1998, A systems' analysis of consumer products, In N.A. Stanton (ed.) *Human Factors in Consumer Products*, London: Taylor and Francis, 75-90.
- Stanton, N.A. and Stevenage, S.V., 1998, Learning to predict human error: issues of acceptance, reliability and validity, *Ergonomics*, 41, 1737-1756.
- Stanton, N.A. and Young, N.A., 1997, Ergonomics methods in consumer product design and evaluation, In N.A. Stanton (ed.) *Human Factors in Consumer Products*, London: Taylor and Francis, 21-53.
- Stanton, N.A. and Young, M.S., 1999, *A Guide to Methodology in Ergonomics*, London: Taylor and Francis.
- Williams, J.C., 1986, HEART – a proposed method for assessing and reducing human error, *Proceedings of the 9th Advances in Reliability Technology Symposium*, Bradford: University of Bradford.
- Yamaoka, T. and Baber, C., 2000, 3 point task analysis and human error estimation, *Proceedings of the Human Interface Symposium*, 395-398.