Task analysis for the investigation of human error in safety-critical software design: a convergent methods approach

N. M. Shivan*, S. J. Westerman, C. M. Crawshaw, G. R. J. Hocken and J. Sauer
Human Factors Group, Department of Psychology, University of Hull, Hull HU6 7RX, UK

Keywords: Safety critical systems; Task analysis; Error analysis; Human error.

An investigation was conducted into sources of error within a safety-critical software design task. A number of convergent methods of task- and error-analysis were systematically applied: hierarchical task analysis (HTA), error log audit, error observation, work sample and laboratory experiment. HTA, which provided the framework for the deployment of subsequent methods, revealed possible weaknesses in the areas of task automation and job organisation. Application of other methods within this more unscrewed context focused on the impact of task and job design issues. The use of a convergent methods approach draws attention to the benefits and shortcomings of individual analysis methods, and illustrates the advantages of combining techniques to analyse complex problems. The features that these techniques should possess are highlighted.

1. Introduction

1.1. Task analysis for the investigation of human error

The term ‘task analysis’ describes a plethora of techniques intended to describe and examine the tasks carried out by human beings within a system (for a comprehensive review see Kirwan and Ainsworth 1992). The range of human factors domains to which task analysis techniques can be applied is broad, including training, task and job design, allocation of function and performance assurance. Although they have the same general goal, different techniques may be suitable for answering different kinds of questions, in different kinds of work systems. This paper is concerned with an investigation into the human factors underlying the commission and detection of human error in railway signalling software design.

Within the context of performance assurance (i.e. the consideration of factors necessary to ensure system performance within acceptable tolerances), human error is of paramount importance, especially in safety-critical systems. Many analysis methods can be used to investigate the role of human fallibility in systems (e.g. SHERPA, human HAZOP; see Kirwan 1992a). Used as Human Error Identification (HEI) techniques, they are often applied within the framework of Probabilistic Risk Assessment (PRA), where the set of undesirable events that could occur within a system is defined, along with the paths that lead to them and the probability of their occurrence. The assessment and use of Human Error Probabilities (HEPs) has been criticized when applying absolute error probabilities (Hollnagel 1993). The authors

* Author for correspondence.
would argue that HEFs are more reliable when computing relative error probabilities probabilities.

1.2. Human reliability analysis and the nature of the system

For some computer system operations and similar environments, where many of these techniques were developed, HEF methods have undoubtedly proved to be useful, delimited, or `bound`, by the physical properties of system elements and the actual or action, a closing a valve will only be able to affect aspects of the system that the explosion, the sphere of influence of event is in principle determinable.

The constraint do not apply when considering the design of software systems, the `bound', of physical principles in the way described above, but the software programming language is used. The development of computer systems requires the use of software faults to system hazards.

1.3. Human error identification methods

The effectiveness of most HEF techniques depends upon the expertise and experience methods at all. This is because, almost without exception, these methods are based upon the judgements of expert practitioners, who are themselves documentation, the information gathered regarding tasks and possible errors with practitioners bias by using expert-system-like computer programs and question incomplete mental model of the system is not addressed. This is of particular importance because in computer programming, where there is unlikely to be outlined in § 1.2.

The development of a programming language, e.g. C++, Ada, is a good

1.4. Convergent methods approach

It can be seen that features of the computer software design task pose problems for the investigation of human error. In the main, HEF techniques are powerful and flexible methods that can be used effectively in a range of task environments. It is concluded, however, that in the study of error in computer programming, and certainly in the context of safety-critical systems, they should not be used alone. It is recommended that reliance is not placed on a single technique of HEF (Kwiat 1992c: 378). A broader-based approach is needed, using more than one method in order to provide convergent validation, and to allow different parts of system performance to be adequately investigated. This calls for the use of a set of different analysis methods and data sources, including actual task performance, rather than traditional `expert-opinion' focused task analysis alone. Using this combination a `matrix' of evidence regarding overall system integrity can be built up.

This paper considers such an approach to the analysis of human error. Specifically, the design of software for a safety-critical railway signalling control system, called `Solid State Interlocking' (SSI), is described in § 2. Section 3 details the initial task analysis of this design process. Section 4 describes an empirical error analysis used to provide convergent evidence for the investigation. It should be noted that this study is an investigation into the factors affecting production and detection of error, not a PRA of the system.

2. Solid State Interlocking

As a case study of a complex safety-critical system, the design of data for a railway signalling safety system was investigated. `Solid State Interlocking' (SSI) automatically controls the signals, points, etc. in a section of railway, ensuring that only safe movements of trains are allowed to take place. Each SSI is typically hardware redundant (three identical central processors), but the `geographic data' that specifies the logic for the movement of trains is unique to each SSI, and the same version is loaded into all three processors. This means that the `data' must be correct, as any faults could allow collisions or data-losses.

Figure 1 shows a small section of a highly simplified signalling diagram. It represents a plan view of the railway layout. Track sections are shown labelled T1, T2, T3. etc. Signals are represented by three schematic lights and are labelled S1, S2, etc. Where tracks converge or diverge there are gaps, representing sets of points, labelled P1 and P2.

Below the diagram is a simplified section of data, showing the conditions that must be fulfilled before Route 2 (R2; from S1 to S7) can be set. This entails checking that the route is available (R2 A), e.g. not barred because of maintenance; that the points are in the correct position, or free to be moved to the correct position (P1 CR, P2 only) and that other, opposing routes are not already set, which is done by checking the two opposing sub-routes to ensure that they are free (U10-AB, U3-BC). If these checks are passed, then the route is set (R2 S); the individual sub-routes in
3. Task analysis

The technique of Hierarchical Task Analysis (HTA; Annett and Duncan 1987) was chosen for the initial investigation of the SSI design process. HTA has existed for so long that it has been so widely used that it could be described as "traditional", and the critical incident—interview, system documentation and operator observation—description process. Used with the data sources listed above, it would provide retrospective validation of the technique.

3.1 Hierarchical task analysis method

A 'process analysis', as suggested by Pino (1981), was conducted concurrently with process, function of equipment, jargon, and general task environment, so that a 'process analysis' sets out to describe the production framework for the HTA itself is provided. HTA is used to describe the goals that the superordinate, goal of the SSI design system is broken down into a number of them. This procedure is then iterated, each sub-goal being seen as a goal or, in practice, task design, performance demanded, office environment informed the assessment of error-processes of the operation or plan, and hence the level of redesign required.
be downloaded to an SSI processor module which acts as a simulator for the
functional testing of the data. The tester performs functional tests of the SSI data by
using a trackball to scroll across a graphical representation of the signalling layout
(similar to figure 1), and setting elements to their various states.
Essentially, the SSI data are written, and then briefly tested using a computer-
based simulation of the railway network. This first testing session is carried out
by the author of the data, to ensure that it meets certain minimum standards. A
printout of the data is then checked by another engineer who has had no part in
the writing process. Finally, the data are again loaded onto a simulator, where
they are subjected to more rigorous and formal testing by a specially qualified
engineer who must have had no part in either of the preceding stages. If any
errors are found during either the check or formal test, they are logged and the
data returned to their author for amendments, and the whole cycle gone through
once more.
Regarding the staffing for the various stages, expertise tends to increase from
left to right. The most expert and qualified signalling engineers are employed at
the testing stage, the ‘last line of defence’ to remove faults in the data. The least
experience are employed in writing the data, although they can seek help from
more senior engineers who are not involved on the project. Expertise is gained in
the first instance by a number of 2-week training courses that instruct the newly
recruited trainee-engineer in basic railway signalling and SSI data. Most training
complicated aspects of the data preparation task, and then going on to qualify
for checking, etc. There is not the space here to review all of the potential
problems highlighted by the HTA, but some of the more interesting ones are
described below.

If the HTA is carried out systematically, structural features of the HTA
hierarchical diagram can be used to highlight elements of the task being studied. Per
instance, the pattern of the overall diagram (not shown) and the specific goals within
this pattern revealed the similarity of the writing and checking tasks compared to the
testing task.
Consideration of an organizational change revealed a potentially serious problem
relating to the introduction of automation in the data writing task. To increase
productivity, all of the signalling firms taking part in this study were in the process of
developing computer-based tools to automatically write much of the simpler, rule-
linked data straight from a signalling plan. This will have the effect of removing much
of the training that novice data writers gain by tackling these tasks, and leave them
less equipped to handle the more complex, knowledge-based data that was identified
in interviews as the most problematic.
Time pressure occasionally forces some checking and testing to be carried out in
parallel, leading to a revised Plan 1 (figure 3). This means that the version control
for the data must be very tight, or unchecked data could be signed-off as safe by the
tester. Normally, each new version of the data is given a unique version number by
the data writer. This number records how many cycles of checking and testing the
data has gone through, but not whether the latest version was generated because
errors were found in a check or a test. If it was a test, has that version of data been
checked as being error-free before? The danger point is shown by the dashed
diagram in figure 3. If this decision is made incorrectly, unchecked data could be
released into service. This problem is exacerbated by the coexisting-out of the
checking or testing of these ‘rush’ jobs to other signalling firms, with an attendant
increase in the difficulty of version control.

Figure 3. Redraw plan 1, for when checking (1.4) and testing (1.3) are carried out in parallel.
3.4. The need for error analysis
The first phases of the analysis showed that HTA provides a useful framework for the
detection of potential problem areas in a task. As discussed above, structural
elements of the hierarchical diagram can be used to show similarities and differences
within tasks. However, these similarities do not necessarily equate to similarities in
performance in writing and checking may not be identical even when identical data,
i.e. data that are difficult to write may be easy to check and vice versa. What HTA
actually combines to produce error.

4. Error analysis
The HTA was used to identify the key stages of SSI data production: writing,
checking and testing. These were then analysed using a variety of empirical
evaluation of tasks that together make up the overall system need to be assessed.
These were chosen partly on the basis of availability, but also to give a
broad range in terms of the type of data that they would provide. Existing error logs
and laboratory experiments would be used to investigate in detail issues brought
to further details see Westerman et al. (1995a).)

4.1. Error log audit
4.1.1. Method
The logs used to communicate faults found in formal checking and
testing to the data writer were the data source for this method. They revealed errors
made not only by the data writer (boxes 1 and 1.3 in figure 2), but also by the data
checker (box 1.4 in figure 2). As faults found by the tester necessarily have been
missed by the checker. These logs detailed the SSI data faults in terms of their
functionality and their underlying psychological cause, but still
provided rich information about the nature of errors. The main strength of this
method was its intrinsic validity: the errors were all committed and detected by
signalling engineers carrying out their normal job of work.

The primary weakness of the method is the uncontrollable nature of the variables
underlying the production and detection of errors. The number of faults in a
snapshot of the data depends mainly upon the skill of the data writer and the complexity of the work.
Usable measures of these factors were not available, however. In an attempt to
reduce bias, logs were included from 12 different signalling schemes, conducted by
seven different signalling firms at nine sites. Even when faults within a single scheme
were compared, so controlling for expertise and complexity, the detection of a fault
by the checker means that the fault is therefore unavailable for detection by the
tester.

Two classes of error were not recorded in the logs. The first class is that of the
errors committed by the data writer, but also detected by him or her. These would be
corrected when discovered, and not be logged and passed on to the later stages. The
second class is that containing faults that are not caught by either checking or
testing—perhaps the most important to study. Information regarding any faults

4.1.2. Results: Table 1 shows the breakdown of SSI faults detected by checking
and testing, 250 from each. The 'T/C ratio' specifies the ratio of faults detected by
checking versus checking within each category (i.e. testing divided by checking). The
higher the figure, the greater is the frequency of detection by testing compared to
checking. The nature of the information in the logs means that the fault categories
more closely relate to the railway signalling principle that they would violate. The
exceptions are the 'None' category, which relates to false-alarm errors by the checker
or tester (the data were actually correct); and the 'Other' category, which details
intrinsically SSI data-language specific errors, which do not relate to specific
signalling principles.

There are a number of reasons why it would be misleading to use these results to
make a formal quantitative comparison of the relative efficacy of the checking and
testing processes. First, the error logs were gathered from a number of different
schemes and therefore the checking and testing logs were not fully matched. Second,
any faults detected at the checking phase were not available for detection at the
testing phase, and consequently there is no means of estimating the efficiency of the
testing process in detecting these faults. Third, there were no measures available of
the total numbers of faults that escaped both checking and testing, again making
estimation of the efficacy of the testing phase problematic.

Given these reservations, a number of qualitative features are of note, however.
The T/C ratio shows an advantage of checking over testing for the 'Other' category.
The 'Opposing locking' category, which contains errors made in the setting of
conflicting routes, shows a bias towards testing. Also of note is the
propenseness of faults in the 'Identity', 'Other' and 'Route' categories (62.8 % of
time). Most of the SSI data faults that relate to these categories is similar to that
shown in figure 1. Overall, it is simpler and more straightforward than data
controlling other functions, and could be considered as requiring more skillful
and context-based performance and less knowledge-based performance (Kasmussen
1983) than data relating to 'Aspects' sequences.

A total of 12.4% of all faults logged turned out to be false alarms, while no
actual fault was present in the data. Several 'repeat' faults were found in the logs.
These refer to faults found at either the check or test that were still present when the

<table>
<thead>
<tr>
<th>Fault category: signalling principle</th>
<th>Checking</th>
<th>Testing</th>
<th>T/C ratio</th>
<th>Category (of total)</th>
</tr>
</thead>
<tbody>
<tr>
<td>None (false alarm)</td>
<td>46</td>
<td>26</td>
<td>0.57</td>
<td>12.4</td>
</tr>
<tr>
<td>Identity and labelling errors</td>
<td>26</td>
<td>10</td>
<td>0.64</td>
<td>9.7</td>
</tr>
<tr>
<td>Route setting</td>
<td>74</td>
<td>120</td>
<td>0.62</td>
<td>33.4</td>
</tr>
<tr>
<td>Signal aspect control</td>
<td>14</td>
<td>12</td>
<td>1.25</td>
<td>7.8</td>
</tr>
<tr>
<td>Approach locking</td>
<td>11</td>
<td>24</td>
<td>0.46</td>
<td>6.0</td>
</tr>
<tr>
<td>Opposing locking</td>
<td>4</td>
<td>39</td>
<td>0.10</td>
<td>5.3</td>
</tr>
<tr>
<td>Aspects sequence</td>
<td>4</td>
<td>12</td>
<td>0.33</td>
<td>2.8</td>
</tr>
<tr>
<td>Other</td>
<td>166</td>
<td>4</td>
<td>0.07</td>
<td>10.7</td>
</tr>
<tr>
<td>Total</td>
<td>290</td>
<td>290</td>
<td>1.00</td>
<td>100</td>
</tr>
</tbody>
</table>
SSI data were re-checked or re-tested, and testifies to difficulties in writing the data correctly. These faults applied to particularly novel, knowledge-based SSI data. Where information was available to show the number of faults logged at successive checking cycles, seemingly simple faults made in the 'Identity' and 'Other' categories were the only ones to escape detection until the fourth cycle. Similar information was not available for the testing stage.

4.2. Error observation

4.2.1. Method: Errors committed but subsequently detected by the same writer, checker or tester were not recorded in the error logs. To compensate for this, a second error observation was conducted. In addition to the task areas of the error logs, this method provided some insight into the fallibility of testers (box 1.5, figure 2). Different engineers were video recorded performing several hours of the three main task areas. Interference of the task could affect error-rate, so to maintain the tasks as little as possible the participants were asked to carry on their work as normal, but to comment if they found errors while they had, or had nearly made an error. They were then prompted as to the reasons they identified as causing the error, in an attempt to classify it as a skill- or rule-based slip or lapse, or a knowledge-based mistake (Rasmussen 1983; Reason 1990). The complexity of the task being undertaken, and the expertise and experience of the engineer were also used to inform the categorization of errors. While the possibility of taking a lack of errors existed, the video recording of the sessions and possibility of peer review by other engineers effectively minimized this.

4.2.2. Results: Table 2 shows the number of errors flagged by participants during the periods of task observation shown, and is divided up by task stage. The writing data does not equate to re-reading typing editing, but also includes reviewing, checking specifications and requirements, and planning. It can be seen that data checking was the most error-prone signalling principle in the data editing was 'Opposing locking', with errors. Of these, the most time consuming of all the errors was observed. Fully paper-based manual available for the task. In the testing stage, most errors related to based errors were observed in any stage. No errors at all were observed in 3 h of

| Task analysis for safety-critical software design |

| Table 2. Errors observed at the writing, checking and testing stages of SSI data production. |

<table>
<thead>
<tr>
<th>Error category</th>
<th>Writing (all including editing) (5.8 h)</th>
<th>Writing (editing only) (5 h)</th>
<th>Checking (3 h)</th>
<th>Testing (6.25 h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skill- and rule-based errors</td>
<td>31</td>
<td>25</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Knowledge-based errors</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Errors per hour</td>
<td>4.1</td>
<td>8.3</td>
<td>0.3</td>
<td>0.8</td>
</tr>
</tbody>
</table>

4.3. Work sample test

4.3.1. Method: Since the above two techniques are based on naturalistic analyses of work, they are susceptible to a lack of control over the initial SSI data. To compensate for this, a data writing work sample test was devised that would further inform the analysis of box 1.2 of the HTA (figure 2). A previously completed data set (for an SSI currently in service) had sections of data removed, and signaling engineers were recruited to complete the work under controlled, but work-based, conditions. This enabled task factors (e.g. complexity) to be studied. Fifteen data writers took part in the test. Although this is a small number for statistical purposes, it represents approximately one-third of all the suitable candidates working for organizations participating in the research project. The participants in the work sample test had a wide range of experience, from 6 months to 10 years, and were drawn from three different signalling offices. To ensure the representativeness of the work sample a highly experienced engineer was employed to select the SSI data, which included all of the main aspects of the writing task. The input of the researchers was used to ensure that both straightforward and novel SSI data elements were represented, so that rule-based and knowledge-based performance could be assessed to be utilized by the participants.

4.3.2. Results: Table 3 shows the completion times and faults for the different task types used in the test. For the rule-based task performance, faults in the completed data were scored per signalling function violated, as for the error log audit. This avoided the biasing of the results possible when one cognitive error resulted in many faults in the SSI data. For instance, one error made by a number of engineers, involved the unintentional omission of a whole signalling function. This single cognitive error resulted in the omission of several lines of data and tens of data 'words'. Assessing errors by the line or data 'word' would have given arbitrary scores dependent on how many lines or words made up a particular signalling function. This scoring system would have been inappropriate for the knowledge-based data as the small amount of code involved related to only one signalling function, which nobody got completely correct. Additionally, each data 'word' in the knowledge-based section contributed a particular aspect to the overall functionality of the feature, independently of the other 'words' (which was not generally the case for the rule-based data). The knowledge-based task element was thus scored by the data word, to give a usable index of 'correctness'.

| Table 3. Completion times and data faults in the work sample test. |

<table>
<thead>
<tr>
<th>Aspects of the task</th>
<th>Mean</th>
<th>SD</th>
<th>Time per line</th>
<th>Mean</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule-based performance</td>
<td>9518</td>
<td>3958</td>
<td>76.8</td>
<td>9.5</td>
<td>7.6</td>
</tr>
<tr>
<td>Knowledge-based performance</td>
<td>1120</td>
<td>458</td>
<td>200</td>
<td>3.7</td>
<td>1.5</td>
</tr>
<tr>
<td>Total performance</td>
<td>10102</td>
<td>2013</td>
<td>78.9</td>
<td>10.9</td>
<td>4.4</td>
</tr>
</tbody>
</table>
The differences between the two scoring systems makes formal statistical comparison between rule- and knowledge-based fault performance misleading. However, were the rule-based faults to be scored by the data available this would constitute a preponderance of rule-based over knowledge-based faults. On the other hand, it can be seen that knowledge-based data is much more time-consuming to complete, per also demographically by the fact that the two least experienced participants could not complete it. (Their data are thus not included, giving n = 13 for the knowledge-based elements.)

Regarding rule-based data, the problem of 'common-mode failure' was highlighted. A term usually used in system safety, it refers to situations in which seemingly independent components can fail due to the same underlying cause. Evidence for common-mode failure was found when it was seen that four of the specific faults were made by seven participants or by all of the rule-based category, encompassing 40 of the 142 errors. There was no pattern relating to test-related factors, and their interaction with human cognition, are thus implied. It was found that three of the four common-mode errors related to familiar data that, because of the specific instances used, required infrequently to be noticed. These errors can be characterized as false alarms, or 'strong but wrong' errors (Reason 1990).

4.4. Laboratory experiment
4.4.1. Method In real work, the detection of a fault by a checker renders this fault unavailable to a tester. Therefore, to investigate the relative efficacy of the completion by novice participants acting as either checkers (n = 13) or testers (16 faults per simulation), and performance of the novice checkers and testers chosen from actual logs in the error log audit. 'Route setting' was chosen because it contained frequent faults. 'Signal aspect control' and 'Opposing locking' as opposed to checking and testing. Two categories of fault were chosen, reflecting the two different methods by which this signalling principle is dealt with in the SSI data. It was thought that checking versus testing might give differential effects on the relative efficacy of could not be used because of the need to make the errors equally visible to task.

4.4.2. Results Table 4 shows the probability of detection of the four types of faults. The distribution of the four types of faults into the checking and testing task simulations. Analysis of variance revealed no main effect of task type (checking versus testing). There was a significant interaction between task type and fault type ($F(3,72) = 10.58$, p < 0.001). This suggests that 'Opposing locking' faults, end of testers in detecting 'Signal aspect control' faults.
A factor not considered in the above argument is fault detection. The error logs do not provide an accurate ratio of knowledge-based to rule-based faults pre- and post-checking and testing, which would allow the relative detectability of each to be known. However, there is some continuous data (i.e. from the same SSI dataset) relating to faults detected at each stage of checking. This showed that the error rates to pass through three cycles without detection were ones in simple, rule-based data. Again, this may not necessarily be because simpler faults are actually any harder to detect than the knowledge-based ones. Indeed, even if they are easier to detect, the number of simple faults escaping detection may be higher because of their higher prior probability in the data.

Another factor in the number of rule-based faults may be the relative lack of emphasis placed on their importance, as shown by the HTA interviews. This would lead to more attention being given to parts of the data seen as challenging or virtually any faults in the data do have disastrous consequences.

The HTA suggested that knowledge-based data writing was more complex and made more demands on the operator than simpler data, and this was supported by the work sample test. However, HTA has little provision to represent explicitly the event-rates that would have shown that knowledge-based data, although perhaps more error-prone, is written far less often than the simpler data, and so has correspondingly less influence on overall system integrity.

The prevalence of simple errors and their difficulty in detection has unanticipated consequences for the introduction of automation. Although the original reasons for the development of automation was economic, its introduction may have greater than expected benefits if it can eradicate a potentially dangerous source of data faults. However, its implications for training will still need to be addressed.

5.2. Common mode error

Common mode error was another problem area uncovered by the study. Checking, computer programming and testing can be seen as the most efficient method for revealing error in the data, and the principle reason for this is that the whole of the code can be inspected, as opposed to the limited amount of functionality that can usually be tested in complex systems. The error logs probably because a greater proportion of common errors is testable than for common programmes.

However, the use of continuous data recording has implications for the consequences of the check stage being missed out when carrying out checking and testing in parallel (figure 3). This is because some of the faults in the data, such as those that fall into the 'other' category, are less likely to be detected by testing alone. To help to ameliorate this problem a computer-based logging tool is currently under development by the research team, to assist with fault logging and data version control. It will also record some of the psychological error mechanisms that lead to the faults, and help to aid the identification of further error reduction techniques.

5.3. Task diversity

The converse of task similarity is task diversity. Differences between checking and testing, suggested by the HTA, were confirmed by qualitatively different error detection performance between checking and testing found in the error log audit. Although encouraging, the error log results may have been due to a number of factors other than the task environment (i.e. the initial number and type of faults in the data). However, the result was confirmed in the laboratory experiment, where the seeded faults were exactly the same for checkers and testers. The results showed that while there was no difference in the overall fault detection performance between the two methods, they did lead to the discovery of different types of fault. Almost inevitably, because of the differences between the two sets and the simulations, there were some differences of detail between the error-log and laboratory experiment results, e.g. in Signal aspect control faults. A number of factors may have contributed to this, e.g. the difference in part-task characteristics, the reduced range of faults in the simulation (16 versus 290 in the error log). Even these differences, however, the fact that both studies still showed different fault detection characteristics between checking and testing indicates that rather than the two stages being an example solely of redundancy in the system, they are instead an example of task diversity.

The use of task diversity can have positive implications for fault detection tasks (Pagett 1986). Different task representations are likely to engender different mental models in the operators, and lead to different emphasis and task performance strategies. This can render people less vulnerable to the threat of common mode error present when tasks are too similar, as shown by SSI data writing and checking. This diversity of knowledge, strategy and mental model of a task is known as 'cognitive diversity' (Westerman et al 1995). The diversity between checking and testing, and has implications for the consequences of the check stage being missed out when carrying out checking and testing in parallel (figure 3). This is because some of the faults in the data, such as those that fall into the 'Other' category, are less likely to be detected by testing alone. To help to ameliorate this problem a computer-based logging tool is currently under development by the research team, to assist with fault logging and data version control. It will also record some of the psychological error mechanisms that lead to the faults, and help to aid the identification of further error reduction techniques.

5.4. 'Opposing locking' faults

A consistent difference between checking and testing performance was found for 'Opposing locking' faults. It was apparent that checkers found these faults more difficult to detect than testers, so a number of further experiments were performed to ascertain why this might be the case (Shyane et al. 1986). 'Opposing locking' is dealt with in SSI data by the use of sub-route labels, which define a section of track and also the ends of this section that the train will enter and leave by. For example in

highly trained engineers to exactly the same errors of cognition when in the same task environment, Checkers will be less to those same errors because of the similar task factors to writing and self-detection of these 'habit intrusions' will not be good, as they are errors of intention, not action. Common-mode errors were also seen in the laboratory experiment, with task environment (checking or testing) effectively predicting which types of detection errors would be made.
5.5. Convergent methods

The methods used in this analysis all had a role to play in the investigation of human error. In the first instance, HTA provided an overview of the task not afforded by the other methods. Additionally, HTA does not have to be a study of the task as it, by itself, features allowed the discovery of the organizational issues regarding the introduction of automation. It is also useful for the consideration of sequencing and scheduling of tasks. Thus, this information was not brought to light by the other methods. HTA’s weakness was found to be in part due to the subjective nature of its data sources. However, the inability to show quantitative aspects of tasks, such as event and error rate, and how this affects the system is the biggest drawback of HTA when considering the study of error.

The error log audit and error observation are both essentially error sampling procedures of the task. Although the information that they produce can be noisy due to its work environment origin, they do provide quantitative data on events, and errors, however (e.g., self-detected errors), and so to compensate for this in the current study, error observation was used. The error observation seemed to be useful in recording errors in relation to overt actions, rather than to covert cognitive processes, as shown by the lack of knowledge-based errors while checking. 'There is some evidence... that while people are good at catching their own errors of action, they are much less good at catching their own errors of thinking, decision making, and perception' (Smedley and Moray 1991: 78).

Error observation was useful for pointing out interface and task support factors.

The errors in the data editing part of the writing task point to the error proneness of inputting data through a standard keyboard, which may be reduced by the use of visual programming environments and direct manipulation of code. However, this would then make the data writing task environment more similar to the testing task, so reducing overall system diversity. The system-wide effects of such factors need much further investigation. The inadequacy of existing manuals, the primary reliance on mail to get the help, and so on, was also observed. As a further part of this study, a computer-based help system is being developed. This will include the information contained in the existing paper-based manual, but with improved searching and cross-referencing, and the ability to annotate and personalize the manual to support individual working styles. Job support tools such as these may help to reduce the problem of less training, but again research is needed to see how this might apply.

The work sample test and laboratory experiments are both types of experiment. These are needed to study the variables identified as important by earlier stages.

6. Conclusions

From the evidence presented above, two dimensions of variation can be identified with respect to techniques used for the investigation of human error in this study. First, the techniques varied in their capacity to represent events—and therefore error—frequencies; HTA lacks the capacity of the other, more empirical, techniques in this respect. Second, the empirical techniques differ in the familiar trade-off between validity and control. Error logging and observation represent highly externally valid techniques. Laboratory experimentation represents the extreme of control and internal validity, with work sample tests offering characteristics between the two ends of the spectrum. Used here to investigate human error in safety critical systems, analysis of human error will in any work-based system benefit from the application of techniques that vary in these properties.

Acknowledgments

The work described here forms part of the project ‘Human Factors in the Design of Safety Critical Systems’ (Grant No. GR/J17319) from the UK Safety Critical Systems Research Programme supported by the Department of Trade and Industry and the Engineering and Physical Sciences Research Council. It was completed with the collaboration of Interactive Control Engineering, GEC Alsthom Signalling Ltd, Signalling Control UK Ltd and Westinghouse Signals Ltd.

References

Learning to predict human error: issues of acceptability, reliability and validity

NEVILLE A. STANTON* and SARAH V. STEVENS
Department of Psychology, University of Southampton, Highfield, Southampton SO17 1BJ, UK

Keywords: Human error; Methodology; Reliability; Validity; Sensitivity.

Human Error Identification (HEE) techniques have been used to predict human error in high risk environments for the past two decades. Despite the lack of supportive evidence for their efficacy, their popularity remains unimpaired. The application of these approaches is increasing, to include product assessment. The authors feel that it is necessary to prove that the predictions are both reliable and valid before the approaches can be recommended with any confidence. This paper provides evidence to suggest that human error identification techniques in general, and SHERPA in particular, may be acquired with relative ease and can provide reasonable error predictions.

1. Introduction

1.1. Introduction to human error

Human error is an emotive topic and psychologists have been investigating its origins and causes since the dawn of the disciplines (Reason, 1990). Traditional approaches suggested that errors were attributable to individuals. Indeed, so-called 'frequent slips' were treated as the unwitting revelations of intention: errors revealed what a person was really thinking but did not wish to disclose. More recently, cognitive psychologists have considered the issues of error classification and explanation (Senders and Moray, 1991). The taxonomic approaches of Norman (1988) and Reason (1980) have enabled the development and formal definition of several categories of human error (such as: capture errors, description errors, data driven errors, associated activation errors and loss of activation errors) while the work of Reason (1990) and Wickens (1992) attempts to understand the psychological mechanisms that underlie cause errors (such as failure of memory, poor perception, errors of decision making and problems of motor execution). Reason (1990, in particular), has argued that one needs to consider the activities of the individual if one is to be able to identify what may go wrong. Rather than viewing errors as unpredictable events, this approach regards them to be wholly predictable occurrences based upon an analysis of an individual's activities. Reason's definition proposes that errors are:

those occasions in which a planned sequence of mental or physical activities fail to achieve its intended outcomes, [and] when these failures cannot be attributed to the intervention of some chance agency. (Reason, 1990: 9)

* Author for correspondence.