A Model for the Human Computer Interface Evaluation in Safety Critical Computer Applications

Fabio A. Schreiber
Dipartimento di Elettronica e Informazione
Politecnico di Milano

Proceedings of the IEEE
International Conference and Workshop:
Engineering of Computer-Based Systems

March 1998, Jerusalem, Israel

A Model for the Human Computer Interface Evaluation in Safety Critical Computer Applications[i]

Fabio A. Schreiber
Dipartimento di Elettronica e Informazione - Politecnico di Milano
Piazza Leonardo da Vinci 32, 20133 Milano, Italy
e-mail: schreibe@elet.polimi.it

Abstract

An abstract quantitative model of a Human Computer Interface is introduced in order to evaluate the impact of the operator's errors on the total dependability figures in safety critical systems.

Usability, expressed as a function of the workload on the operator, carelessness, and the difficulty of learning, modeled as a function of the time needed to use the interface without making mistakes, are considered as the evaluation parameters of the model.

The proposal requires a strong experimental work to determine the exact form of the functions and the values for the numerical coefficients.

1. Introduction

The embedding of computers in control systems for safety critical applications requires an in depth evaluation of their dependability features. Figures established by standardization committees are orders of magnitude higher than what can be proved today for software, which is an important part of the system. However both certification authorities and insurance companies need quantitative tools for assessing the dependability of safety critical systems.

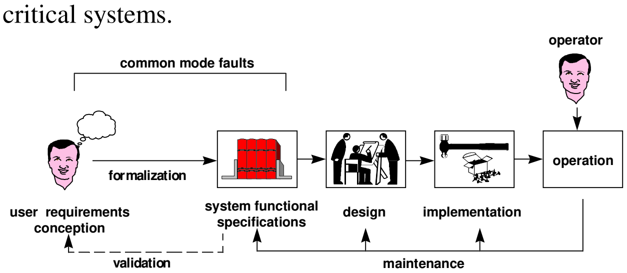

Fig. 1 - The design process

Aim of the research, this work is part of, is to understand if and how there is any chance of bridging this gap. Human beings enter in the picture since they are involved in at least two very important functions: at the very beginning (the user requirements conception) and at the end (the operator-system interface, often called Human-Computer Interface *HCI*) of the design process (Fig. 1); even if the entire design and implementation process of the system could be automatically verified and tested by formal methods, so obtaining the certainty of its correctness, the aforementioned functions would still be a source of human errors [MoSc97]. Errors in the requirements analysis phase, which are independent of the physical nature (hardware or software) of the components, are the most dangerous since they affect - as common mode faults - all the design process. However they are dealt with by means of inspections and reviews of the functional specifications and we do not consider them as a part of the design process in a proper sense. On the other hand, the operator's error directly affects the reliability of the system just like a software bug or a hardware failure.

While some work has been made on the psychological side of the conception and of the design process [Cock94, Curt88], most researchers are dealing with the ergonomy of HCI in order to ease the job of the operator and to minimize the probability of errors in his/her work. However ergonomic studies often end with a list of guidelines and recommendations, but very few of them give a quantitative evaluation of the operator's reliability [Dhil87, GeGr96, HuBl96]. The goal of other studies, based on very detailed models, is the discovery of causal links leading to the operator's errors, in order to avoid possible future accidents [CaHo93, LeTu93].

In this paper an abstract quantitative model of the HCI is proposed. Its use is twofold: first it guides experimental researchers in collecting data about the performance of the HCI, data which are used to fit the model to the experiments; then it can be used to evaluate the HCI dependability. Therefore, the goal of the model is to estimate the traditional parameters of the theory of reliability starting from measurable quantities and *without entering into the details of the system's control features*.

2. The model

The probability that an operator makes an error depends on two kinds of factors:

- subjective (or psychological) - errors depend on the operator's perception and skills;
- objective (or engineering) - errors depend on the engineering aspects of the design.

Therefore a model for evaluating a HCI should account for both of them [GeGr96]. In figure 2 a framework is shown in which both aspects are present:

- the definition of the operator's functions, their design and implementation represent the engineering side of the problem and the *usability* and *carelessness* are the evaluation parameters;
- the learning and training procedures depend both on the design of each function and on the skill of the single operator [Kont96] and the *difficulty of learning* is the evaluation parameter.

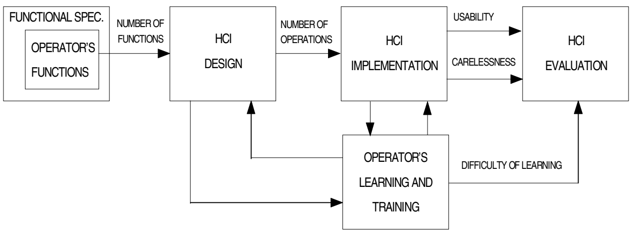

Fig. 2 - Evaluation framework of a HCI

A first point in designing a HCI is the partition of tasks between the operator and the machine; this entails a philosophical (but with very effective practical consequences) debate on how much of real *control* functions and how much of simple *monitoring* functions should a human being be assigned, the trade-off being fallibility vs. flexibility [Wien87]. In any case, in the functional specification phase, a *number of functions* (*F*) are identified as requirements for the HCI. In the design phase each function is exploded into a sequence (or into a set) of *operations* (or *events*) to be performed in order to accomplish it; the dynamics of the operator-system interaction is also to be considered in the modeling [CaCV93, CaHo93]. Due to the output of the design phase, errors can be made by the operator *in the sequence* of the operations, *in their timing*, in taking the *wrong decisions*. In the implementation phase the *technological* factors are considered; the design and implementation phases are interactively based on engineering as well as on ergonomic and learning requirements.

We can define now the three evaluation parameters of the model:

Usability (*U*): a HCI is "usable" when one can *easily learn* exactly *when* and *what* to do in order to control the system and this is *within the range of his/her skills*. However many authors [AmDe92, Kont96] relate the usability of a system to the workload the operator(s) must support to monitor/control it. Usability is also affected by the overhead introduced by the HCI with respect to a virtual direct chain of the type "I think it - I do it".

In the following definitions we separate the learning from the workload aspect and we do not take into account the technological aspects emerging from the implementation phase, but only focus on the manual/decisional issues; further work should include the technological aspects as well. Let us define:

e_{ri} the i-th monitoring (reading) event type
e_{ai} the i-th control (action) event type
n_{ri} the number of e_{ri}
n_{ai} the number of e_{ai}

required of the operator to perform the set of functions. Let also α_i and β_i be two sets of coefficients which express the weights of the different events in determining the workload and T a time interval; then we can define the usability of a HCI as in (1), where M represents the saturation workload, i.e. the maximum stress the operator can afford within safety limits, and the fraction represents the actual stress, i.e. the weighted "density" of events requiring the operator's attention.

$$U = M - \frac{\sum_i \alpha_i * n_{ri} + \sum_i \beta_i * n_{ai}}{T} \qquad (1)$$

Therefore a negative value for U would show an unbearable workload for the operator. This definition allows a very fine tuning of the usability parameters; more approximated models can use $\alpha_i=\alpha=cons.$ and $\beta_i=\beta=cons.$ giving a simpler expression (2) (with obvious meaning of n_r and n_a)

$$U = M - \frac{\alpha * n_r + \beta * n_a}{T} \qquad (2)$$

Even $\alpha=\beta=1$ could be used in a very rough model; however we can notice that for monitoring oriented designs $\alpha > \beta$, while for control oriented designs $\beta > \alpha$.

Carelessness (*C*): from the analysis of a number of accidents in flight as well as in industrial control systems, it is well known that lowering too much the workload on the operator can be a potential cause of errors owing to the possible inadequacy of his/her response time and of the quality of the decision in case of emergency [Kont96]. The inadequacy may arise from inadvertence, frustration, lack of skill or of training. Therefore it should be important to quantify the ability of the HCI of keeping the operator alert and of avoiding safety measures elusion (avoidance of default values, etc.). The technological choices can greatly influence this parameter as, for instance, in the case of multi mode displays and command consoles; in an extensive research on the perceived human factors problems of flightdeck automation [Funk96], 3% of the citations claimed that mode awareness may be lacking and this may lead to reduced situation awareness and errors, while 0.7% pointed out that pilots can inadvertently select the wrong automation mode.

It is very difficult to give an analytic expression for C; it should account both for the real workload and for the workload added to keep the operator's attention high. At the moment we can only conjecture *it correlates to the inverse of the workload* (3) and let the experimental researchers to find the actual form of g.

$$C = g\left(\frac{1}{W}\right) \tag{3}$$

Learning difficulty (L): let us suppose the harder is the job the longer it takes to learn how to do it correctly. We define $t(f_i)$ to be the *learning time* for function f_i (i.e. the time needed to learn how to perform f_i); then we can define L as in (4), that is *the learning difficulty of a HCI is represented by the average learning time of its functions.*

$$L = \frac{1}{F}\sum_{i=1}^{F} t(f_i) \tag{4}$$

However we must define what the learning time is. To this purpose, let τ_{ij} be the time from the beginning of the training session to the most recent error incurred by a trainee operator o_j on function f_i. We can say that o_j *learned* f_i *if he/she did not make further errors over a time period* $t_j(f_i)=\gamma*\tau_{ij}$. This definition closely looks like that of a transient phenomenon in a physical system, where usually γ has a value of 4 or 5; however in physical systems the transient is represented by a continuous function and τ represents its time constant (fig. 3a), while the typical learning curve is discontinuous. Therefore our definition of τ_{ij} is conservative and it does not pose any constraint on the form of the learning function, as it is shown in figure 3b.

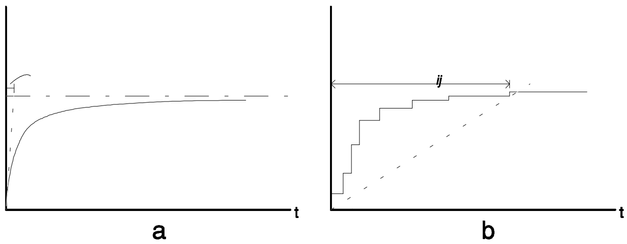

Fig. 3 - The learning function

Therefore the learning time L_i for function f_i can be taken as the average performance over a number O of operators (5)

$$L_i = t(f_i) = \frac{1}{O}\sum_{j=1}^{O} t_j(f_i) \tag{5}$$

The learning difficulty then becomes

$$L = \frac{1}{F}\sum_{i=1}^{F} L_i = \frac{1}{F*O}\sum_{i=1}^{F}\sum_{j=1}^{O} t_j(f_i) \tag{6}$$

We can also define a derived measure which, even if not directly included in the HCI model, can be useful in evaluating the operators performance. Let us call L_j the learning time of operator o_j averaged over all of the functions of the HCI, as in (7);

$$L_j = \frac{1}{F}\sum_{i=1}^{F} t_j(f_i) \tag{7}$$

then we can define the *skill* S_j of operator o_j the ratio of his/her learning time to the learning difficulty of the HCI.

$$S_j = \frac{L_j}{L} \tag{8}$$

Obviously this is a very rough averaged index, which could only be used in mass screening; a much finer analysis of operators skills can be found in [DiWa96].

The goal of this work is the evaluation of a Mean Time Between Errors (*MTBE*), homogeneous with the *MTBF* (Mean Time Between Failures) arising from the analysis of the other components (hardware and software) of the system, which could be used in reliability analysis methodologies such as FMEA (Failure Modes and Effects Analysis) or FT (Fault Tree analysis). FMEA is a "bottom-up" approach to the analysis of system reliability which requires the listing of all the system's subsistems and parts, their failure modes, and the failure rate of each mode; FT is a "top-down" approach which, leaving from an "undesirable event" taken as the root, builds a tree of all the possible events which could be the cause of the root event, connecting them by means of logic gates (AND, OR, …), and evaluating their occurrence probability [Dhil87].

At the moment we can only conjecture that *MTBE* is directly correlated to Usability and inversely correlated to the number of functions to be performed, to the Carelessness, and to the learning time (9); however a strong experimental effort is needed here too to fit an actual function to field data.

$$MTBE = h\left(U, \frac{1}{F}, \frac{1}{C}, \frac{1}{L}\right) \qquad (9)$$

3. Future work

The formulas introduced in the previous section give some hints on the way to define and measure some performance/reliability parameters of a HCI; however, up to now, they have no experimental evidence and a long way is still between this abstract model and its practical effects. A substantial amount of experimental work must be made before the model can be used in actual evaluation procedures:

- the saturation workload *M*, which may change from different application domains, must be the result of statistical analysis on historical data and of field measures;
- the two weighting coefficients α and β can be inferred by inspection of the complexity of each type of operation and of the consequences of its results. In particular they should account for the dynamics of the operator-system interaction;
- the coefficient γ and the parameters τ_{ij} must be experimentally measured; while γ should be mostly invariant with respect to the different applications, the τ_{ij} must be measured for each function of the HCI;
- finally the two functions *g* and *h* should be experimentally inferred from field tests.

Probably some preliminary experimental work can be based on simulation (e.g. the choice of α and β), however field test should be the main source of significant data. The reward for the expensive empirical activity, that in any case takes a large part of this kind of studies [AmDe92, HuBl96], should be the availability of meaningful numerical values, homogeneous with those coming from other classes of physical components of the system, obtained without "throwing the dices".

4. Conclusions

In the previous sections we discussed the four major components of *MTBE* (Mean Time Between Errors), a quantitative parameter for the evaluation of a HCI with the goal of estimating the probability that a human operator makes an error owing to stress, inadvertence or poor training. It results that Usability, expressed as a function of the workload on the operator, should be the main factor in determining the operator's reliability since, besides being the measurable stress cause, it also depends on the total number of functions performed by the HCI.

Therefore the quantitative knowledge of the different parameters of a HCI allows the evaluation of the overall reliability of the computer based system. Furthermore, the Usability and the Carelessness values can guide the designer in the choice of the amount of "decision power" to be given to the automated system with respect to that of the human operator. Finally, being able to evaluate the learning difficulty of each command and control function, the designer is helped in determining the bottlenecks in the operator's training activity and in optimizing the HCI under this aspect.

5. References

AmDe92 Amalberti R., Deblon F. - *Cognitive modeling of fighter aircraft process control: a step toward an intelligent on-board assistance system* - Int. J. Man-Machine Studies, Vol. 36, pp. 639-671, 1992

CaCV93 Cacciabue P.C., Carpignano A., Vivalda C. - *A dynamic reliability technique for error assessment in Man-machine systems* - Int. J. Man-Machine Studies, Vol. 38, pp. 403-428, 1993

CaHo93 Cacciabue P.C., Hollnagel E. - *Human models in reliability and safety analysis of interactive systems* - Proceedings PSA 93, Clearwater Beach (Fl.), 1993

Cock94 Cockram T. et Al. - *Human error in the software generation process* - in Technology and assessment of safety-critical systems, Redmill F., Anderson T. (eds.), Springer-Verlag, 1994

CuRo91 Cushman W.H., Rosenberg D.J. - *Human factors in product design* - Elsevier, 1991

Curt88 Curtis B. et Al. - *A field study of the software generation process for large systems* - CACM, Vol. 31, n. 11, pp. 1268-1287, 1988

Dhil87 Dhillon B.S. - *Human reliability in information systems* - in: Information systems failure analysis, Wise J.A., Debons A. (eds.), ASI Vol. 32, Springer-Verlag, 1987

DiWa96 Dillon A., Watson C. - *User analysis of HCI - the historical lessons from individual differences research* - Int. J. Human-Computer Studies, Vol. 45, pp. 619-637, 1996

Funk96 Funk K., Lyall B., Riley V. - *Perceived Human Factors Problems of Flightdeck Automation* - FAA Grant n°. 93-4-039, Phase 1 Final Report, http://www.engr.orst.edu/~funkk/Auto/autoprob.html

GeGr96 Gentner D.R., Grudin J. - *Design models for computer-human interfaces* - IEEE Computer, Vol. 29, n. 6, pp. 28-35, 1996

HuBl96 Hubona G.S., Blanton J.E. - *Evaluating system design features* - Int. J. Human-Computer Studies, Vol. 44, pp. 93-118, 1996

Kont96 Kontogiannis T. - *Stress and operator decision making in coping with emergencies* - Int. J. Human-Computer Studies, Vol. 45, pp. 75-104, 1996

LeTu93 Leveson N.G., Turner C.S. - An Investigation of Therac-25 accidents - IEEE Computer, Vol. 26, n. 7, pp. 18-41, 1993

MoSc97 Morzenti A., Schreiber F.A. - *Analysis of techniques in the assessment of ultra-reliable computerized systems* - ACM Software Engineering Notes, Vol. 22, n. 2, pp. 87-91,1997

Wien87 Wiener E.L. - *Fallible humans and vulnerable systems: lessons learned from aviation* - in: Information systems failure analysis, Wise J.A., Debons A. (eds.), ASI Vol. 32, Springer-Verlag, 1987

[i] Work partially supported by CESTIA-CNR