# Reliability Analysis of Systems Based on Software and Human Resources

Alberto Pasquini, Giuliano Pistolesi, and Antonio Rizzo

***Abstract*—Safety-critical systems require an assessment activity to verify that they are able to perform their functions in specified use environments. This activity benefits from evaluation methods that consider these systems as a whole and not as the simple sum of their parts. Indeed, analysis of accidents involving such systems has shown that they are rarely due to the simple failure of one of their components. Accidents are the outcome of a composite causal scenario where human, software, and hardware failures combine in a complex pattern. Unfortunately, dependability analysis and evaluation of safety critical systems are usually based on techniques and methods that consider human and computer separately, and whose results can hardly be integrated.**

**The analogies between the processes of: 1) software-reliability growth due to testing and the related fault removal; 2) improvement of man–machine interface due to preliminary operative feedback; and 3) improvement of operator performance due to learning activity; all suggest a common evaluation approach. Only the first one of these processes is currently modeled using mathematical methods.**

**This paper extends these methods to study the reliability-growth process of other system components: operator and man–machine interfaces. To study the feasibility of the approach, this paper analyzes the results of an experiment in which the reliability of a system is evaluated using trend analysis and reliability-growth models. The evaluation concerns the graphic man–machine interface and the operators, and could easily be extended to the software control system. The experimental results show that trend analysis and reliability-growth models could be complementary to the qualitative evaluation performed within the cognitive science approach. They could offer a quantitative support especially when their information is based on analysis of average values. In this case they could assist several decisions during operator training and especially during interface design, when comparing the effect of different possible interfaces on operator behavior. Moreover they can support the share of the same tools and of the related know-how between the fields of human and software dependability.**

***Index Terms*—Cognitive science, human reliability, man–machine interface, software reliability, system reliability, system safety.**

## ACRONYMS[1]

| | |
|---|---|
| K-S–D | Kolmogorov Smirnov distance |
| L&V | Littlewood & Verrall |
| M–O | Musa Okumoto |
| NHPP | nonhomogenous Poisson process |
| PL | prequential likelihood |
| r.v. | random variable |

Manuscript received February 12, 1998; revised November 11, 1999. This work was supported in part by the European Union-DGXII-Program Human Capital and Mobility, via the "OLOS" research network (Contract CHR-X-CT94-0577).

A. Pasquini is with ENEA, Rome, Italy (e-mail: Pasquini@casaccia.enea.it).

G. Pistolesi is with ISTAT, Rome, Italy (e-mail: Pistoles@istat.it).

A. Rizzo is with the University of Sienna, Sienna, Italy (e-mail: Rizzo@unisi.it).

Publisher Item Identifier S 0018-9529(01)11348-5.

[1]The singular and plural of an acronym are always spelled the same.

## I. INTRODUCTION

INCREASINGLY, computers support humans in carrying out functions requiring either prompt answers, or the solution of complex problems, or decisions based on a large amount of information. The resulting system exhibits a tight integration of software and human resources. In some applications, a failure of this type of system can produce severe harm in terms of human life, environmental impact, and/or economic loss. Typical current examples are the control of nuclear or chemical processes, air traffic control, and transportation systems.

While automation was originally anticipated to decrease the risk arising from operator error, it does not remove people from the systems. Automation merely increases the responsibilities of designers, and moves operators to higher level of supervisory control and decision making. There has been growing recognition of the fact that moving humans to supervisory and emergency-response tasks brings new, previously underestimated, risks. Errors in human decisions and actions still have the potential for extremely serious consequences. At the same time, increased system complexity makes the decision-making process more difficult, both when it is the responsibility of an on-site operator and when it is to be designed (before the system has ever operated) into automated responses by computers and other equipment.

An integrated approach should be applied during the evaluation of systems based on software and human components, to consider their component interactions as well as the interface between the human and the machine. Indeed, analysis of accidents involving such systems has shown that they are rarely due to the simple failure of one of their components. Accidents are the outcome of a composite causal scenario where human, software, and hardware failures combine in a complex pattern. A few, well-known examples of such accident patterns are: 1) The flight-control system of the Phobos I spacecraft, interacting with the ground control, caused the failure of its mission [6]. 2) The Therac 25 radiation therapy machine caused the over radiation of some cancer patients, due to the combination of an architectural flaw with a software bug and operator error [10]. 3) A failure in the hardware of the Crystal River nuclear-plant control-system, combined with a software fault, caused a release of radioactive water [31].

Currently, systems are evaluated using methods that address human, software, and hardware components separately. Most integration efforts are limited to hardware and software co-design, and to some aspects of human–computer interface.

Research usually addresses only one among the hardware, software and human components.

This paper is an experimental attempt to study the reliability trend of systems based on software and human resources. This is done during the phases of operator training, and man–machine interface design. Reliability is evaluated using trend-analysis techniques and reliability-growth models.

Section II introduces one of the most used human-cognitive model in cognitive engineering, describing how humans learn from experience and why their performance and reliability improve with training.

Section III describes the state of the art in modeling and quantifying the reliability-growth process for both human and software, and considers the use of some mathematical methods for evaluating these two system components.

Section IV describes the experiment we organized to study the reliability trend of a simulated system, based on software and human resources.

Section V discusses the results obtained from this experiment.

## II. Cognitive View of Human Performance and Reliability

Cognitive psychology is a discipline studying how humans acquire information, represent it internally, and use it to guide their behavior. This discipline emphasizes the role of intentions, goals, and meanings, as a central aspect of human behavior. An influential classification of the different types of information processing involved in control of systems such as chemical-process plant or nuclear-power generation was developed by Rasmussen and is described in [22], [23]. Rasmussen identified three levels of information-processing at growing levels of conscious control, on which the human behavior is based. He defined these levels as "skill-, rule-, and knowledge-based behavior," describing how switching occurs between the different level of information processing in process control, and how an operator learn from experience.

The skill-based level involves an automated sensory-motor behavior in responding to external signals, with the operator performing the required control-task without conscious attention. Riding-a-bicycle is a good example of this type of behavior: the task is very complex but is performed automatically with the human responding with no conscious attention to signals giving information about speed, slope, and direction. The ability to use this type of behavior in some control tasks is reached and maintained by learning from experience and errors, and using higher levels of information processing for checking progress in the goal-directed activity.

The rule-based level requires a more conscious involvement. Actions are controlled by stored rules or procedures (heuristics); selection of appropriate rules is controlled by inferences about the current state and events. For example, an operator gathers information from various sources and uses them as input to diagnostic rules of the type:

<IF> symptoms are $X$ <THEN> cause of the problem is $Y$; <END_IF>.

Then, having established a plausible cause of the problem on the basis of the pattern of indications, an action rule can then be invoked of the form:

<IF> the cause of the problem is $Y$ <THEN> do $Z$; <END_IF>.

If, as a result of applying the action rule, the problem is solved, the operator will switch to the skill-based level. If the problem is not resolved, further information can be gathered, to try to identify a pattern of symptoms corresponding to a known cause. If the cause of the problem cannot be established, then the operator must use the highest level of information processing. Again, training, experimentation, and errors are necessary to develop and adjust efficient rules, and to identify the conditions under which the rules should be applied.

The knowledge-based level is used to solve problems that cannot be identified and solved using available rules. "In this situation, the goal is explicitly formulated, based on an analysis of the environment and the overall aims, and a plan is constructed. The plan can be formulated: 1) by selection, where different plans are considered and their effect is tested against the goal; 2) by physical trial and error, or 3) by a conceptual understanding of the functional properties of the environment, and prediction of the effects of the plan being considered" [11].

According to this view, Reason defined [26] four ways by which human cognition shows its processing limits leading to human errors:

- **Slip** occurring when there is a mismatch between intention and action: the intention is satisfactory, but the actions are not carried out as planned. A slip is mainly due to some kind of attentive failure in the low-level of action control, and usually occurs in routine situations characterized by automatic and over-practiced behavior.
- **Lapse** consisting of memory failures, and concerning either the intention of the action under execution, or its correct execution, or the information necessary to perform the action that cannot be retrieved from memory (e.g., tip of tongue).
- **Rule-based error** usually consists of the wrong activation of well-known rules or procedures, either in identifying the situation where the rule should be applied or in adopting the plan of action.
- **Knowledge-based error** occurs when a selected plan, or even the intended goal, is not adequate to solve the problem. Knowledge-based errors are attributed to lack of completeness of the mental models used, and/or a fault in causal thinking. People are not able to recognize properly the relation between different aspects of the problem or to achieve an adequate diagnosis of the problem.

These 4 error-categories show how the computational power of the human mind is also the main source of errors, "Knowledge and error flow from the same mental sources, only success can tell one from the other" [15]. Indeed, considering Slips, Lapse, and Rule-Based errors, one can easily observe the enormous limits of the human brain in performing symbolic computation. Complex symbolic processing can occur only through the mediation of external artifacts such as writing, maps, notation [20]. Without such artifacts, even very simple symbolic processing such as deductive processing are not possible.

An operator performing process control and interfacing with software, uses all three levels of information processing and is subject to all four error-types. The selection of the level depends on training, experience, and novelty of the situation. During training, the operator moves naturally down from knowledge-based to skill-based behavior. The operator also uses errors, especially rule based and knowledge based, as a method to improve and maintain the information-processing ability at all the described levels. Thus one can assume, and easily verify with observation, that the reliability of an operator performing a specific control task improves with appropriate training.

## III. State of the Art in Quantitative Reliability Evaluation

Statistical testing for trend analysis gives quantitative information about the reliability trend of a system, it can be used to help determine whether the system reliability is growing or decreasing. It can be particularly useful to understand the trend when there is a high variation in reliability that alternates local increasing and decreasing periods. In practice, it represents a quantitative support for management for decision making during the testing and validation process [9]. Software-reliability growth models attempt to predict the reliability of software on the basis of its failure and fault-removal history. This is defined as the realization of a sequence of r.v. $T_1$, $T_2$, ..., $T_n$, where $T_i \equiv$ "time spent in testing the program after the fault causing failure #($i-1$) has been removed until failure #$i$ occurred." The reliability-growth models follow a black-box approach. No care is given to the single actions causing the reliability-growth and to their interactions. The focus is in their effect, that is in the reliability growth process in its entirety. Several models have been proposed to estimate the reliability in terms of Mean Time To Failure or number of residual faults. References [16] and [32] contain a detailed survey of most of these models. References [2], [3], and [8] contain proposals to decide the most appropriate one for each application, combine the information they provide, or compare them. These models might provide a first, rough reliability estimation and might support project management. In other words they represent a modest but well-understood prediction tool for decision-makers.

In cognitive science, the approach followed while analyzing the reliability of humans in control and supervision is different. The focus of current cognitive engineering is in optimizing the role of the individual in human–machine systems, by understanding how people acquire information, represent it internally, and use it to guide their behavior, in the continuous interaction with the external environment. Little attention has been paid to the quantitative evaluation of human reliability. Cognitive scientists tried to understand the meaning and the sequence of human actions when performing control functions [24]. They developed cognitive models where the single information processing activities and external actions are considered [28]. These models are far too complex for a quantification of their elements and of their interactions. They are used mainly for qualitative consideration with the aim of improving training, equipment design, and procedures. Quantitative considerations regarded essentially timing aspects of human perceptual-motor learning. Perceptual-motor performances of human improve with practice with a relationship that is approximately proportional to a power of the amount of practice [4]. This relation, called power-law of practice, applies to all skilled behavior, both cognitive and sensory motor [19]. But, it has seldom been used to quantify changes in the quality of performances. Most of the work in man–machine interfaces is aimed at providing conditions to optimize human performances and to reduce the probability of failures due to the interface. Guidelines and checklists have been produced to improve the design of interfaces in new systems or to evaluate possible deficiencies in existing ones [27], [29].

System reliability studies are based mainly on the use of formal techniques such as Fault Tree Analysis. Use of this technique and the need to quantify the probability that "human actions are successfully carried out" raised the need of human-error probability estimation. A methodology to provide such an estimation is in [1]; it encompasses task analysis and human-error rate prediction. But, in Fault Tree Analysis, and in similar formal techniques for system reliability evaluation, there is a very mechanistic consideration of humans: they are modeled as hardware components that provide a function when required. The black box approach is applied to humans. A detailed critical analysis of this approach and of its limits is in [25]. One of the main criticisms is that human reliability cannot be assessed in isolation from the external tools that humans use to accomplish a given activity. External tools are essential components of human cognition and, with practice, they become, for humans, the equivalent of internal cognitive tools.

The reliability growth of a system based on software and human resources is the result of:

- software reliability growth due to testing and the related fault removal;
- improvement of man–machine interface due to preliminary operative feedback;
- improvement of the operator performances due to their learning activity.

Table I shows the factors that affect these processes, together with a list of the main techniques used to stress and improve the reliability of the human, interface, and software components in late phases of the system development process.

All the processes lead to a component-reliability growth and these growths are likely to have a similar trend: an initial fast increase in reliability is followed by an asymptotic trend due to limiting factors some of which are in Table I. For example, human learning does not give a complete guarantee that the human who has learned the reason for a particular failure will not fail in those circumstances again. Skill, as well as rules and knowledge, can deteriorate with time for lack of practice; and it is wrong to assume that, if a person possesses a piece of knowledge in a circumstance, this knowledge should be available under all conditions in which it might be useful. Often, the opposite effect is observable: knowledge accessed in one context remains inert in another. For software, something producing a similar effect on reliability can happen, e.g.: there is a certain probability that new faults are introduced during the fault removal process. Even if a fix ensures that the same input con-

TABLE I
PROCESSES LEADING TO RELIABILITY GROWTH

| | Software component | Interface component | Operator component |
|---|---|---|---|
| Technique used to stress the component | Testing according to a specified operational profile. | Evaluation using guidelines. Simulated operative usage and preliminary operative feedback. | Training and simulation. |
| Event leading to system modification | System, sub-system or module failure. | System failure. Interface evaluation results. Operative usage feedback. | Simulation results. System failure. System abnormal/normal working conditions. |
| Event leading to reliability growth | Fault detection and removal. | Interface modification and improvement. | Learning from experience (increasing skill, building rules, increasing knowledge). |
| Factors limiting the reliability growth | Saturation effect on testing strategies. Imperfect debugging. Low frequency faults and operating conditions. Use of incorrect operation profiles. Fault masking. | Use by different operator with different needs. Operator needs changing with experience. Low frequency condition. | Possibility of rare and never-experienced events. Deterioration of skill, rules, knowledge for lack of practice. Possible phenomenon of fatigue, stress, de-motivation, *etc.* |

ditions will not cause the failure to occur once again, it could happen that the software reliability does not increase (because of new faults introduced) or increases less than anticipated.

This paper experiments with a new approach in the reliability analysis and prediction of control systems: to model the reliability-growth process of the system by considering the human, interface, and software system together as a whole. The approach assumes that system-failure is a process that appears to the observer to be random, and that this assumption can be adopted for the failures due to the software, interface, and human components. Then, the whole process can be modeled as stochastic. Software reliability growth is currently analyzed and modeled using mathematical methods such as trend-analysis techniques and reliability models. Adopting the described assumption, these tools can be extended to model the reliability growth and to estimate the future behavior of each system component and of the system as a whole.

This present study focuses on:

- verifying the usability of trend analysis and reliability-growth models to study the reliability-growth process of the operators during training, when using a specific man–machine interface and a specific piece of software;
- verifying the usability of trend analysis to compare the influence of various man–machine interfaces on the reliability-growth of the operators.

The potential positive results of using these mathematical tools are quite evident. Quantification of the reliability growth of the operator can support the:

- Identification of stopping criteria for the operator training. Trend-analysis and reliability-models can provide useful information when controlling the progress of operators during training. For example, they might draw attention to learning problems, or support the instructor in deciding when to stop or change the training activity because the operator performances do not increase any more.
- Comparison of various interfaces on the basis of the operator reliability growth. The trend in reliability growth of a group of operators might provide useful, quantitative information about the man–machine interface they are dealing with. This type of information can be used during interface design that is usually based on an experimental evaluation and a comparison with working prototypes, after a heuristic evaluation performed on mock ups.
- Compatibility of the results obtained while evaluating the system components. The use of the same mathematical methods for different system components ensures that results are comparable and, even keeping in mind the substantial difference between human and machine behavior, represents a first step in the direction of an overall figure of the system reliability.
- Comparison of the effectiveness of various testing, evaluation, and training strategies.

## IV. AN EXPERIMENT OF RELIABILITY TREND-ANALYSIS AND ESTIMATION

*Notation*

| | |
|---|---|
| $n$ | discrete time variable |
| $N$ | observation period of time |
| $i$ | input variables |
| $I$ | set of the input variables |
| $c$ | control variables |
| $C$ | set of the control variables |
| $s$ | output variables describing the state of process |
| $S_a$ | subset of acceptable output variables |
| $S_u$ | subset of unacceptable output variables |
| $S$ | $S_a \cup S_u$: set of the output variables |
| $P$ | process to be controlled |

### A. Assumptions

1) A system failure, due to the software, interface, and human components, is a process that appears to the observer to be random, and can be modeled as stochastic.
2) The set of the input variables ($I$) represents the actual operational variables of the system.
3) Application of the reliability-growth models requires the assumptions of the Geometric [17], M–O [18], and L&V [12] models.

### B. Description of the Approach

Through a Man–Machine Interface the operator had to control a simulated process, by setting the control variables to keep the process output variables within a predefined range.

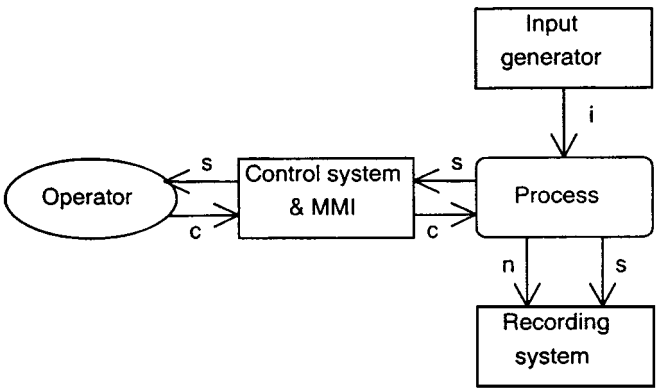


Fig. 1. Flow of the variables.

The value of the output variables, and any failures in keeping them within the required range was recorded.

The process to be controlled was simulated with a function $P$: $S \times I \times C \rightarrow S$.

After giving the initial conditions $i(0)$, $i(1)$, and $s(1)$, the process evolved with:

$$s(n) = P(s(n-1), i(n), c(n)), \qquad n = 2, \ldots, N; \quad (1)$$

$$i(n) \equiv f(r(n), i(n-1) - i(n-2), i(n-1)); \quad (2)$$

for a given function $f$, and with $c = c(n)$ given by the operator as shown in Fig. 1.

After a simple preliminary training, the operator was required to select the $c$ such that $s \in S_a$ for a pre-defined period of time, $N$. The condition $s \in S_u$ is considered a control-system failure due to an operator error, and the interfailure times were recorded.

### C. Hypotheses

From the aims described in Section III, and the organization of the experiment, the following work hypotheses were derived.

1) Operators can learn from their experience and can improve their performance over time. Thus, interfailure times grows with $n$.
2) Operators using a better Man–Machine Interface will learn their task more easily than operators with a less usable interface; this affects the interfailure times.
3) Trend-analysis and reliability-growth models can be used to study the reliability-growth of the operators during training, when using a specific man–machine interface and a specific piece of software.
4) Trend-analysis and reliability-growth models are useful to compare the influence of different man–machine interfaces on the reliability-growth of the operators.

### D. Description of the Experiment

*1) Simulated Process and Control System:* Several process-control problems have been proposed in the literature, to be used as subjects for research in neural-network control methods, or in other forms of learning control. Our selection criterion was to have a control problem requiring from the operator a cognitive effort analogous to the one required in real control and supervision tasks. Use of 3 levels of information processing (skill-, rule-, knowledge-based) was considered more important than the size or the complexity of the problem. The simulated process is a modified version of the blending-process simulator [30] and is shown in Fig. 2.

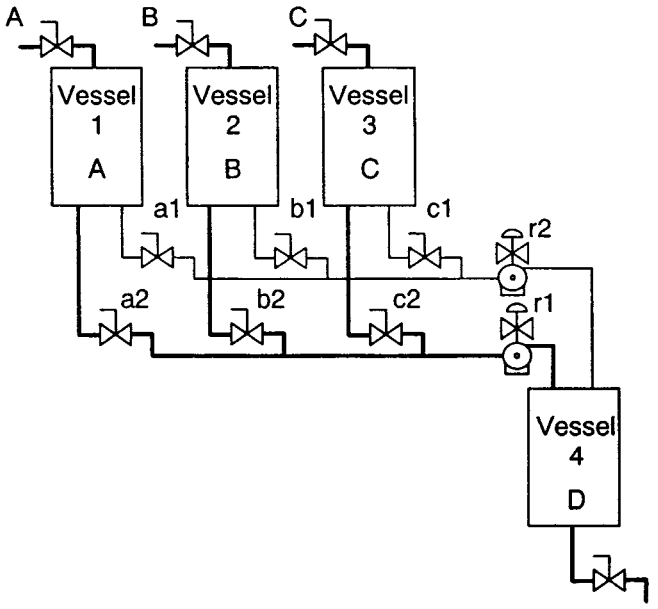


Fig. 2. Simulated process to be controlled.

From the vessels 1–3 the operator has to transport the liquids A, B, C to vessel 4; A, B, C are to be mixed. The operator has to fill vessel 4 with a specified mixture of A, B, C. To fulfill this task, the operator can use a complex of tubes of different flow, controlled by on–off valves and 2 pumps having an adjustable flow and a nonlinear response. The composition of the mixture can change with time, and the operator must perform some additional, less important, control activities at the same time (e.g., control of liquid-level in additional tanks). A software system, developed for the purpose,

- simulates the dynamic process to be controlled;
- provides the interface between the operator and the software control system;
- records all the failures that occur.

The software system and the experimental set-up were validated by a first set of 14 subjects playing the role of operators. On the basis of this preliminary use, a second version of the interface was designed, with a more explicit and "transparent" representation of the system to be controlled. This was obtained by showing "the control elements inserted in the architecture of the system to be controlled." This way, operators can distribute the cognitive process between their mind and the interface, tolerating the mental workload, and having a better comprehension of their control task [33].

*2) Subjects:* The subjects were 20 students (male and female) aged between 20 and 30 years, with a University or high school degree. The only specific experience required was the use of computer-pointing devices.

*3) Experimental Variables:* All subjects received written instructions followed by an oral explanation of the control system and of the control task to perform. They operated contemporaneously, in the same environment, using the same type of hardware. Ten subjects were assigned to each of the two versions of

the interface, and performed their control task for a predefined time. They are called group A (those using the first version of the interface) and group B (using the second version). Subjects were assigned to group A or B randomly, with an equal distribution of males and females to avoid intersexual effects. All the operators were interviewed after the experiment using the Yoked technique to control: how they used the different level of information processing during the experiment, and the adequateness of the interface used.

*4) Selection of the Trend-Test Techniques and of the Reliability-Growth Models:* A comparison of some analytic trend-tests was reported in [9]; it showed the adequateness of Laplace test [5] when the successive failures to be studied are governed by a NHPP. The NHPP assumption is quite obvious here because of the high variation in reliability, that alternates local increasing and decreasing periods. The Laplace test was complemented with the arithmetic mean of the interfailure times. This is a simple, straightforward trend test, directly related to the observed data, that can be useful to study the average trend. As for reliability-growth models, recent work, e.g., [2], [13], have shown that the evaluation of the model performances (statistical evaluation of the fit of the data estimated with the real data) and recalibration of the model are sometime more important than the characteristics of a reliability model in itself. Due to the complexity of a system involving humans as well as the man–machine interface, a new model risks having parameters whose physical meaning is ambiguous or not adequately considered. Thus, the correct use of existing software reliability-growth models was considered as more advisable than the development of new, specific models, at least in this preliminary study. Selection between existing software reliability-growth models was based on assumptions: assumptions that are acceptable for software might not be admissible when dealing with human behavior. A finite number of faults can have a meaning in software, but does not hold for humans when there is always the possibility that the same error is repeated. Then, three infinite-failure category models were selected: Geometric model that assumes an exponential distribution of the time between failures; M–O model that assumes a Poisson-type distribution; L&V model, a Bayes model. These models were applied and evaluated using the tool CASRE [14].

## V. Experiment Results and Discussion

### A. Results

Fig. 3 shows, for each failure, the average of the interfailure times for the operators of the groups A, B. Data of the single operator are not reported here for reasons of clarity and simplicity. Table II shows the standard deviation and the analysis of variance, with the effect on $s$-significance for hypotheses 1 and 2.

Fig. 4 shows the arithmetical mean of the interfailure times for the 2 operator-groups, with the mean calculated for each single failure as

$$\tau(i) = \frac{1}{t} \cdot \sum_{k=1}^{i} \theta_k \tag{3}$$

$\theta_k$ are the interfailure times.

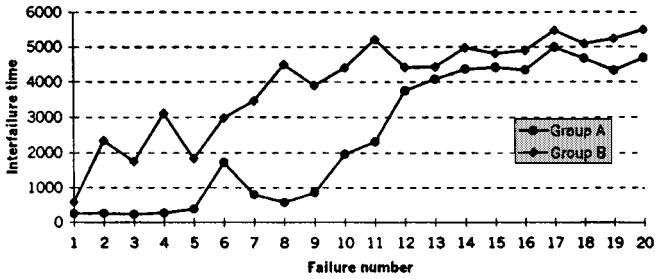


Fig. 3. Average of the interfailure times for groups A and B.

TABLE II
Analysis of Variance for the "Operator Data" and "Level of $s$-Significance for Hypotheses 1 and 2"

| | Hypothesis 1 | Hypothesis 2 |
|---|---|---|
| Std.Dev. group A | 2137 | N/A |
| Std.Dev. group B | 3018 | N/A |
| Degree of freedom | 19 | 19 |
| Fisher dist. | 17.60 | 1.88 |
| $s$-Significance | $p < 0.001$ | $p < 0.05$ |

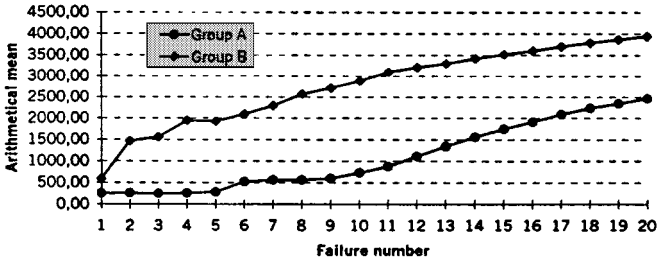


Fig. 4. Arithmetic mean of the interfailure times for operators of groups A and B.

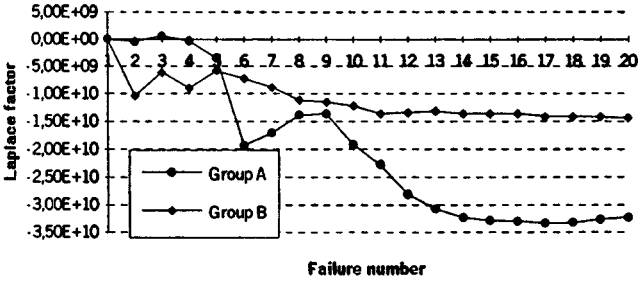


Fig. 5. Laplace test results for the two groups of operators.

The increasing series of $\tau(i)$ indicates a clear reliability growth for the operators, confirming the intuition given by the analysis of the raw data.

Fig. 5 shows the results of the Laplace test for the operators of groups A and B. The value of the Laplace factor is derived as in [9]. The Laplace test shows that the growing trend for the operators of group A is not constant until failure 8. For this group the Laplace factor is stable from failures 1 to 4, indicating local reliability fluctuations. For failures 6–8, the Laplace factor is increasing, indicating a local reliability decrease, despite an overall reliability-growth. After failure 9, the Laplace factor indicates a clear growing trend in reliability.

Fig. 6 shows the results of the Laplace test considering a reduced set of failure for group A. This graph skips the local reliability fluctuations that group A showed until failure 8. After this

TABLE III
EVALUATION OF THE PREDICTIVE ACCURACY OF THE MODELS FOR THE DATA OF GROUPS A AND B

| Model name | Data Group | − log(PL)* | Model bias | Bias trend | Model noise | K-S − D | 95% Fit of K-S − D |
|---|---|---|---|---|---|---|---|
| L&V | A | 8.191E001 | 4.796E-001 | 2.233E-001 | 1.078 | 4.171E-001 | No |
| M-O | A | 8.277E001 | 4.541E-001 | 2.564E-001 | 1.452 | 4.079E-001 | No |
| Geometric | A | 8.556E001 | 3.465E-001 | 2.795E-001 | 1.541 | 3.898E-001 | No |
| L&V | B | 8.615E001 | 4.777E-001 | 1.111E-001 | 1.458 | 4.635E-001 | No |
| M-O | B | 8.59E001 | 5.749E-001 | 1.189E-001 | 2.221 | 4.475E-001 | No |
| Geometric | B | 8.604E001 | 4.866E-001 | 1.112E-001 | 1.082 | 4.570E-001 | No |

* With an estimate of the parameters based on the first half of the raw data

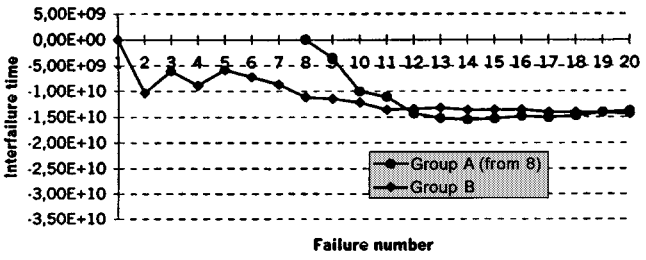


Fig. 6. Laplace test over a reduced set of data for group A.

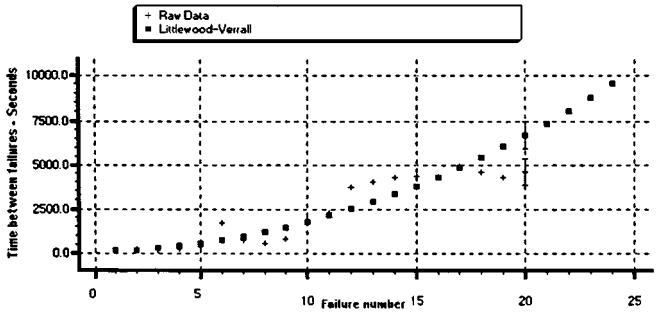


Fig. 7. L&V model applied to the data of group A.

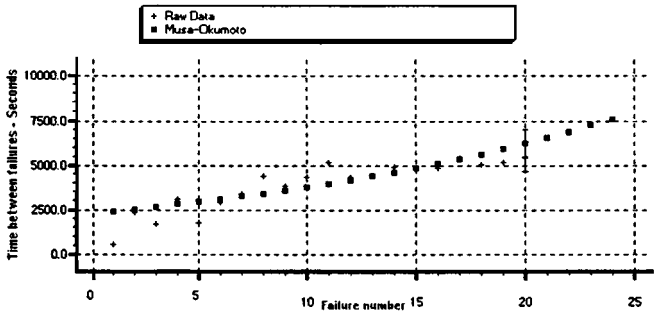


Fig. 8. M–O model applied to the data of group B.

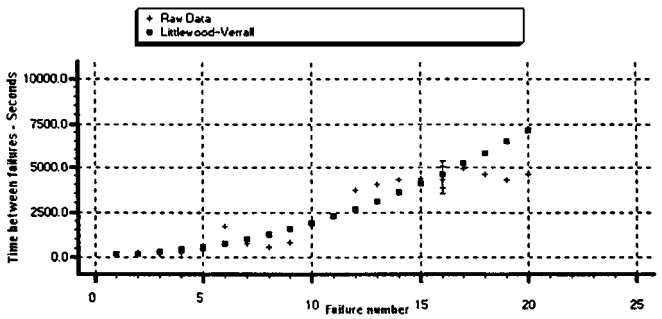


Fig. 9. Example of the predictive ability of the L&V model applied to data of group A.

filtering activity, the reliability trend of the two groups are quite similar. In practice, only the trend of the test is relevant, negative values of the Laplace factor indicate a decreasing failure intensity, while positive values indicate an increasing failure intensity. The possible reason of these global trends and of the initial local fluctuations of group A are discussed at the end of this section.

Three reliability-growth models were applied to raw data of groups A and B. The best models for predictions from specific data must be selected by analyzing the accuracy of past predictions on the same data. CASRE provides several analysis techniques (goodness of fit, PL, bias, noise, and trend) for the models available. Table III shows the result of this analysis for the models used.

The L&V model was selected for the data of group A. This model has a low "goodness of fit" (it does not fit the data at the 95% $s$-significance level), but it has the lower "noise" and best PL. In addition its bias is very stable (optimistic) during the whole period of observation.

Fig. 7 shows the results of the application of this model compared with the raw data. The M–O model was selected for the data of group B. This model shows the best goodness-of-fit (even if this model does not fit the data at the 95% $s$-significance level) and PL with this set of data. The points from 21 to 24 in Figs. 7 and 8 show the reliability trend predicted by the models. They confirm the growing reliability trend of the operators, but one must consider that the models' evaluation of Table III has shown a $s$-significant optimistic bias for both models.

In Fig. 9 the L&V model is applied to the raw data of group A, using only the first 16 data points. The model is then used to estimate the reliability growth from point 17 to point 20; then this estimate is compared with the actual data points.

Fig. 9 offers an intuitive piece of information about the predictive ability of the model, that can complement the PL data shown in Table III. Fig. 10 shows the same information for the M–O model applied to the raw data of group B.

### *B. Discussion*

The first two work-hypotheses are confirmed by the analysis of variance in Table II. The first work-hypothesis is a quite obvious result that can be easily confirmed by observation. The second work-hypothesis supports the idea that different interfaces could be evaluated on the basis of the operator reliability-growth. But this result has a meaning only if one considers the average performance of different operators. There was a high

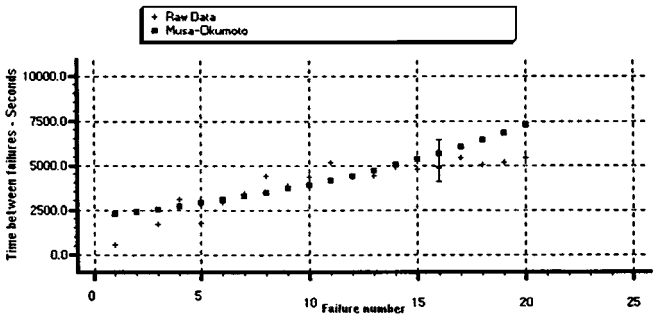


Fig. 10. Example of the predictive ability of the M–O model applied to data of group B.

level of fluctuation in the reliability of single operators and a high variance between different operators. This is also shown by the $s$-significance level $= 0.05$ for the second work-hypothesis.

The validity of the other two work-hypotheses is less evident and is based more on interpretation of the experimental data. Even the analysis of the raw data can show intuitively the superiority of the new interface, used by group B. This analysis shows that with the new interface, operators have part of the knowledge required to control the system, immediately available and represented externally by the interface display. This reduces the effort required to begin working correctly with the plant. After a certain period (after failure 12 in our data), operators working with the old interface reach the same level of knowledge, creating an internal representation of the same information. At this point the difference between the two interfaces decreases. The remaining difference could be because: in particular stress conditions the new interface still has the advantage of reducing the mental workload of the operator. This claim is confirmed and strengthened by the analysis of the trend tests. The arithmetic mean clearly shows the learning process of the operators, eliminating the local fluctuation. This advantage is more visible with the data concerning the single operators (not reported here for lack-of-space) because local fluctuations are more evident than when dealing with average values. Another phenomenon that can be appreciated from trend test is the "saturation" effect of the training process. After a certain period (approximately 14 failures) the increasing trend is less evident, perhaps because of a reduced ability of the operators to increase their skill just from simulation.

This type of information could be used to complement those that are usually used to control the learning process of the operators.

The initial differences between groups A and B are shown also by the Laplace test. But this test evidences also what happens after the initial phase, when the differences between the two interfaces affect less heavily the operator performances. Consider a reduced period of observation for group A, skipping the first seven failures, as shown in Fig. 7, then the Laplace test shows that the reliability trends of the two groups are quite similar.

The reduced number of failures makes the advantages of using trend analysis less evident: most of the observation described in this section could be based on the direct analysis of the raw data. Such an analysis can become much more complex and even unfeasible with the larger failure data sets available in real cases.

Figs. 7 and 8 show the reliability-growth predicted by the two reliability-growth models offering the best predictive ability. These predictions confirm the asymptotic trend of reliability that can be extrapolated from the observation of the raw data and from the trend analysis. The applicability and usefulness of these models is perhaps less evident. Models offering the best predictive accuracy show a constant optimistic bias, and none of the applicable models could fit within 95% of K–S-D. This might be due to the scarcity of the failure-data available (only 20 for each group) that reduces the importance of this study, especially for the reliability-growth models. For larger failure-data sets, trend analysis could improve model applicability by supporting the selection of the best model to use, and by filtering the raw data to eliminate local reliability fluctuations. Use of other models, able to consider the learning process implicit in fault removal, such as the one in [7], could perhaps simulate better the effect of the human knowledge-growth during training. This experiment considered the set of the input variables as representative of the actual operational variables. An analysis of the sensitivity of reliability-growth models, to possible errors in the operational profile [21], should be considered for this specific application context.

### C. Limitations

Quantification is extremely difficult when dealing with human behavior, and inappropriate generalizations of preliminary results can be completely misleading. Thus, consider carefully the limitation of this study when trying to draw general conclusions.

The results changed substantially from subject to subject, even when the subject operating conditions (e.g., interface they were using, task to perform) remained the same. Results obtained from one operator performing a specific task can be considered valid for that operator in those specific operating conditions only. General observations can be drawn only when averaging the results of several different subjects and keeping in mind that single operators can have appreciable distance from the mean results. The complexity of the control problem was limited for practical reasons. Then, the operators reached a good level of control of the simulated process with a limited number of training sessions and a limited number of failures. To mitigate this limitation, we verified the cognitive effort and the type of information processing used by the operators when dealing with the simulated control problem. This was done by analyzing the results, and interviewing the subjects after the control sessions. Even with the obvious differences of scale and complexity, there was the activation of the same cognitive mechanisms and the use of all levels of information processing required by a real process control problem.

The experimental conditions can be quite different from the real operating conditions of an operator. The sessions were concentrated within a reduced period of time. This resulted in reduced deterioration of skill. And the quality of human answer is strongly affected by factors such as stress, fatigue, and motivation, over which there was very limited control. For example,

the operator does not give the same value to simulation as to reality. Again this could result in high fluctuation in reliability and in appreciable differences among different subjects.

In real conditions, operator training simulation is complemented by several other techniques. The experimental conditions reproduced only part of the operator training process. Conclusions about the usability of trend analysis and reliability-growth models during operator training must consider this limitation.

A limited number of failures were considered during the experiment, while trend tests and software reliability-growth models would require a much larger set of data. But, from a practical view-point, this limitation could strengthen the experimental results. Trend tests are likely to show their effectiveness, in supporting the analysis of the reliability trend, much better with larger samples of data, while the direct analysis of the raw data can be less intuitive and immediate. Analogously, software-reliability growth models are likely to fit better when dealing with larger data sets.

## REFERENCES


[1] B. J. Bell and A. D. Swain, "Overview of a procedure for human reliability analysis," *Hazard Prevention*, vol. 1, pp. 22–25, 1985.

[2] S. Brocklehurst and B. Littlewood, "New ways to get accurate reliability measures," *IEEE Software*, vol. 9, pp. 34–42, 1992.

[3] S. Brocklehurst, M. Lu, and B. Littlewood, "Combination of predictions obtained from different software reliability growth models," in *Proc. 10th Annual Software Reliability Symp.*, 1992, pp. 24–33.

[4] S. K. Card, T. P. Moran, and A. Newell, "The model human processor. An engineering model of human performance," in *Handbook of Perception and Human Performance*, K. R. Boff, L. Kaufman, and J. P. Thomas, Eds. New York: Wiley, 1986.

[5] D. R. Cox and P. A. W. Lewis, *The Statistical Analysis of a Series of Events*. New York: Wiley, 1978.

[6] R. Hartley, "Phobos 1 & 2 computer failures," *Science*, vol. 245, no. 8, p. 47, Sept. 1989.

[7] R. H. Hou, S. Y. Kuo, and Y. P. Chang, "Hyper-geometric distribution software reliability growth model with imperfect debugging," in *Proc. 6th Int. Symp. Software Reliability Eng.*. New York: IEEE Press, 1995, pp. 195–200.

[8] A. Iannino, J. D. Musa, K. Okumoto, and B. Littlewood, "Criteria for software reliability model comparisons," *IEEE Trans. Software Eng.*, vol. 10, pp. 687–691, 1984.

[9] K. Kanoun and J. C. Laprie, "Trend analysis," in *Software Reliability Engineering*, M. R. Lyu, Ed. New York: McGraw-Hill, 1996.

[10] N. G. Leveson and C. S. Turner, "An investigation of the Therac-25 accidents," *IEEE Computer*, pp. 18–41, July 1993.

[11] N. G. Leveson, *Safeware*: Addison-Wesley, 1995.

[12] B. Littlewood and J. Verrall, "A Bayesian reliability growth model for computer software," *J. Royal Statistical Soc.*, vol. 22, no. 3, pp. 332–346, 1973.

[13] B. Littlewood and L. Strigini, "Validation of ultra-high dependability for software-based systems," *Communication ACM*, vol. 36, no. 11, pp. 69–88, 1993.

[14] M. R. Lyu and A. P. Nikora, "CASRE—A computer aided software reliability estimation tool," in *Proc. CASE*, 1992, pp. 264–275.

[15] E. Mach, *Knowledge and Error*: Reidel Publishing, 1967.

[16] Y. K. Malaiya and P. K. Srimani, *Software Reliability Models: Theoretical Developments, Evaluation, and Application*: IEEE CS Press, 1991.

[17] P. B. Moranda, "Event-altered rate models for general reliability analysis," *IEEE Trans. Reliability*, vol. R-28, pp. 376–381, 1979.

[18] J. D. Musa and K. Okumoto, "A logarithmic Poisson execution time model for software reliability measurement," in *Proc. 7th Int. Conf. Software Eng.*. New York: IEEE Press, 1984, pp. 230–238.

[19] A. Newell and P. S. Rosenbloom, "Mechanisms of skill acquisition and the law of practice," in *Cognitive Skills and Their Acquisition*, J. R. Anderson, Ed: Erlbaum, 1981.

[20] D. A. Norman, *Things That Make Us Smart*. Reading, MA: Addison Wesley, 1995.

[21] A. Pasquini, A. N. Crespo, and P. Matrella, "Sensitivity of reliability growth models to operational profile errors *vs* testing accuracy," *IEEE Trans. Reliability*, vol. 45, no. 4, pp. 531–540, Dec. 1996.

[22] J. Rasmussen, "Some remarks on mental load," in *Mental Workload: Its Theory and Measurement*, N. Moray, Ed. New York: Plenum, 1979.

[23] ——, "Human factors in high risk technology," in *High Risk Safety Technology*, A. E. Green, Ed. New York: Wiley, 1982.

[24] ——, *Information Processing and Human–Machine Interaction*: Elsevier Science, 1986.

[25] J. Rasmussen, K. Duncan, and J. Leplat, *New Technology and Human Error*. New York: Wiley, 1987.

[26] J. Reason, *Human Error*: Cambridge Univ. Press, 1988.

[27] A. Rizzo, O. Parlangeli, E. Marchigiani, and S. Bagnara, "Guidelines for managing human error," *SIGCHI Bullettin*, vol. 6, pp. 125–131, 1996.

[28] T. B. Sheridan, "Task allocation and supervisory control," in *Handbook of Human–Computer Interaction*, M. Helander, Ed: Elsevier Science, 1988.

[29] I. Smith and J. Mosler, "Too much too soon: Information overload," *IEEE Spectrum*, pp. 51–55, June 1987.

[30] K. Van Gelder, "Human error with a blending-process simulator," *IEEE Trans. Reliability*, vol. R-29, pp. 258–264, 1980.

[31] H. Voysey, "Problems of mingling men and machines," *New Scientists*, pp. 416–417, Aug. 1977.

[32] M. Xie, "Software reliability models—A selected annotated bibliography," *Software Testing, Verification and Reliability*, vol. 3, pp. 3–28, 1993.

[33] J. Zhang and D. A. Norman,, "Representations in distributed cognitive tasks," *Cognitive Science*, vol. 18, no. 1, pp. 87–122, 1994.



**Alberto Pasquini** received the degree of Doctor in electronic engineering in 1978 from the University "La Sapienza" of Rome.

He is with the Italian research agency for new technology, energy, and environment (ENEA). His research interests are in software reliability and safety, software quality, and human–computer interaction. He was the scientist responsible of the OLOS research network.

**Giuliano Pistolesi** received the M.S. degree in psychology in 1994 and the Ph.D. degree in medical application of the information technology in 1998 from the University "La Sapienza" of Rome.

He collaborated in the study presented in this paper during the year spent as visiting researcher with Human Reliability Associate in the U.K. He is now with the Italian Institute of Statistics (ISTAT). His research interests are in artificial intelligence, neural networks, and human–computer interaction.

**Antonio Rizzo** received the degree of Doctor in psychology in 1984 from the University "La Sapienza" of Rome.

He spent nine years with the Italian Research Council (CNR), and has been with the University of Sienna since 1993. He is president of the European Association for Cognitive Ergonomics. His research interests are in cognitive science, distributed cognition, safety, and the human-factor in complex working organizations.