SoRel: A Tool For Reliability Growth Analysis and Prediction From Statistical Failure Data*

K. Kanoun M. Kaâniche J.-C. Laprie S. Metge**

LAAS - CNRS
7, Avenue du Colonel Roche - 31077 Toulouse Cedex (France)

Abstract

This paper presents a tool for ***S****oftware (and hardware)* ***Rel****iability analysis and evaluation: SoRel. The tool implements a global method for reliability follow up and evaluation in presence of reliability growth due to design fault removal. SoRel is composed of two parts allowing respectively application of trend tests and reliability growth models. The paper presents the method, the trend tests and reliability growth models implemented by SoRel, shows how they can help during the validation process and describes some functionalities of the tool. The main features of the demonstration are outlined.*

Introduction

The objectives of software reliability evaluation in presence of reliability growth are numerous and are closely related to the point of view adopted (the supplier or the customer), and the life-cycle phase concerned. The supplier is interested in the management of the validation and maintenance activities whereas the customer is more concerned by the reliability of the resulting product in operational life. **SoRel** — which is a tool for **So**ftware (and hardware) **Rel**iability analysis and prediction — helps in achieving these objectives thanks to the combined use of trend tests and of reliability growth models. It provides qualitative and quantitative elements concerning, for instance, a) the evolution of the reliability in response to the debugging effort, b) the estimation of the number of failures for the following periods of time so as to plan the test effort and the numerical importance of the test and/or maintenance team and c) the prediction of reliability measures such as the mean time to failure, the failure rate or the failure intensity.

SoRel is based on a global method for software reliability follow up and evaluation that has been developed at LAAS and applied to several real-life systems (see e.g., [7] or [8]). This method relies on qualitative and quantitative analyses, it is intended to better define the users' real needs in the field of software reliability. It is briefly reviewed in the paper.

The paper is divided into three sections. Section 1 outlines the method implemented by SoRel, gives a general overview of the tool and shows the type of results that can be obtained from SoRel. Section 2 is devoted to the description of the tool. Section 3 describes the demonstration.

1. SoRel general presentation

SoRel is composed of two modules allowing respectively the application of **trend tests** and **reliability growth models**. It is able to operate on two types of failure data a) inter-failure times and b) number of failures per unit of time (i.e., failure intensity), allowing application of two types of reliability growth models, respectively time domain and interval domain as called in [15]. Figure 1 gives the organization of SoRel as it is seen by the user. The reliability evolution is analysed through trend test application. Selection of the model to be applied is based on the result of trend tests and the objectives of the analysis. The remaining part of this section presents the trend tests and the reliability growth models implemented in SoRel.

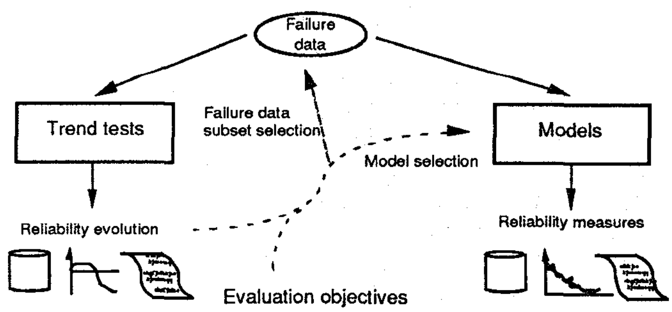


Figure 1: SoRel organization

* The work reported in this paper was partially supported by the ESPRIT Basic Research Action on Predictably Dependable Computing Systems (Action no. 3092).

** Now, with CEP-SYSTÈMES Toulouse

0731-3071/93 $3.00 © 1993 IEEE

1.1. Trend tests

Two reliability trend tests are available: the arithmetic mean and the Laplace test—for both inter-failure times and failure intensity data.

1.1.1. Arithmetical mean test

This test consists of calculating τ_k, the arithmetic mean of the first k inter-failure times (resp. number of failures per unit of time). When τ_k's form an increasing series (resp. decreasing), reliability growth can be deduced. This test is very simple and its interpretation is easy as it is directly related to the collected measure.

1.1.2. Laplace test

The Laplace test, which is a statistical test, consists of calculating the Laplace factor, *u*, whose expressions are given in Figure 2. In practice, in the context of reliability growth, negative values of the Laplace factor suggest reliability growth whereas positive values suggest reliability decrease; values oscillating between -2 and +2 indicate stable reliability.

These practical considerations are deduced from the significance levels associated with the statistics, for instance, for a significance level of 5% the null hypothesis "no trend against trend" is rejected for $|u|>1.96$.

Inter-failure time data

$$u(k) = \frac{c - m}{s_k} \sqrt{12\,(k-1)} \qquad k = 2,\dots, n$$

$$c = \frac{1}{k-1} \sum_{i=1}^{k-1} s_i$$

$$m = \frac{s_k}{2}$$

$$s_i = \sum_{j=1}^{i} t_j$$

t_j : time interval between failures (j-1) and j

s_i : instant of failure i occurrence

n : number of failures observed

Failure intensity data

$$u(k) = (c - m) \Big/ \sqrt{\frac{k^2 - 1}{12\, y_k}} \qquad k = 2,\dots, p$$

$$c = \sum_{i=1}^{k} (i-1)\, n_i / y_k$$

$$m = \frac{k-1}{2}$$

$$y_k = \sum_{i=1}^{k} n_i$$

n_i : number of failures during time unit i

y_k : cumulative number of failures until time unit k

p : total number of units of time

Figure 2: Laplace factor expressions

The users of SoRel can utilize the Laplace test as a conventional statistical test. However, in our approach, we extended it to identify global and local trends [9]. As the Laplace factor is evaluated using all the data collected up to the unit of time considered, it reflects the *global* variation of reliability. *Local* fluctuations can be detected by studying the variation of u(k): for example, when u(k) is positive and tends to decrease it suggests a decrease in the number of failures observed over the considered period which means that, locally, reliability tends to increase although a global decrease is observed. This is summarized in Figure 3. Global reliability decrease over A and B is due to data observed during period A, if the latter are not considered for evaluation purposes, period B will display local and global reliability growth.

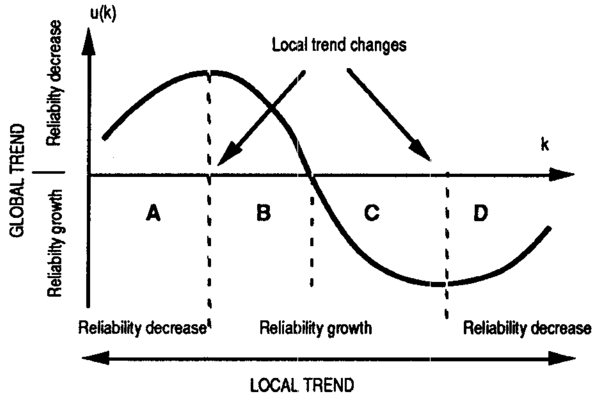


Figure 3: Laplace test interpretation

1.1.3. Practical use of trend tests

Trend analyses are of great help in appreciating the efficiency of test activities and controlling their progress. They help considerably the software development follow up. Indeed, graphical tests are very often used in the industrial field [3, 14, 16], even though they are called differently, such as descriptive statistics or control charts. Three typical situations are outlined hereafter.

Reliability decrease at the beginning of a new validation activity is generally expected and is considered as a normal situation. Reliability decrease may also result from regression faults. If the duration of the period of decrease seems long, one has to pay attention and, in some situations, if it keeps decreasing this can point out some problems within the software: the analysis of the reasons of this decrease as well as the nature of the activated faults is of prime importance in such situations. **Reliability growth** after reliability decrease is usually welcomed since it indicates that, after first faults removal, the corresponding validation activity reveals less and less faults. **Stable reliability** with almost no failures indicates that the related validation activity has reached a "saturation": application of the associated test sets does not reveal new faults, or the corrective actions performed are of no perceptible effect on reliability. One has either to stop testing or to introduce new sets of tests or to proceed to the next phase.

Furthermore, trend analyses may be of great help for reliability growth models to give better predictions as shown in the next section.

1.2. Reliability growth models

Four reliability growth models are implemented: the hyperexponential model (Kanoun-Laprie) [11], the exponential model (Goel-Okumoto) [4], the S-Shaped model (Yamada et al.) [17] and the doubly stochastic model (Littlewood-Verrall) [12]. The S-Shaped model (SS) is an interval domain model, the doubly stochastic model (DS) is time domain whereas the hyperexponential model (HE) and the exponential model (EXP) are both time and interval domain.

These models allow different kinds of behavior to be modeled: HE, EXP and DS model a decreasing failure rate; they yield better results when applied to data displaying reliability growth, that is, interval C of Figure 3. Indeed, these models can be applied also to interval B-C of Figure 3 discarding failure data pertaining to A. The SS model is characterized by an increasing failure rate followed by a decreasing failure rate, it produces good results when applied to failure data belonging to an interval such as A-B-C. For D, prediction is delayed until observing an interval of reliability growth.

Figure 4 summarizes the characteristics of these models. Depending on the model, the main quantities that can be evaluated are: the mean time to next failure (or MTTF), the failure intensity, the cumulative number of failures and the residual failure rate of the software. It is worth noting that HE is the only model allowing evaluation of the residual failure rate in operation.

Model execution is carried out into two steps: parameter estimation (i.e., model calibration using an inference procedure) and reliability evaluation. Two inference procedures are used, depending on the model: maximum likelihood or least square. Both of them need a numerical optimization procedure to estimate the parameters of the models. For two-parameter models (EXP and SS), the numerical values are obtained via the Newton-Raphson iterative method whereas for three-parameter models (HE and DS) they are evaluated via the Powell numerical method [13].

SoRel enables the user a) to determine how well the selected model fits the data and b) to compare estimations issued from several models. The goodness-of-fit criteria are: a) the Kolmogorov-Smirnov statistics [1], b) the prequential likelihood [2] and c) the residue [6]. The first two criteria are evaluated only for inter-failure time data. The residue is evaluated for both failure data types, it is based on the difference between the observed measure and its expectation from the model.

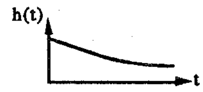
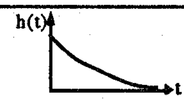
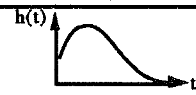
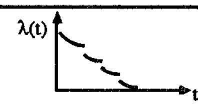
Model	h(t) or λ(t) shape
Hyperexponential $h(t) = \frac{\omega \zeta_{sup} e^{-\zeta_{sup} t} + \varpi \zeta_{inf} e^{-\zeta_{inf} t}}{\omega e^{-\zeta_{sup} t} \; \varpi e^{-\zeta_{inf} t}}$	
Exponential $h(t) = N \phi \exp^{-\phi t}$	
S-Shaped $h(t) = N \phi^2 t \exp^{-\phi t}$	
Double Stochastic $\lambda_i(t) = \frac{\alpha}{t+\psi(i)} \qquad \psi(i) = \beta_1+\beta_2 i$	

Figure 4: Reliability growth models implemented in SoRel

A model can be analysed according to its *retrodictive capability* and *predictive capability*. The retrodictive capability expresses model ability in reproducing the observed behavior of the software. The predictive capability reflects the model ability in predicting future behavior of the software, from the observed failure data. Retrodictive and predictive capabilities are measured through goodness-of-fit criteria.

1.3. SoRel within the software validation process and reliability evaluation

SoRel has been used to follow up and evaluate the reliability of several real-life systems. The characteristics of some of these systems are summarized in Table 1.

For the validation phase, the main results concern the evolution of reliability in response to debugging activities and the prediction of the number of faults that will be activated over the next periods of time [5]. During operation, the objectives of reliability analysis are more various, we give hereafter some examples illustrated through the results obtained for the first three systems which are electronic switching systems (ESS). For the E10-B, the evaluation of the residual failure rate in operation carried out from failure data collected on the software in operation [6] allowed the dependability of the whole ESS to be evaluated (accounting for hardware and software). For the TROPICO-R ESSs we have followed two complementary approaches [8, 10]:

- from the supplier point of view, estimation of the maintenance effort to provide in operation in order to satisfy the correction reports issued from the various customers,
- from the customer point of view, estimation of the residual failure rate in operation in order to evaluate the impact of software reliability on the whole ESS reliability.

System	Languages	Volume	Observation	Phases	# Systems	# FR and/or CR
E10-B	Assembler	100 k-bytes	3 years	Val. / Op.	1400	58 FR / 136 CR
TROPICO-R 1500	Assembler	300 k-bytes	27 months	Val. / Op.	15	461 CR
TROPICO-R 4096	Assembler	350 k-bytes	32 months	Val. / Op.	42	227 CR
Telecommunication Equipment	PLM-86	$5\ 10^5$ inst.	16 months	Val.	4	2150 FR
Work station	various	--	4 years	Op.	1	414 FR

FR: Failure Report CR : Correcrion Report Val.: Validation Op.: Operation

Table 1: Characteristics of some real-life software systems studied by SoRel

2. SoRel Description

SoRel runs on Macintosh II-xx computer equipment with an arithmetical co-processor. The human / machine interface has been denoted special attention. It is interactive: it is menu-driven and uses the multiple window management facilities of the Macintosh. The program is modular and new reliability growth tests and models can easily be added. It is written in Pascal (5.000 lines of code) and requires about 200 K bytes of memory. A user guide and a tutorial are available. The user guide explains how to use the tool and provides example sessions as well as samples of input and output files. The tutorial presents the method, the trend tests and the models implemented. On line help is provided for the main commands.

SoRel is composed of two modules "**TREND**" and "**MODELS**" corresponding respectively to trend analysis and model application The two modules accept the same input data files which can be created and changed by a word processing or graphic editor. Numerical results are displayed immediately on the screen during the execution process. Additionally, the corresponding curves can be plotted upon user's request. The results are also recorded in the form of ASCII files that can serve as input to other Macintosh applications (such as Excel) allowing for instance comparison of results issued from different model applications.

The main menu commands allow cancellation of the last changes (Resume) and exit from the program (Quit). The features specific to each module are described in the rest of the section.

2.1. "TREND" module

Selection of the trend test to be applied is achieved as indicated in Figure 5. SoRel prompts the user for the data set type as shown in Figure 6 and then for the input data file name (Figure 7). The user can either a) choose the file name among the available data set names in the same folder or not, or b) enter directly the name of the file input. (Figure 8).

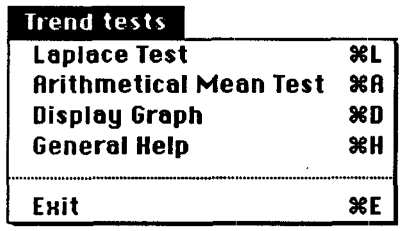


Figure 5: Trend test selection

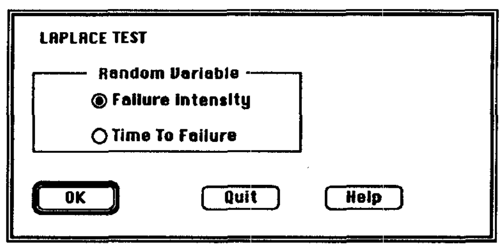


Figure 6: Selection of data type

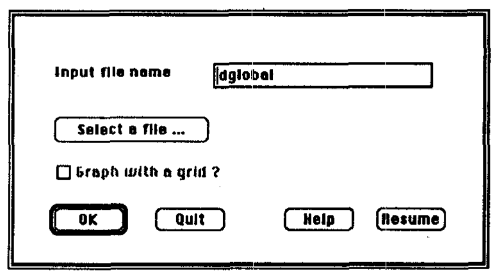


Figure 7: Input file name entry

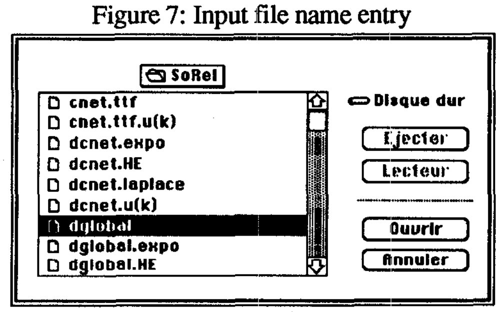


Figure 8: Selection of the input file name

The trend test may be applied to sub-sets of the data recorded in the selected file in order to highlight the local trend: the user indicates the rank of the first data item to be considered (Figure 9). Figure 10 gives an example of graphical results.

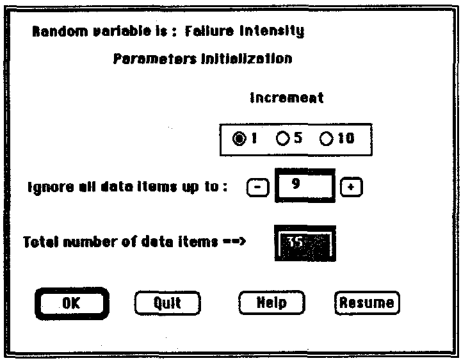


Figure 9: Definition of the failure data sub-set

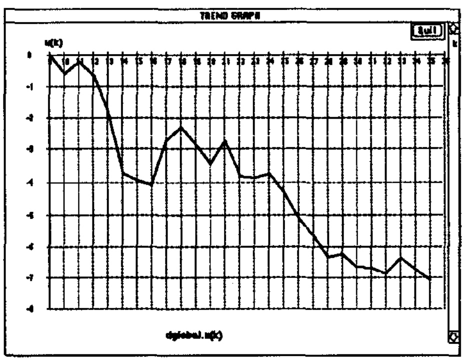


Figure 10: Graphical results for trend tests

2.2. "MODELS" module

The user selects the model (Figure 11) according to the trend displayed by the data set and to the evaluation objectives. The user has to indicate the input data type as well as the measure to be evaluated (Figure 12), then he is prompted for all the input needed to apply the selected model as indicated in Figure 13.

The dialogue areas are defined as follows. Area **PS**, for *initial Parameter Setting, is needed when the model* parameters are evaluated via the Powell method [13]. The user has to supply initial approximations of the parameters at the optimum. Area **DP**, for Data Partition, allows the user to define a) the data sub-set from which the parameters will be evaluated and b) the prediction interval. Another required input is the use of a window or not for model Calibration, area **C**. Finally area **V** points out the interval over which the Validation criteria will be evaluated (this interval may or may not correspond to the prediction interval). The **Set Options** command allows the user to adjust some parameters of the optimization procedure, such as the maximum number of iterations and the convergence criteria which have been given fixed values by default.

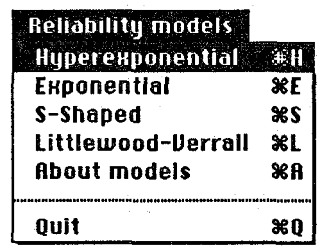


Figure 11: Model selection

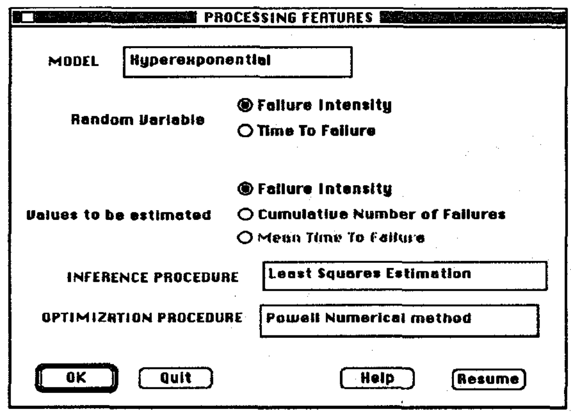


Figure 12: Input data type and output measure selection

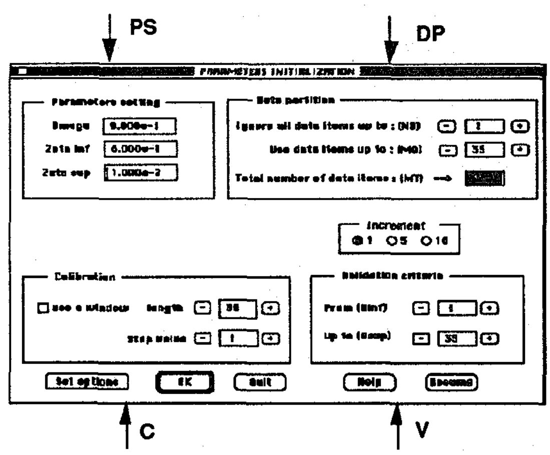
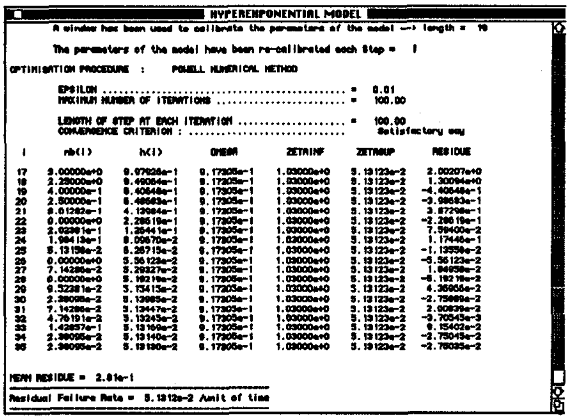


Figure 13: Model application initialization

Figure 14 gives an example of numerical results whereas graphical results are displayed on user's request (Figure 15). The goodness-of-fit criteria are given with the numerical results.

3. The demonstration

SoRel will be demonstrated using failure data collected on a real-life software system: the TROPICO-R 4096 ESS.Data have been collected over about two years including the end of validation and the beginning of operational life. The two types of input data files (inter-failure times and failure intensity) will be addressed. Since the results are displayed immediately, it is possible to make several executions to show the main features of the tool.



HYPEREXPONENTIAL MODEL

A window has been used to calibrate the parameters of the model --> length = 10

The parameters of the model have been re-calibrated each Step = 1

OPTIMISATION PROCEDURE : POWELL NUMERICAL METHOD

EPSILON .... = 0.01
MAXIMUM NUMBER OF ITERATIONS .... = 100.00

LENGTH OF STEP AT EACH ITERATION .... = 100.00
CONVERGENCE CRITERION : .... Satisfactory way

I	nb(I)	h(I)	OMEGA	ZETAINF	ZETASUP	RESIDUE
17	3.00000e+0	9.97028e-1	9.17305e-1	1.03000e+0	5.13123e-2	2.00207e+0
18	2.25000e+0	9.49064e-1	9.17305e-1	1.03000e+0	5.13123e-2	1.30094e+0
19	4.00000e-1	8.40648e-1	9.17305e-1	1.03000e+0	5.13123e-2	-4.40648e-1
20	2.50000e-1	6.48683e-1	9.17305e-1	1.03000e+0	5.13123e-2	-3.98683e-1
21	8.01282e-1	4.13984e-1	9.17305e-1	1.03000e+0	5.13123e-2	3.87298e-1
22	0.00000e+0	2.28619e-1	9.17305e-1	1.03000e+0	5.13123e-2	-2.28619e-1
23	2.02381e-1	1.26441e-1	9.17305e-1	1.03000e+0	5.13123e-2	7.59400e-2
24	1.98413e-1	8.09670e-2	9.17305e-1	1.03000e+0	5.13123e-2	1.17446e-1
25	5.13158e-2	6.26715e-2	9.17305e-1	1.03000e+0	5.13123e-2	-1.13558e-2
26	0.00000e+0	5.56123e-2	9.17305e-1	1.03000e+0	5.13123e-2	-5.56123e-2
27	7.14286e-2	5.29327e-2	9.17305e-1	1.03000e+0	5.13123e-2	1.84958e-2
28	0.00000e+0	5.19219e-2	9.17305e-1	1.03000e+0	5.13123e-2	-5.19219e-2
29	9.52381e-2	5.15413e-2	9.17305e-1	1.03000e+0	5.13123e-2	4.36968e-2
30	2.38095e-2	5.13985e-2	9.17305e-1	1.03000e+0	5.13123e-2	-2.75889e-2
31	7.14286e-2	5.13447e-2	9.17305e-1	1.03000e+0	5.13123e-2	2.00839e-2
32	4.76191e-2	5.13245e-2	9.17305e-1	1.03000e+0	5.13123e-2	-3.70543e-3
33	1.42857e-1	5.13169e-2	9.17305e-1	1.03000e+0	5.13123e-2	9.15402e-2
34	2.38095e-2	5.13140e-2	9.17305e-1	1.03000e+0	5.13123e-2	-2.75045e-2
35	2.38095e-2	5.13130e-2	9.17305e-1	1.03000e+0	5.13123e-2	-2.75035e-2

MEAN RESIDUE = 2.81e-1

Residual Failure Rate = 5.1312e-2 /unit of time

Figure 14: Numerical results

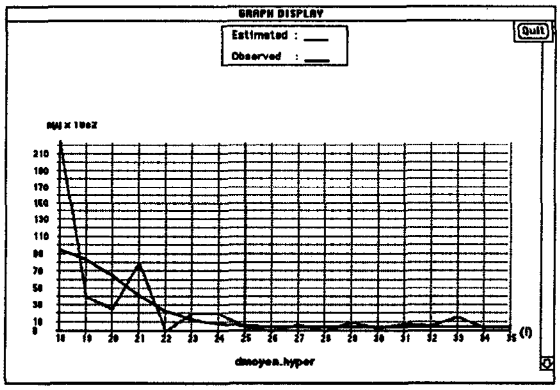


Figure 15: Graphical results

The input data files will be displayed and commented. Trend tests will be applied to the considered data sets. Emphasis will be put on the conclusions that can be drawn from trend test application and the results will be commented showing the link between the reliability trend change indicated by SoRel and the different phases of the software life-cycle. We will also show how to use trend results before reliability growth model application.

Reliability growth models will then be applied to subsets of these input data files with and without accounting for trend results in order to show the improvement of the estimations. We will first carry out retrodictive evaluations and show the capability of the models to reproduce the observed behavior of the software. Application of the models in a predictive way will then be carried out. The two types of failure data will be used during the demonstration allowing various reliability measures to be evaluated, i.e., MTTF, cumulative number of failure, failure intensity and failure rate. We will also compare results issued from application of various models to the same data sets.

Acknowledgments

We wish to acknowledge the very helpful advice we received during the development of SoRel from our colleagues, in particular Yves Crouzet and the late and sorely-missed Christian Béounes.

References

[1] D.R.Cox, P.A.W.Lewis, *The statistical analysis of series of events*, London, Chapman & Hall, 1966.

[2] A.P.Dawid, "Statistical theory: The prequential approach", *J. Roy. Statist. Soc. A*, vol 147, 1984, pp. 274-292.

[3] R.B.Grady, D.R.Caswell, *Software metrics: establishing a company-wide program*, Hewlett Packard Company, Prentice Hall, Inc., 1987.

[4] A.L.Goel, K.Okumoto, "Time Dependent Error-Detection Rate Model for Software and other Performance Measures", *IEEE Trans. on Reliability*, vol R-28, no. 3, 1979, pp. 206-211.

[5] M.Kaâniche, K.Kanoun, S.Metge, "Failure analysis and validation monitoring of a telecommunication equipment software system", *Annales des Télécommunications*, Vol. 45, no. 11-12, 1990, pp. 657-670, *in French*.

[6] K.Kanoun, T.Sabourin, "Software dependability of a telephone switching system", Proc. 17th *IEEE Int. Symp. on Fault-Tolerant Comp. (FTCS-17)*, Pittsburgh, Pennsylvania, 1987, pp. 236-241.

[7] K.Kanoun, J.-C.Laprie, T.Sabourin, "A method for software reliability growth analysis and assessment", Proc. *Software Engineering & its Applications*, Toulouse, France, 1988, pp. 859-878.

[8] K.Kanoun, M.Bastos Martini, J.Moreira De Souza, "A method for software reliability analysis and prediction — application to the TROPICO-R switching System", *IEEE Transactions on Software Engineering*, Vol. 17, no. 4, 1991, pp. 334-344.

[9] K.Kanoun, J.-C.Laprie, "The role of trend analysis in software development and validation", Proc. *SAFECOMP'91*, Trondheim, Norway, 1991, pp. 169-174.

[10] K.Kanoun, M.Kaâniche, J.-C.Laprie, S.Metge, "SoRel: a tool for software reliability analysis and evaluation from statistical failure data", Proc. *8th Symposium on Reliability and Maintainability*, Grenoble, France, 1992, pp. 422-431, *in French*.

[11] J.-C.Laprie, K.Kanoun, C.Béounes, M.Kaâniche, "The KAT — (Knowledge-Action-Transformation) — Approach to the modeling and evaluation of reliability and availability growth", *IEEE Trans. on Software Engineering*, Vol. 17, no. 4, 1991 pp. 370-382.

[12] B.Littlewood, J.L.Verrall, "A Bayesian Reliability Growth Model for Computer Software" J. Royal Stat. Soc., C(App. stat.), 22, 1973, pp. 332-336.

[13] S.S.Rao, *Optimization Theory and Applications*, Wiley Eastern Limited, 1978..

[14] N.Ross, "The collection and use of data for monitoring software projects", *Measurement for software control and assurance*, Edited by B.A.Kitchenham and B.Littlewood, Elsevier Applied Science, London and New York, 1989, pp. 125-154.

[15] G.E.Stark, "A survey of software reliability measurement tools", *Proc. Int. Symp. on Software Reliability Engineering (ISSRE)*, Austin, Texas, 1991, pp. 90-97.

[16] V.Valette, "An environment for software reliability evaluation", Proc. *Software Engineering & its Applications*, Toulouse, France, 1988, pp. 879-897.

[17] S.Yamada, M.Ohba, S.Osaki, "S-Shaped Reliability Growth Modeling for Software error Detection", IEEE *Trans. on Rel.* vol. R-32, no. 5, 1983, pp. 475-478