A PROBLEM STATEMENT AND CLASSIFICATION SCHEME FOR COMBINED HARDWARE/SOFTWARE RELIABILITY MODELING

Jorge L. Romeu; IIT Research Institute; Rome

Kieron A. Dey; IIT Research Institute; Rome

Key Words: Reliability Modeling, Hardware, Software, System Reliability Models.

ABSTRACT

The need for a combined hardware/software reliability model is increasing day by day; the need is due, among other factors, to increasing software-induced hardware failures which endanger system missions. On the other hand, already-developed models have not completely fulfilled this need. In this paper, the relevant literature is reviewed and a classification scheme for reliability models based upon model requirements, users' objectives and other related criteria is given. Some existing models are classified according to this scheme. Finally, from a macro point of view, the statement of the combined system reliability problem is approached, several areas of conflict are identified and some general conclusions are presented which may help to provide a better understanding of the problem and its eventual solution.

1.0 INTRODUCTION

Recently, the problem of searching for a combined hardware/software reliability model has been increasing in importance for both model users and model builders. Examples of this can be seen in the RADC solicitation "Impact of Hardware/Software Faults on System Reliability" (September 1982) in the appearance for the first time of an independent session on combined hardware/software reliability during the 1983 Reliability and Maintainability Symposium (ROME83), and in the workshop dedicated to hardware/software reliability entitled "Total Systems Reliability Symposium" to be held in December 1983.

Several reasons may be offered for this increased importance. Systems are becoming so large and complex and time constraints so tight that their operation is possible only through the extensive use of software. The fact that software-induced hardware failures are beginning to appear more frequently as a cause of severe system failures (HOWE83) stresses the need for the development of these combined system reliability models. Also, it is very difficult if not impossible to study these different types of failures and to draw valid inferences by looking at the two subsystems separately.

Why, then, if there is such a need and if several very capable researchers in the field have been working on this subject does not a widely accepted combined model exist? Also, why is it that the existing combined models do not fulfill the pressing needs of the potential model users? The present paper addresses this problem through a review of the literature concerning hardware/software reliability modeling. The problem chronology will be followed and several developmental phases revisited from a macro point of view. At the end some general conclusions will be tentatively derived. With this macro approach to the combined hardware/software problem and the systematic presentation of model inputs and outputs, we expect to highlight some critical aspects that have hindered the appearance of a generally acceptable model. Through this approach a better understanding of the problems involved may be obtained. This may help to channel research efforts and resources in a direction which leads eventually to a satisfactory solution.

For the purpose of avoiding ambiguities, system will be defined as an entity having the following characteristics: its components i) are organized, ii) are interdependent and interacting, iii) pursue an objective or mission, and iv) have some type of feedback mechanism (TRAC71).

2.0 BACKGROUND

In order to perform an efficient and effective system study and to provide a sound basis for model selection, a series of questions were raised before starting the literature review:

- What kind of model is desirable to build? Is it simple? How simple? Is it complex? How complex? Is it parametric? Is it non-parametric?
- To whom will this model be addressed? Will it serve the system designer? Will it serve the acquisition manager? Will it serve the system developer? Will it serve the field engineer?
- What objective will this model serve? Will it be a predictive model? Will it predict reliability? Will it predict time to a given reliability level (reliability growth)? Will it be a control model? Will it serve for reliability apportionment? Will it serve for comparing the effectiveness of different designs? Will it serve for theoretically studying the system?
- What kind of data does it need? Does it require time between failures? Or number of failures per time interval? Or knowledge about the systems parts and configuration? Does the required data exist? Is it possible to obtain the data in the future?
- Does this model exist? Where are the basics developed? What are the basic skills required for developing this type of model? What purpose will it serve? Does the capability exist to develop it today? Does it need more resources for this effort? What resources?
- What profit will be obtained from the model or models? Which type has the largest number of potential users? What is the cost of developing these models? How much will they save the potential users? What type of models are currently being developed? How? Is it

more effective to explore new avenues rather than to extend existing models?

The literature search was organized and implemented with the objective of obtaining enough elements to answer some of the above questions and to provide criteria for an effective modeling strategy. The approach taken has been to study the reliability modeling activity from its origins and by following its development to detect some of the characteristics, pitfalls and trends that may provide answers to these questions.

3.0 CHARACTERISTICS OF THE LITERATURE

If there is a common feature for the literature reviewed through this effort, it is its extensiveness and the diversity of criteria and approaches found.

Early developments concerned hardware reliability, and there are several well-documented models with different objectives and users in mind. Since this was the original field, enough time has elapsed to settle some points, and the divergence of criteria is not so pronounced ((MIL-HDBK-217D), (HUGH75)).

As software became increasingly complex and expensive, software reliability models started to appear and every year there appear new (and not validated) models (SCHA79). There is large divergence with respect to these models and even with respect to what variables and what units should be used in them ((IEEE79), (MIYA78), (DUVA80), (THAY76), (BROO77), (RAMA82), (MUSA75)). It is helpful to note that many of today's software engineers and researchers were orginally trained as electrical engineers. Many of the reliability and complexity models and many of the underlying hypotheses and concepts of software are naturally an extension or adaptation of those found in hardware. This area is entering a new phase with the concept of system modeling as opposed to separate hardware or software modeling; even at this early stage there already exist a number of system models and criteria, and others are in the process of being developed ((HUGH82), (GOEL81b)).

Under the circumstances stated above we adopted the following approach:

- o Classify the literature into homogeneous groups and subgroups
- o Extract from each of these groups the common features which characterize them
- o Between each different classification group, extract the relevant features that differentiate them
- o Study these common and different characteristics to try to derive some general consequences

This classification scheme and critical review approach comprised the backbone of the literature search and problem study and provided the tool with which to obtain the salient features as well as the advantages and disadvantages of each model type from a macro point of view.

4.0 CLASSIFICATION SCHEME

As stated in the previous section the first division considered will be the natural evolution of these models, i.e., hardware, software, and combined (system) reliability models. At the next level the origin of the model will be considered, i.e., if the model developed was through a theoretical approach, an empirical approach or simply by a simulation.

The approach will be considered theoretical if some inherent feature of the problem determines the model (i.e., if it is determined that the failure rate has a given functional form) and the data are used to fit the model and estimate its parameters. An example is the Lloyd-Lipow model of hardware reliability (HUGH75).

The approach will be considered empirical if the data are fitted conveniently. An example is the Duane reliability growth model (DUAN64). Frequently, when an empirical fit like Duane's yields a consistently good result, there may be an underlying theoretical model. As it happened with this model, later work by Crow and Finkelstein showed that the failure process followed a non-homogeneous Poisson process with Weibull intensity ((HUGH75), (LEWI68)).

The approach will be considered a simulation model when the system is studied and modeled through a computer simulation program and its characteristics and performance measures studied by letting this simulation program run. An example is provided by the SPARCS-2 system simulator ((SPAR78), (TIGER82), and (HART80)).

In addition to their origins, the models will be classified according to their use. To this effect they will be divided in three classes: models to study and describe, models to forecast and models to control.

Models to study and describe are very general, include many variables of different kinds, and are in some instances highly theoretical. They may be simplified after the study phase is completed and eventually become classified by one of the other two uses.

Models to forecast will be considered here as those which can predict reliability growth as a function of time. These types of models are valuable for quality assurance engineers or configuration managers to estimate the time and effort necessary to release a product at a pre-determined reliability level. These models are not useful in the design phase ((HUGH75), (MANN74)).

Models to control (design) will be considered here as those which may estimate reliability as a function of the system components and operating characteristics. These types of models are of value to system designers who need to evaluate different design alternatives. Models to control are more useful in the design rather than the development, testing or operating phases ((KALB80), (MIL-HDBK-217)).

Also, as a function of their data requirements (i.e., the type of variables included), models will be classified as time series, structure and component, according to the information that is required in order to develop and use the model in question.

A model will be classified as time series if only the failure times or the number of failures in given

time intervals are required. Reliability growth models are usually of this type.

A model will be classified as component if in addition to the failure information it is necessary to have knowledge about the number and type of system components and its operating characteristics. Models that reflect complexity through a count of their integrating parts are usually of this type.

Finally, a model will be classified as structural if an intimate knowledge of the system and how it operates is necessary. This knowledge is usually used either to model the system as a function of its states and the transition rates between states (Hughes, Goel-Soenjoto) or to simulate its function (as in the Tiger or SPARCS-2 simulator programs).

A classification tree follows (see Figures 1, 2 and Table 1).

TABLE 1: EXAMPLES OF MODELS ACCORDING TO ORIGIN AND OBJECTIVES

	STUDY	CONTROL	FORECAST
THEORETICAL	COX	COX	IBM
EMPIRICAL	--	MIL-HDBK-217	DUANE
SIMULATION	SYSTEM	SYSTEM	NETWORK

5.0 MODEL REQUIREMENTS AND MODEL USES

A model will be useful to the extent that it fulfiles the user's needs. A perfectly valid and fruitful model for one user may be totally worthless for another. Hence, great importance must be placed upon determining beforehand the user population and its needs that the selected model will address. On the other hand, a model will be totally determined by the type of data (or knowledge) the model builder is able to get, in the form:

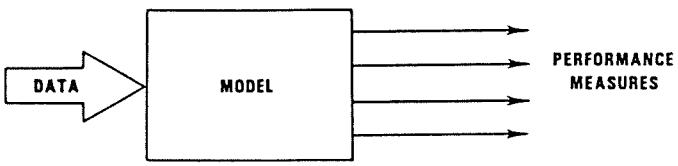

FIGURE 3: MODEL FLOW

The dilemma lies therefore in the fact that for one model user the performance measure of interest will be what time the system will attain a specified reliability level; for another, the measure of interest will be what the reliability level obtained so far is; for yet another, it will be how much the reliability will improve/decrease by changing the design in a given way. With this in mind and with enough information about the process, a model builder may be able to fulfill the user's expectations. It would be useful now to examine the trade-off between requirements and information a potential model user needs in order to evaluate and select a reliability model.

5.1 Inputs or Requirements

For the time series models, the only data requirements are the failure history, either in time between failures or in number of failures by time intervals, and enough knowledge of the system to allow the correct selection of a model among the existing ones. For combined systems the problem lies in justifying the failure process in this new setting.

For the component class, besides the failure history it is necessary to know the types and quantities of components, the environments of operation and the interfaces. This is a particularly difficult situation in combined hardware/software systems. Component models have been extensively used in hardware reliability (MIL-HDBK-217). There is still a great deal of disagreement in software reliability due to variable definition (BASI80). Some researchers use lines of code (LOC) as an indicator of software complexity. Others use the number of operators and operands (Halstead), the cyclomatic complexity (McCabe), the number of ifs, go-tos, etc. (THAY76), as variables that correlate highly with software complexity, and that can be used to model this abstract entity ((THAY76), (BROO77)). The problem lies in coming up with a model whose variables may not be recognized or accepted by the software community or cannot be accurately measured (ROME82) or will not be general (even the concept of lines of code depends upon the language, i.e., a particular higher-order language vs machine language). Moreover, to construct these types of models a large variety of systems have to be examined to provide variability within the model's variables. It is suggested as a rule of thumb that 10 data points for every model variable should be obtained. It may be possible that there will not be enough system data available 1) that variables will be ill-defined on some of the available systems, 2) that not all systems function at the same level 3) that there is an ill-defined common time frame. All of these contribute to poorly defined and weak measurement units.

In addition there is a new concept, firmware which is neither totally hardware nor software (FIRM81). Even more, the problem of how to include the effects of non-physical interfaces between hardware and software in the model constitutes a very challenging difficulty for the system models.

The existing situation with the human element is worst of all. Some of these systems are real time, and human intervention is present. When the system is considered as an integration of three interrelated subsystems, hardware/software/human, the modeling process becomes even more difficult. A model which excludes the effect of human intervention may be of questionable accuracy even though simpler. As can be seen, a part count or component system reliability modeling activity is no easy task.

Structural models require the most knowledge about the system and the largest quantity of data for model building. It is necessary to know and understand the system structure in order to determine its possible states and transitions. It is necessary to have enough level of detail in the data to estimate the distribution of the transition times from state to state and to estimate the corresponding parameters of these distributions. It is necessary to have enough data to validate the model constructed, whether it be a theoretical or a simulation model, before using it as a working tool (SARG81). As can be seen, the cost (in time and effort) invested in acquiring the data, in model building, and in model validating is very large and probably the largest of the three types of models discussed here.

These three types of models have been considered here in increasing order of development effort. We

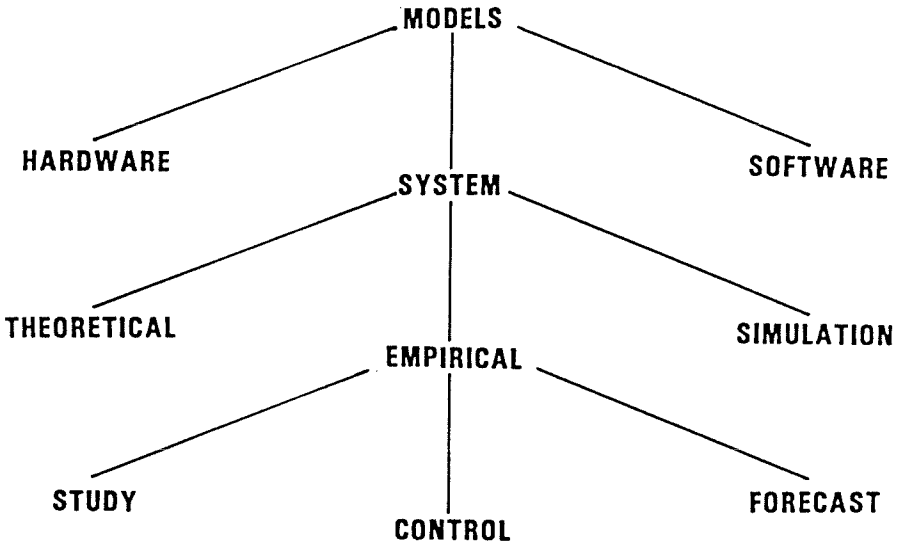

FIGURE 1: HIERARCHICAL RELIABILITY MODELS CLASS SCHEME

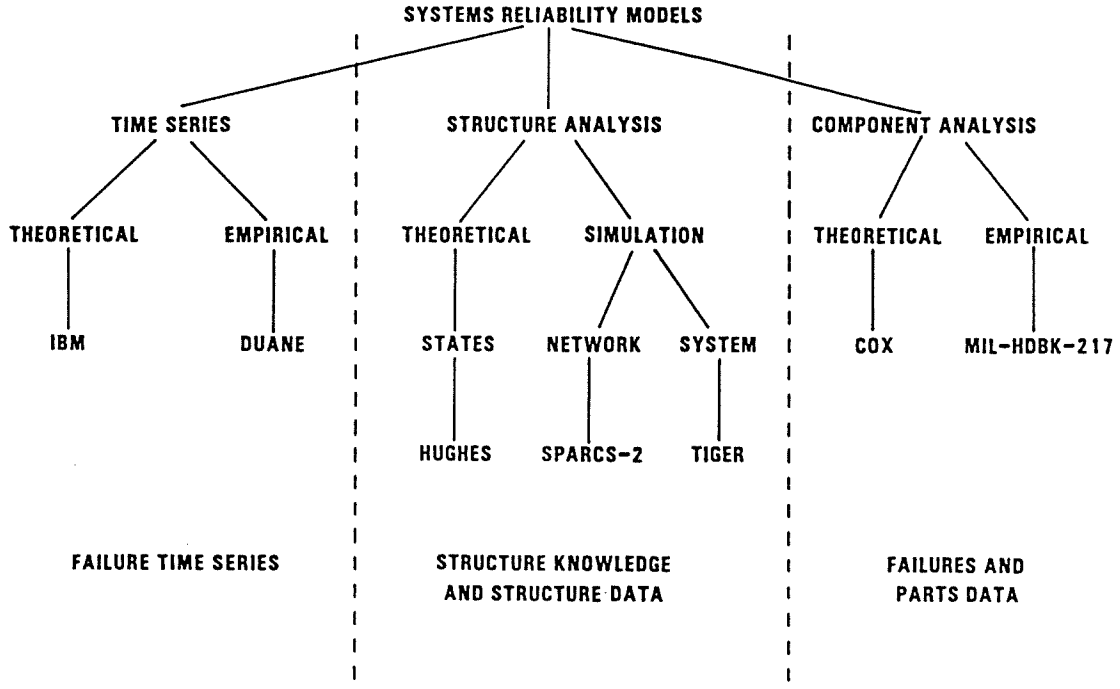

FIGURE 2: CLASSIFICATION EXAMPLES WITH DATA REQUIREMENTS

will now consider the performance measures obtained from each of them and the types of potential audiences that would be interested in using them. In addition, several of their advantages and disadvantages as perceived at this stage will also be examined.

5.2 Model Outputs and Advantages

The class of existing models based upon time series analyses of the failure history are probably less expensive to implement since they are less complex. They are only time-dependent and therefore require less data from the user and are easier for him to understand. There still remains the problem of their applicability to the combined system environment. Once a specific model is selected and the data fitted, the performance measures obtained include:

- MTTF or failure rate at a given time (instantaneous or cumulative)
- reliability level for a specified mission time
- expected time for the reliability growth process to achieve a given level

These models are not directly useful for design because the required data are available only at a later phase of development. Also, they are not directly useful for comparison of alternative designs before their implementation. They are very simple to carry out, enabling the user to do it by himself.

The class of models based upon component analyses are more complex but comparably not very expensive to develop if the data exist and the variables are well-defined. Once a model is developed for a class of systems it becomes general and can be used for any particular system of this class (MIL-HDBK-217). This feature makes it particularly attractive for system designers and evaluators since they can perform sensitivity analysis on different alternatives. In the present situation these type of models present very attractive features for combined hardware/software systems if the variables representing software interfaces can be well-defined and measured.

The models based upon the structure of the system are the most complex, expensive and time-consuming ones but at the same time are the ones that provide the largest amount of information, since it is possible to obtain among other performance measures:

- steady state results and transient state results for:
  - reliability
  - availability
  - MTTF or failure rate
  - maintainability
- bottleneck and load analyses
- sensitivity analysis
- experimentation

The models may be individual (i.e., every system may be modelled uniquely) or general (for a wide class of systems it can become a working approximation for all members of the class), offsetting in this way its large cost by being simultaneously usable in many environments (i.e., the Tiger and SPARCS simulation models or Hughes and Goel-Soenjoto stochastic models may be used to study the performance of large classes of systems).

6.0 LITERATURE SEARCH CONCLUSIONS

The results of the literature survey provided an overview of the problems involved in studying and developing a reliability model of the general type for combined hardware/software systems.

The approach taken to state the problem has been to classify the existing literature according to model development through time (hardware, software and system), according to origin (theoretical, empirical and simulation), by users' needs (study and analysis, design and control) and according to data requirements (time series, component and structure).

After the class structure was defined, the existing models were classified accordingly. From here, the similarities and contrasts between the different classes were established in order to isolate and discuss the characteristics, advantages and disadvantages of each of these classes. A summary of these differences may be seen in Table 2.

MODEL CLASS	ADVANTAGES	DISADVANTAGES
TIME SERIES ANALYSIS	LESS EXPENSIVE, LESS COMPLEX, REQUIRE LESS DATA, EASIER TO UNDERSTAND, EASIER TO IMPLEMENT.	NOT FOR DESIGN, USED IN LATER PHASES, INDIVIDUAL CASES, LESS INFORMATION.
SYSTEM STRUCTURE ANALYSIS	MOST COMPLETE INFORMATION, (RELIABILITY, MAINTAINABILITY, AVAILABILITY, SENSITIVITY, BOTTLENECKS, TRANSIENT & STEADY STATE ANALYSES), EXPERIMENTATION, DESIGN PHASE.	EXPENSIVE, TIME-CONSUMING, COMPLEX, THEORETICAL, INDIVIDUAL CASE, DATA { ESTIMATION / VALIDATION
SYSTEM COMPONENT ANALYSIS	DESIGN PHASE, SENSITIVITY ANALYSIS, INTUITIVE, GENERAL, LESS EXPENSIVE.	NOT SIMPLE, INTERFACES, VARIABLE DEFINITION, CONFIDENCE INTERVAL, LARGE DATASET.

TABLE 2: GENERAL CHARACTERISTICS OF MODELS BY CLASS GROUPS

Finally, the following reasons for not yet having a combined system model are proposed:

- for the time series model: the problem of justifying the extension of existing theoretical or empirical models to the combined system environment
- for the component and structure models: the problem of definition of variables/structures that take into consideration the non-physical hardware/software interaction and human intervention
- for all classes, the definition of measurement units for the analysis variables

A better understanding of the problem may help in the development of a solution.

7.0 ACKNOWLEDGEMENTS

The first author is indebted to Professor Amrit L. Goel of Syracuse University for introducing him to the area of Software Reliability.

REFERENCES

(BASI80) Basili, V., "Tutorial on Models and Metrics for Software Management and Engineering." IEEE catalog EHO-167-7.

(BROO77) Brooks, W.D., and R. Motley, "Statistical Prediction of Programming Errors." RADC-TR-77-175, May 1977.

(DUAN64) Duane, J.T., "Learning Curve Approach to Reliability Monitoring." IEEE Transactions on Aerospace, Volume 2, 1964.

(DUVA80) Duvall, L., and S. Gloss-Soler, "Data Needs for Software Engineering." Proceedings of the National Electronics Conference, 1980, Volume 34, pp. 473-477.

(FIRM81) Giloi, W.K., R. Gueth and B.R. Shriver, "Firmware Engineering: Methods and Tools for Firmware Specification and Design." Proceedings of AFIPS, 1981, Volume 50, pp. 49-55.

(GOEL81b) Goel, A.L., and J. Soenjoto, "Models for Hardware/Software System Operational Performance Evaluation." IEEE Transactions on Reliability, Volume R-30, August 1981, pp. 232-239.

(HART80) Harter, H.L., A.H. Moore, and R.C. Snead, "Comparisons of Monte Carlo Techniques for Obtaining System Reliability Confidence Limits." IEEE-Transactions on Reliability, Volume R-29, No. 4, October 1980.

(HOWE83) Howell, J.M., "A Software Evaluation: Results and Recommendations." 1983 Proceedings of Reliability and Maintainability Symposium (RAMS).

(HUGH75) Hughes Aircraft Co., "Reliability Growth Study." RADC-TR-75-253.

(HUGH82) Augus, J. and L.E. James, Hughes Aircraft Co., "Combined Hardware/Software Reliability Models." 1982 Proceedings of Reliability and Maintainability Symposium (RAMS).

(IEEE79) Special Issue on Software Reliability. IEEE Transactions on Reliability, Vol. R-28 No. 3, August 1979, pp 178-272.

(KALB80) Kalbfleisch, J., and R. Prentice, The Statistical Analysis of Failure Time Data. Wiley, 1980.

(LEWI68) Cox, D.R., and P.A.W. Lewis, The Statistical Analysis of Series of Events, London, Methuen & Co, Ltd, 1968.

(MANN74) Mann, Schafer, and Singpurwalla, Methods for Statistical Analysis of Reliability & Life Data. Wiley, 1974.

(MIL-HDBK-217) Reliability Prediction of Electronic Equipment, MIL-HDBK-217D, Notice 1, June 1983.

(MIYA78) Miyamoto, I., "Towards an Effective Software Reliability Evaluation." Third International Conference on Software Engineering, May 1978, pp. 46-55.

(MUSA75) Musa, J.D., "A Theory of Software Reliability and its Application." IEEE Transactions on Software Engineering, Vol. SE1 No. 3, September 1975, pp. 312-327.

(RAMA82) Ramamoorthy, C.V., and F.B. Bastiani, "Software Reliability-Status and Perspectives." IEEE Transactions on Software Engineering, Vol. 8 No. 4, July 1982, pp 354-371.

(ROME82) Romeu, J.L., and C. Turner, "Parametric vs Non Parametric Techniques in the Analysis of Software Engineering Data." DACS, December 1982.

(ROME83) Romeu, J.L., "RAMS83 From a Combined Hardware/Software System Perspective." DACS Bulletin, Vol. III, No. 2, 1983.

(SARG81) Sargent, R., and O. Balci, "Validation of Simulation Models." Winter Simulation Conference, San Diego, CA, December 1979.

(SCHA79) Schafer, R.E., et al., Validation of Software Reliability Models. RADC-TR-79-147, June 1979.

(SPAR78) Locks, M.O., "Monte Carlo Bayesian System Reliability and MTBF Confidence Assessment (SPARCS-2 Simulation Program)." Oklahoma State University, AFFDL-TR-78-18, 1978.

(THAY76) Thayer, et al., "Software Reliability Study." RADC-TR-76-238, August 1976.

(TIGER82) Luetjen, P., and P. Hartman, "Simulation with the Restricted Erlang Distribution (RED)." Reliability and Maintainability Symposium (RAMS) 1982 pp. 233-237.

(TRAC71) Tracey, W., Designing Training and Development Systems. American Management Association Inc., 1971.

JORGE LUIS ROMEU
IIT RESEARCH INSTITUTE
199 Liberty Plaza
Rome, New York 13440

Mr. Romeu is a Research Engineer at IIT Research Institute where he performs mathematical and statistical analysis on software and hardware data. Previous to his employment by IITRI Mr. Romeu was a Teaching/Research Assistant at Syracuse University for two years where he taught regression and data analysis courses and performed research on reliability problems. In his native country, Mr. Romeu worked for eight years as statistician/applied mathematician for the petrochemical, construction and agricultural industries. He holds degrees in Mathematical Statistics and in Operations Research and is currently working on his Ph.D. in Operations Research at Syracuse University. He is an elected Fellow of the Institute of Statisticians, and a member of the American Statistical Association and the Society of Reliability Engineers. Mr. Romeu has published several articles and papers describing the solution of applied research problems through multivariate analysis, nonparametric statistics and mathematical modeling techniques.

KIERON A. DEY
IIT RESEARCH INSTITUTE
199 Liberty Plaza
Rome, New York 13440

Mr. Dey graduated from the University of Reading, England, in 1974 with a B.Sc. in Mathematics/Statistics. He worked for two years in a statistical service in London and was concerned with experimental design, data analysis and reliability studies. He moved to Marconi-Avionics in 1976, concentrating on reliability studies for signal processing systems. Since joining IIT Research Institute in January 1981 as statistical advisor to the Reliability Analysis Center (RAC), he has applied statistical/operations research methods to a variety of studies and developed non-parametric or Monte Carlo solutions to non-standard data structures. He also lectures regularly on a training course for government personnel and teaches experimental design at local colleges. He manages the engineering group at RAC.

Mr. Dey is a Fellow of the Royal Statistical Society and a Member of the American Statistical Association.