NUREG/CR-1381 SAND80-7028 Unlimited Release

A Methodology for Evaluating Safeguards Capabilities for Licensed Nuclear Facilities - Final Report

Harold A. Bennett, M. Teresa Olascoaga, Sandia National Lal ratories Alan Sicherman, Gary Smith, Woodward-Clyde Consultants

Printed March 1980

Sandia National Laboratories

F 2900-Q(3-80)

Prepared for

U. S. NUCLEAR REGULATORY COMMISSION

THIS DOCUMENT CONTAINS POOR QUALITY PAGES

NOTICE

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, or any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use, or the results of such use, of any information, apparatus, product or process disclosed in this report, or represents that its use by such third party would not infringe privately owned rights.

The views expressed in this report are not necessarily those of the U. S. Nuclear Regulatory Commission

Available from

GPO Sales Program

Division of Technical Information and Document Control U.S. Nuclear Regulatory Commission Washington, D.C. 20555

and

National Technical Information Service Springfield, Virginia 22161 NUREG/CR-1381 SAND80-7 28 Unlimited kelease

A METHODOLOGY FOR EVALUATING SAFEGUARDS CAPABILITIES FOR LICENSED NUCLEAR FACILITIES - FINAL REPORT

.

H. A. Bennett, M. T. Olascoaga Sandia National Laboratories Albuquerque, NM 87185

A. Sicherman, G. Smith Woodward-Clyde Consultants San Francisco, CA 94111

Date Published: March 1980

Submitted by Woodward-Clyde Consultants San Francisco, CA 94111 under Contract Document No. 13-2215 to Sandia National Laboratories Albuquerque, New Mexico 87185 operated by Sandia Corporation for the U.S. Department of Energy

Prepared for Division of Safeguards Office of Nuclear Material Safety and Safeguards U.S. Nuclear Regulatory Commission Washington, DC 20555 Under Interagency Agreement DOE 40-550-75 NRC FIN No. All53

ABSTRACT

This report describes work performed by Woodward-Clyde Consultants under contract to Sandia Laboratories for assistance in the development and implementation of an evaluation methodology. This methodology was developed to aid the NRC in its evaluation of fixed-site physical protection system performance relative to the Physical Protection Upgrade Rule, 10 CFR Part 73.45

TABLE OF CONTENTS

Section			Page	
1.0	EXEC	UTIVE SUMMARY	1-1	
	1.1	Tasks	1-2	
	1.2	Summary	1-8	
	1.3	Outline of Report	1-11	
2.0	TECHNICAL SUMMARY OF EVALUATION METHODOLOGY			
	2.1	Introduction	2-1	
	2.2	Component Evaluation	2-6	
	2.3	Low Level System Task (Performance Characteristic)	2-14	
		Evaluation		
	2.4	Higher Level Aggregation, Delay/Response and	2-15	
		Multiple Access Points		
	2.5	Capabilities Evaluation	2-18	
3.0	IMPLEMENTATION GUIDELINES		3-1	
	3.1	Introduction	3-1	
	3.2	Development of Hierarchies	3-1	
	3.3	Development of Questionnaires	3-3	
	3.4	Assessing Weights and Agrregation Rules:	3-5	
		Introduction		
	3.5	Assessing Weights and Aggregation Rules	3-6	
		for Questions on Component Questionnaires		
	3 6	Interpretation of Component Scores and Ag-	3-12	
		gregation Parameters		
	3.7	Assessing Aggregation Rules for Higher Level	3-14	
		Elements of the Hierarchy		

Section			Page
4.0	SAFEGU	ARDS EVALUATION COMPUTER PROGRAM	4-1
		ntroduction	4-1
		se of the Evaluation Program kample of Algorithm and Computer Program Use	4-2 4-14
5.0	SUMMAR	Y AND RECOMMENDATIONS FOR FUTURE WORK	5-1
	5.2 L:	trengths of the Methodology imitations of the Methodology ecommendations for Future Work	5-1 5-3 5-4
6.0	GLOSSA	RY	6-1
APPE	NDIX Al	FUNCTIONAL FORM FOR SAFEGUARDS EVALUATION ALGORITHM	A1-1
APPE	NDIX A2	ASSESSING ALGORITHM PARAMETERS	A2-1
APPE	NDIX A3	COMPUTER PROGRAM DOCUMENTATION	A3-1
APPE	NDIX A4	DATA FOR EVALUATION EXAMPLE	A4-1

ii

		LIST OF TABLES
Table	<u>a</u>	Page
2-1	COMPONENT SCORING FORMULAS	2-13
4-1	DATA BASE DEFINITIONS	4-8
A3-1	PROGRAM ROUTINES AND DATA FILES	A3-3

LIST OF FIGURES

Figu	re	Page
1-1	HIERARCHY FOR SAFEGUARDS EVALUATION	1-3
2-1	COMPONENT SELECTION MATRIX EXAMPLE	2-2
2-2	ADVERSARY PATH	2-21
3-1	SYSTEM QUESTIONNAIRE EXAMPLE: MULTIPLE SENSORS	3-16
3-2	EXAMPLE QUESTION TO AID IN ASSESSING A DELAY- RESPONSE TYPE SCORE	3-17
4-1	MNEMONICS FOR ALGORITHM EXAMPLE	4-16
4-2	READING IN THE HIERARCHY STRUCTURE	4-17
4-3	COMPUTING QUESTIONNAIRE SCORES	4-18
4-4	DISPLAYING QUESTIONNAIRES	4-19
4-5	HIERARCHY DISPLAY	4-20
4-6	HIERARCHY GRAPHICAL DISPLAY	4-22
4-7	AGGREGATION EXAMPLE	4-24
6-1	DISAGGREGATION STRUCTURE	6-2

1.0 EXECUTIVE SUMMARY

Sandia Laboratories contracted with Woodward-Clyde Consultants (WCC) to assist in developing a methodology for evaluating safeguards capabilities at licensed nuclear facilities. The total effort was divided into two phases. Phase I was concerned primarily with the development of a preliminary evaluation algorithm. The results of this effort were described in [1]. Phase II which is reported here was devoted to completing and refining the algorithm. The evaluation m thodology is to aid in the implementation of new NRC regulations (the Physical Protection Upgrade Rule), which are designed to upgrade the physical security of fuel cycle facilities. The methodology could also be used to provide guidance to licensees in meeting the safeguard system capability requirements.

Five performance capabilities¹ of physical protection systems were specified by the NRC in the Physical Protection Upgrade Rule, 10 CFR Part 73.45, paragraphs (b) - (f):

- Prevent unauthorized access of persons and materials into Material Access Areas (MAAs) and Vital Areas (VAs).
- Permit only authorized activities and conditions within Protected Areas (PAs), MAAs, and VAs.

¹Throughout this report, certain terms are used that have a particular meaning in the context of the evaluation procedures. To avoid confusion, these terms are defined in Section 6.0 (glossary).

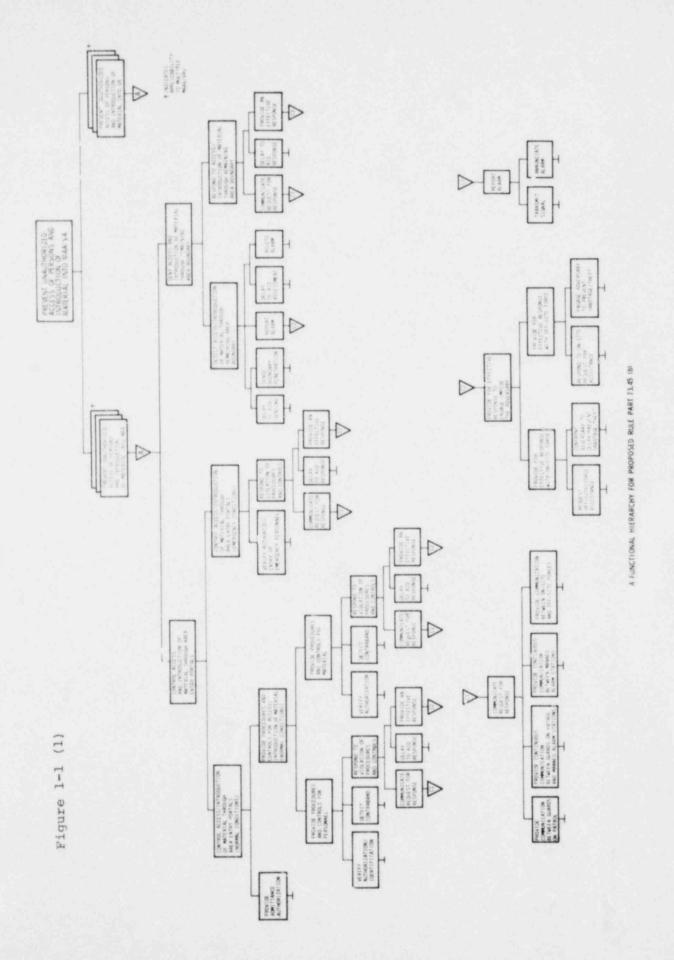
- Permit only authorized placement and movement of Strategic Special Nuclear Material (SSNM) within MAAs.
- Permit removal of only authorized and confirmed forms and amounts of SSNM from MAAs.
- Provide for authorized access and assure detection of and response to unauthorized penetration of the PA.

To evaluate these five capabilities in a logical manner, the disaggregation structure shown in Figures 1-1 (1) through 1-1 (5) was developed by Sandia Laboratories with the cooperation of the NRC. WCC was to assist in formulating an algorithm by which individual safeguard system component (equipment and/or procedure) assessments could be combined into a meaningful score or series of scores indicating the total adequacy of safeguards at a particular facility.

1.1 TASKS

Four tasks were specified by Sandia Laboratories for Phase II.

- Task 1: Assist Sandia personnel in defining the remaining performance characteristics not defined in Phase I.
- Task 2: Assist Sandia in developing the remaining effectiveness test questionnaires (for assessing particular components) not defined in Phase I.
- Task 3: Assist Sandia personnel in developing the remaining component combination scoring rules not developed in Phase I for jobs identified in single performance characteristics.



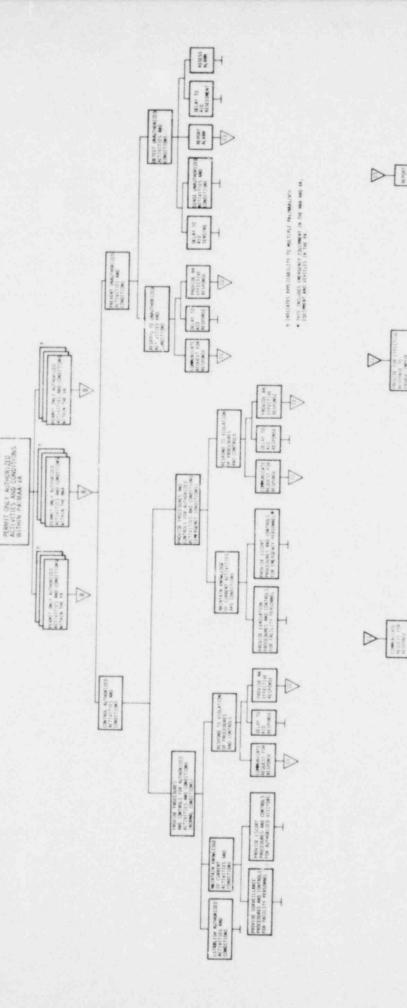


Figure 1-1 (2)



(M24/00) (M24/00) (M24/00) (M24/00) (M24/00)

NUMER INC.

ANAL ST

ACCOUNT OF ACCOUNT OF

ALL REAL

REPUTATION COMMICATION INTERNATION INTERNA

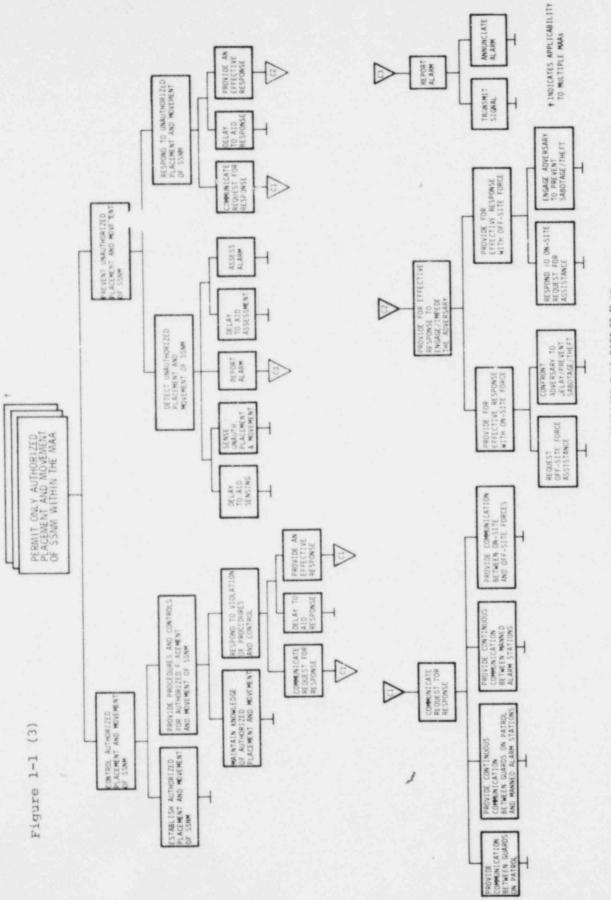
NULLY AND STATE

-

CONTRACTOR INCOME

CTACK CATON

NU POINT



-

.

A FUNCTIONAL HIERARCHY FOR PROPOSED RULE PART 73.45 (d)

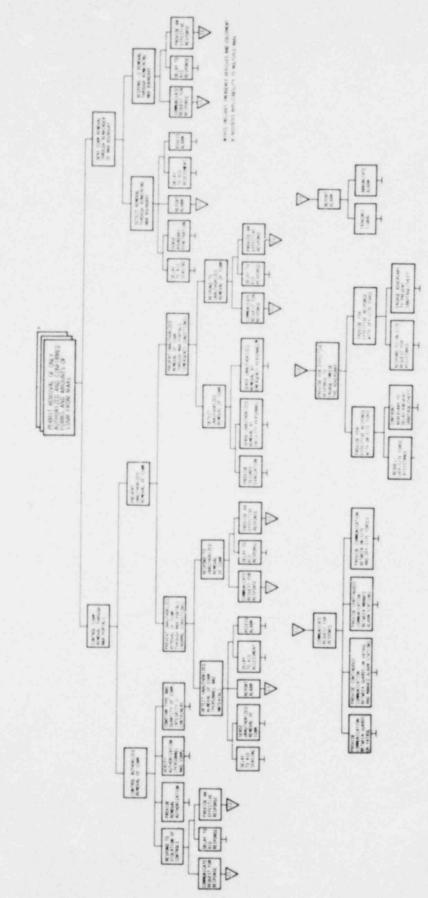
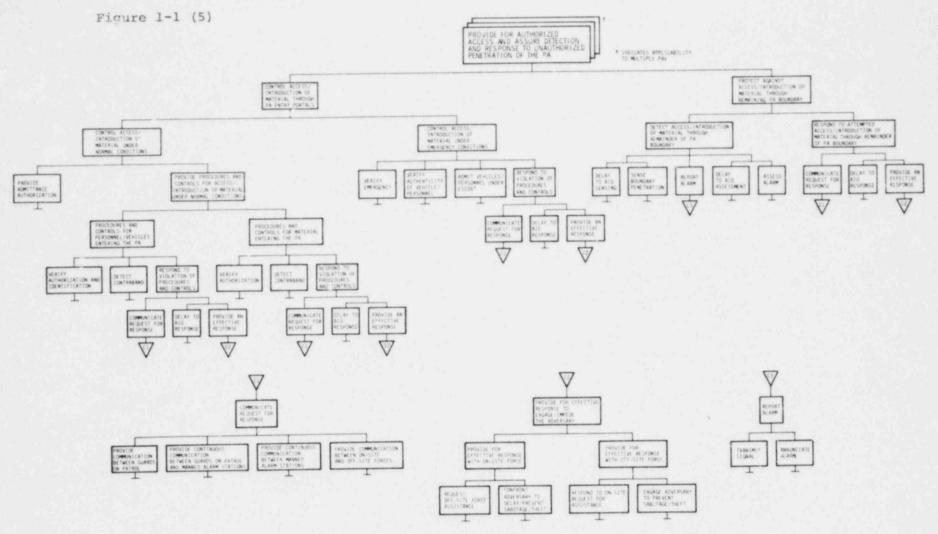




Figure 1-1 (4)



. .

 $\mathbf{\hat{x}}$.

A FUNCTIONAL HIERARCHY FOR PROPOSED RULE PART 73.45 (1)

1-7

1

.

Task 4: Assist Sandia personnel in developing the aggregation algorithm in detail and provide an illustration of its implementation using hypothetical data provided by Sandia.

An ANSI standard FORTRAN computer program which would exercise the aggregation algorithm was to be provided.

WCC was to furnish a comprehensive document describing the algorithm and its development. Detailed guidelines for implementing the algorithm were to be provided along with documentation and instructions on the use of the computer program.

1.2 SUMMARY

The work done by WCC focused on the following areas:

- Providing an algorithm for combining component questionnaire responses to obtain an overall component effectiveness score.
- Providing an algorithm for combining component effectiveness scores to obtain an overall score for the performance characteristic for which the components were selected.
- Providing an algorithm for combining delay/response type scores.
- Providing an algorithm for combining the higher levels of the hierarchy for any particular performance capability.
- Providing a computer program for synthesizing these algorithms to evaluate portions of or an entire performance capability.

- Providing implementation guidelines for the entire methodology.
- · Providing recommendations for future work.

Each of these areas is now briefly summarized.

1. Aggregating Component Questionnaire Responses

A methodology for aggregating questionnaire responses has been developed (primarily in Phase I) that is logical and defensible and yet practical to implement. The basic aggregation formula has a theoretical basis in both utility and probability theory, thus aiding defensibility. The evaluator's task is simplified by only requiring responses to multiple choice questions that concern the description of the component. The methodology allows for individual weighting of questions to reflect their relative importance and also for varying types of interaction (e.g., non-additive, additive) among question responses to yield an overall score for any individual component. The basic formula can also be used at higher levels of the evaluation hierarchy.

2. Aggregating Components to Evaluate Performance Characteristics

A methodology that indicates how several components coordinate with each other in addressing a performance characteristic has been developed. An evaluator answers multiple choice questions that determines how the component scores should be combined. The evaluator's responses can reflect facility dependent implementation of components.

3. Aggregating Delay/Response Type Scores

A methodology that allows for the direct comparison of delay times and response, monitoring, or assessment times has been developed to allow meaningful evaluations of delay/response type elements of a safeguard system.

4. Aggregating Higher Levels of the Hierarchy

The same methodology elements developed for the lower levels of the hierarchy are used to complete the evaluation of the hierarchy. Such aspects as multiple access points are addressed in these aggregations.

5. Computer Program Implementation

A computer program that performs the safeguards evaluation computations has been developed. The program takes as input questionnaire and hierarchy formats and the evaluator's responses to the multiple choice questionnaires. The program computes the scores for all components, performance characteristics and higher level elements of a capability hierarchy. It provides for sensitivity analysis on questionnaire weights and responses, and on the interaction of hierarchy elements. The program is interactive and has hierarchy display features.

6. Methodology Implementation Guidelines

Guidelines for developing component questionnaires, and assessing scoring rules and weights have been developed. Data requirements for the algorithm are specified and ways of interpreting evaluation and sensitivity analysis results are described.

7. Recommendations for Future Work

Suggestions for improving the implementation of the algorithm are presented. Ways of extending the algorithm to combine capabilities across facility area boundaries (e.g., PA and MAA) to evaluate an overall safeguards capability are discussed. Limitations of the methodology and general approach are reviewed.

All of these points are discussed in detail in the various sections and appendices of the report.

1.3 OUTLINE OF REPORT

Section 2 of this report contains a technical summary of the general methodology. Section 3 contains a discussion of how the methodology is implemented. Section 4 describes the computer program that performs the computations specified by the algorithm. A single capability for a safeguards facility using hypothetical data is evaluated to provide an illustration of how the algorithm and computer program work. Section 5 contains a critical summary of the methodology and recommendations for future work. The appendices provide supplementary technical information and data used in the computer example. TECHNICAL SUMMARY OF EVALUATION METHODOLOGY

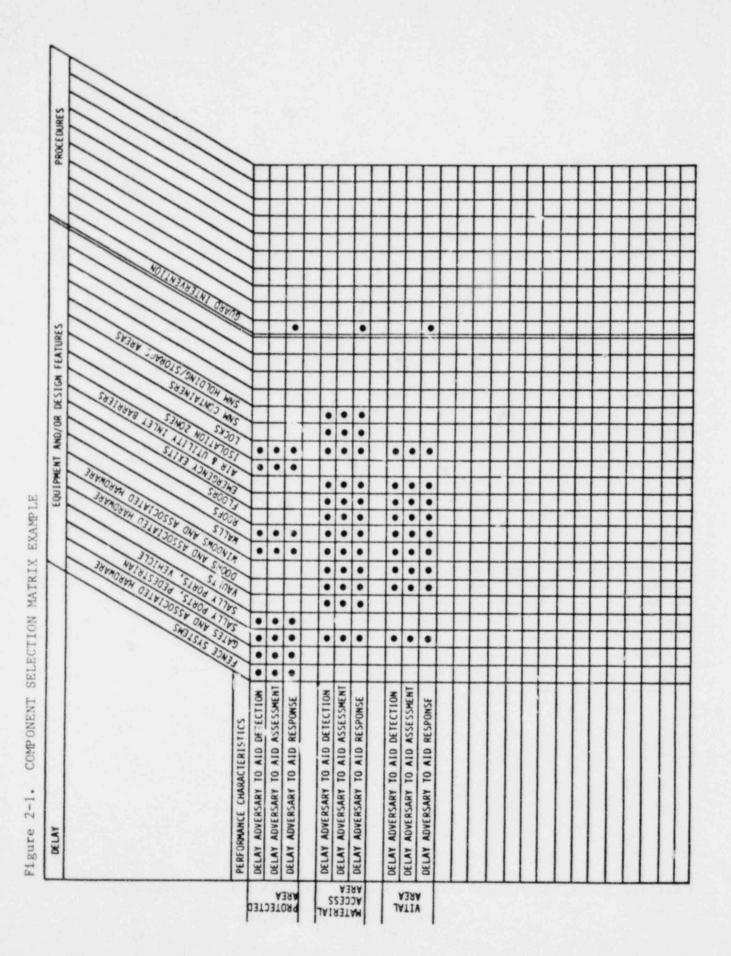
2.0

2.1 INTRODUCTION

In order to address the specifics of the Physical Protection Upgrade Rule the capabilities hierarchy (or disaggregation structure) shown in Figures 1-1(1) through 1-1(5) was developed by the NRC and Sandia Laboratories. Each of the five performance capabilities is treated as a separate objective, with its own independent hierarchy or disaggregation structure.

The upper level of each separate structure is the <u>performance cap-ability</u>, as specified by the NRC Upgrade Rule. These are followed by <u>system functions</u>, which a system must perform in order to meet the specified capability. The major functions are further broken down into system subfunctions, which identify specific tasks to be performed by the system.

Each system subfunction is *c*isaggregated into specific <u>low-level</u> <u>system tasks</u> which form the lowest level of the hierarchy. Performance characteristics relate these low-level system tasks to components. These performance characteristics correspond to the rows of the Component Selection Matrices (Figure 2.1), while the columns of the matrix represent specific components (equipment and procedures). Thus the Component Selection Matrices describe what approaches (equipment and procedures) are acceptable to perform specific tasks (performance characteristics).



WCC's contract required that a set of aggregation schemes or algorithms be developed with which one could work back up the hierarchy to arrive at a single score for each capability. To do this, individual components (the lowest hierarchy level) must be assigned an overall score based on evaluators' responses to questionnaires. In the case of equipment, this single score must reflect such factors as general performance, installation, maintenance, reliability, and vulnerability. For procedures, the score should encompass aspects of craining, vulnerability, and contingencies. All of these issues are addressed in the component questionnaires.

Once each component has received a score, scores for those components that address an individual performance characteristic must be aggregated to arrive at a single score for the appropriate low-level system task via the performance characteristic. Continuing up the hierarchy, scores on low-level system tasks are combined into system subfunction scores, which are then aggregated into system function scores, and finally, into an overall score for each performance capability.

General Aggregation Concepts

To be practical in terms of input requirements, an algorithm needs to operate with input consisting of subjective responses (using descriptive multiple choice response scales) to a large number of questions. Specifically, there are questionnaires for each component or procedure and system questionnaires addressing the interaction between low-level tasks and between components.

Care must be taken when using the responses to these questions. A practical evaluation algorithm cannot require information that is unrealistic to obtain from these questionnaire responses. Because of the large number of questions and their potential modifications, it also should not require a lengthy calibration procedure for each question.

In addition, the range of possible responses to these questions is not uniform, and the questions can differ in relative importance.

Within these practical constraints, an algorithm must still produce meaningful results. The computational rules and assumptions must not be arbitrary. The algorithm should be capable of providing the correct answer where inputs and their interactions are precisely known, since this is a primary means of checking the reasonableness of the algorithm.

Because of the need for meaningful and defensible results, it is desirable to use aggregation schemes based on well-developed methodologies. Two such methodologies are multiattribute utility analysis (decision analysis), and probability analysis. The former provides mechanisms for aggregating multiple criteria into a single overall score, and is particularly useful when subjective considerations are involved. Probability analysis also specifies how to derive a measure for a system in terms of its components and their interactions.

In the safeguards problem, both decision analysis and probability analysis appear desirable. The latter addresses the probability that the system will perform adequately in the event of specific types of adversary actions. This must be the underlying concern when evaluating capabilities and when characterizing interactions between system elements. Individual component successes or failures have different impacts on the probability of total system success or failure depending on system interactions. An algorithm should produce meaningful results if probabilities and system interactions (e.g., fault trees) could be specified. On the other hand, the requirements for practical inputs using subjective questionnaires and responses make it too restrictive to assume that such probabilities can be derived from the input data. The input may be related to probabilities but a direct quantitative relationship cannot be assumed. An alternate approach is to combine

subjective scales using decision analysis preference functions. These do not need to assume a one-to-one correspondence between response scales and probabilities, but rather reflect judgments as to the relative comparison of alternative components and systems. Still, it is desirable that if the probabilities were actually known, the scheme used to combine scores would give the correct results.

The algorithms to be presented in the following paragraphs are based on aggregation rules specified by decision analysis under certain assumptions. The assumptions are made in order to make the implementation procedures as practical as possible, hopefully without omitting any important features of the problem. These assumptions could be relaxed, but more calibration effort would be required. The aggregation models used allow for differential weightings of elements, and can reflect different interactions among elements. The results of the aggregation can be used as part of a logical and consistent judgmental comparison of alternative systems. In addition, the algorithms to be presented yield the correct results when <u>certain assumptions</u> and actual probabilities are used in the computation. Thus the aggregation logic can be interpreted using probability notions as well as preference concepts. The use of both analytical approaches helps to give the algorithm its practical and defensible characteristics.

The algorithms for the specific types of aggregations required to evaluate a capability hierarchy will now be individually discussed. For each case, the algorithm is described in terms of what features can be reflected by the computation. Implementation guidelines, discussed in Section 3 of this report, include further discussion of how certain parameters are assessed and interpreted.

2.2 COMPONENT EVALUATION

The algorithm for evaluating components was developed in Phase I and is discussed in detail in the Phase I report [1]. The discussion here reviews the main assumptions and results.

The response scale for each individual question on a component questionnaire is considered to be an attribute or measure. These measures have been developed with an orientation towards aspects of a component that can hinder its performance. The highest response on a question connotes that the factor being considered will not be compromised because of the particular component. A lower response connotes that the factor will have a certain degree of compromise depending upon the range of the response scale. The methodology allows a scaling parameter or weight to be applied to each question. This weight essentially normalizes response scales whose ranges cause them to differ in relative importance. For implementation practicality, relative preferences over individual response scales are assumed to be linear. We now define the following notation for question "i":

 X_i = unadjusted question response normalized to go between 0 and 1. W_i = weight asigned to reflect total range of the response scale. S_i = adjusted question response or question score.

The formula connecting these three quantities is:

$$S_{i} = 1 - W_{i} (1 - X_{i})$$

where all three quantities are restricted to the range between 0 and 1 as a scaling convention, with 1 being the best.¹ Note that if $X_i = 1$

¹Note that if any other range is used (say 1 to 5) it is trivial to normalize this to the range 0, 1. This also allows direct comparison between questions with differing numbers of possible responses.

then $S_i=1$ and if $X_i=0$, then $S_i=1-W_i$. Thus, only if $W_i=1$, can $S_i=0$. $W_i=.2$, for example, implies a minimum possible score of S_i equal to .8. Thus, a question with little importance with respect to a particular factor can have only a small effect on the score.

The scores for the questions can be thought of as simplified single-attribute utility functions that have been normalized so that they have equal importance. For a particular group of N attributes, we will make the assumption that they are mutually utility independent.² The results of multiattribute utility theory [2] allows us (if certain assumptions are made) to define an overall group score S normalized between 0 and 1 as:

$$s = c \sum_{i=1}^{N} s_{i} + \kappa c^{2} \sum_{i=1}^{N} \sum_{j>i} s_{i} s_{j} + \dots + \kappa^{N-1} c^{N} \prod_{i=1}^{N} s_{i}$$
(1)

(2)

(4)

where $1 + K = (1 + K C)^{N}$

If we define V = KC then

$$s = \frac{1}{K} \begin{bmatrix} v & \sum_{i=1}^{N} s_i + v^2 \sum_{i=1}^{N} \sum_{j>i} s_i s_j + \dots + v^N & \prod_{i=1}^{N} s_i \end{bmatrix}$$
(3)

where $K = (1 + V)^{N} - 1$

This leaves us with only one scaling constant (V) to evaluate.

²See, for example, R.L. Keeney and H. Raiffa, <u>Decisions with Multiple</u> <u>Objectives; Preferences and Value Tradeoffs</u>. New York: Wiley, 1976. Also see Appendix Al.

Before proceeding further, however, we briefly review the results to this point. The necessity of handling large numbers of questions and questionnaires imposes a practicality constraint on the complexity of the aggregation algorithm that can be realistically used. The assumptions leading to the above formula essentially allow us to decompose the problem into considering individual questions somewhat in isolation and then to combine the results. With all the simplified assumptions, the algorithm still reflects the key aspects of the importance range of the questions via the W; and the way in which factors interact via V. Decision analysis enables the use of the formula above to compare different sets of responses. A set with a higher score is preferred to one with a lower score. The consequences of assigning different values of V with respect to both preference and probability interpretations will now be examined. In the discussion to follow, the entire set of questions for a particular component will be referred to as belonging to one group with a single interaction coefficient V. A general case of considering each component questionnaire as a "mini hierarchy" in itself will be discussed in Section 3 under implementation guidelines.

A. V=0. If we take limits as V+O in (3) we get

$$S = \sum_{i=1}^{N} S_{i} / N$$
.

Thus the overall score S is the mean of the individual scores. This is appropriate if an individual score makes the same incremental contribution to component quality regardless of the fixed levels of the other scores. It is also appropriate if the component behaves in a way such that one factor is chosen at random, and the component as a whole succeeds or fails on the basis of the one factor. B. V=-1. If we substitute V=-1 into (3) we get

$$s = 1 - \prod_{i=1}^{N} (1 - S_i)$$
.

This is appropriate if each factor can substitute completely for another factor in order for the component to work. A factor contributes incrementally more when other factors are low (e.g., this one is needed) than when other factors are high (this one is not really needed).

In fault tree theory, this is interpreted as the computation for an \underline{OR} gate. If S_i is the probability that the factor associated with question i works, and the overall component works if any of the individual factors works, then the probability of overall success is given by S.

C. V+ ∞ . If we take limits as V+ ∞ , we get $S = \prod_{i=1}^{N} S_i .$

This is appropriate if a factor contributes incrementally more when other factors are high rather than low, and does not contribute at all if any of the other factor scores is zero.

In fault tree theory, this is as the computation for an <u>AND</u> gate. In this case, every factor must succeed for the overall component to succeed.

D. V=1. If we substitute V=1 into (3) we get

$$S = \frac{1}{2^{N}-1} \left[\sum S_{i} + \sum_{i=1}^{N} \sum_{j \ge i} S_{i}S_{j} + \dots + \prod_{i=1}^{N} S_{i} \right]$$
(5)
$$= \frac{1}{2^{N}-1} \left[\prod_{i=1}^{N} (S_{i} + 1) - 1 \right]$$
(6)

In examining (5) (if the S_i are assumed to be appropriate probabilities, e.g., factors are mutually probabilistically independent or the S_i are appropriate conditional probabilities) it can be seen that this is the average of the probabilities that each factor succeeds in a particular subset of the S_i taken over all possible subsets. Thus S can be interpreted as the probability that all factors will succeed in a subset of the S_i chosen at random. This is appropriate if the factors are related in a way that requires success on each, but it is possible that not all will be relevant in a given situation. This situation represents an intermediate case between the <u>AND</u> gate ($V = \infty$) where all factors are always relevant and the average (V = 0) where exactly one factor (chosen at random) determines the outcome. We will call this case a "soft AND" gate.

E. V=-1/2. If we substitute V=-1/2 into (3) we get

N

$$S = \frac{1}{2^{N}-1} \left[\sum_{i=1}^{N} (1-(1-S_{i})) + \sum_{\substack{i=1,n \\ j>i}} (1-(1-S_{i})(1-S_{j})) + \dots + (1-\prod_{i=1}^{N} (1-S_{i})) \right]$$
(7)

$$\frac{2^{N}}{2^{N}-1} \quad (1 - \prod_{i=1}^{N} (1 - 1/2S_{i}))$$
(8)

In examining (7) it can be seen that this is the average of the probabilities that at least one factor succeeds over all possible subsets of the set $\{S_i | i = 1, \dots, N\}$. The value S can be interpreted as the probability of at least one success in a subset chosen at random. This is an appropriate rule if the factors can substitute for each other but it is possible that not all factors will be relevant in a given situation. This represents an intermediate case between the strict <u>OR</u> calculation (V = -1), where all factors are relevant and the average (V = 0), which can be interpreted as only one factor chosen at random being relevant. We will call this case a "soft OR" gate.

Thus as the constant V ranges from -1 to infinity, the scoring formula (3) covers a complete range of possible factor interactions. The most severe is when questions must all have high scores for an overall high score (V = ∞). The least severe is when a high score on any question gives a high overall score (V = -1). For -1 < V < ∞ there is a complete range of intermediate interactions including the mean when V = 0. These possible interactions have the property that if V > V', then S < S'. As the constant V increases, the score from its related formula decreases. Thus V can be interpreted as a "strength of interrelation" coefficient that ranges from complete redundancy ("parallel circuitry") V = -1 to complete interdependence ("series circuitry") at V= ∞ .

The set of algorithms using the 5 values of V just described is the basis for the aggregation scheme that is used to evaluate component questionnaires and different levels of the capability hierarchy. In principle, the parameters involved in calibrating the evaluation formula can be assessed rigorously, both by multiattribute utility theory methods and/or probability modeling. In practice, less involved calibration methods can be employed as are discussed in Section 3.

In summary, the important features to note about the component evaluation algorithm are:

- The formula can be derived axiomatically from a set of clearly stated assumptions and is defensible in being theoretically sound from a preference function viewpoint.
- 2) The formula reflects the key concepts of weighting questions based on their ranges as to importance and of allowing for a variety of interaction between factors. Although simplified

for practical implementation, the formula still provides much flexibility with respect to modeling safeguard features.

3) The formula can yield the correct results when exact probabilities are known and substituted into the computation formula, given that the fault-tree like gates are assumed appropriate from a probabilistic viewpoint.

Table 2-1 summarizes the algorithm for evaluating components.

Table 2-1. COMPONENT SCORING FORMULAS

 $S_{i} = 1 - W_{i}(1 - X_{i})$ N = number of questions V Formula Interpretation N -1 S = 1 - \prod (1 - S_i) OR Gate i=1 N -1/2 S = ($\prod (1 - 1/2 S_i) - 1$)/(2^N/(2^N - 1)) Soft <u>OR</u> Gate i=1 $s = \sum_{i=1}^{N} (s_i)/N$ 0 Average 1=1 N $s = (\prod (s_1 + 1) - 1)/(2^N - 1)$ Soft AND Gate 1 1=1 N $s = \prod (s_i)$ AND Gate 1=1

2.3 LOW LEVEL SYSTEM TASK (PERFORMANCE CHARACTERISTIC) EVALUATION

Because of the flexibility of the set of aggregation rules described in Section 2.2, it is possible to use these same rules to aggregate scores all the way up to the top of the hierarchy. A performance characteristic evaluation consists of taking the scores of several components and aggregating them to get an overall score. The components themselves may interact with each other in a fashion analogous to the way factors interact as described in Section 2.2. In general, it is not possible to select an interaction rule independently of the specific components and facility features involved. It is also desirable that an evaluator or designer should not have to select an interaction coefficient but rather, as with components, provide responses to some multiple choice questions. "System effectiveness test questionnaires" are used to decide what type of inter ction is appropriate for a particular combination of components. This concept of system questionnaire will now be discussed in more detail.

Just as there are factors that affect how a component performs, so there are system factors that affect the way a combination of components performs. Questions can be developed that address each particular system factor. Since they are multiple choice, they are given weights. An interaction rule that is generic to system factors can be selected (e.g. soft <u>AND</u>) and an overall "compatibility" score can be computed for any set of components. This score can be interpreted in a consistent preference function viewpoint manner to determine an appropriate interaction rule with which to combine component effectiveness scores. Appendix Al discusses this interpretation in more detail. Thus, evaluators can use system test questionnaires to provide input to the algorithm as to which interaction rule is most appropriate for combining component effectiveness scores to produce an overall score for a low-level system task. (Section 3 will discuss the implementation of system questionnaires in more detail.) An example of several components performing a low-level system task can be multiple sensors to detect boundary penetrations. If each sensor is independent and they act as substitutes for each other (high redundancy or "parallel circuitry") a system questionnaire would indicate that their component scores should be <u>OR</u>ed together to produce an overall score for the low-level system task. (Of course, certain performance characteristics may not require system questionnaires if it appears that a fixed interaction rule can be assigned; e.g., multiple components always soft <u>OR</u> for a particular characteristic.)

The system effectiveness test questionnaire is also a possible format for factoring in supplementary information about a facility where that is necessary for evaluation. Such information can include features not addressed by any particular component questionnaire but still necessary to properly evaluate a performance characteristic. In this case, the system questionnaire is analogous to a component questionnaire where some of the questions involve using calculated scores for other components. The system questionnaire can also contain questions pertaining to delay/response type scores and multiple access points, when it is natural to do so in terms of the subject matter involved in the questionnaire. These issues are the subject of the following paragrap's.

2.4 HICHER LEVEL AGGREGATION, DELAY/RESPONSE AND MULTIPLE ACCESS POINTS

At higher levels in the hierarchy, the aggregation should proceed in a straightforward manner. Since the hierarchy is fixed in advance, the appropriate interaction for combining low level task scores into subfunction scores etc., into an overall capability score can be specified before a evaluation is made. Since these hierarchy elements are more generic in nature, the interaction rule for combining elements should not require system questionnaires.

There are two aggregation issues, however that need further discussion. These are multiple access point type concerns and delay-response type evaluations. Multiple access points refers to the fact that different parts of the facility can each be viewed individually with respect to certain elements of the generic hierarchy. These parts then need to be aggregated together. For example, in the "Prevent Unauthorized Access of Persons and Introduction of Material into the MAA" capability hierarchy, one segment refers to denying access through the remaining (non-portal) area boundary. If a building forms part of the boundary, possible access points could include windows, walls, floors, roofs, vents, etc. Each of these access points may have its own sensor system. Conceptually, the algorithm can treat each access point individually and then use an interaction rule to combine the scores of all access points. Thus, the hierarchy segment on denying access through the area boundary could multiply into denying access through walls, floors, vents, etc. In practice (see Section 3) it may be possible to avoid subdividing the boundary into many smaller units and thus avoid proliferating hierarchy elements. These same concepts just discussed can apply to multiple portals. or multiple MAAs, etc.

The second issue requiring discussion is the evaluation of delayresponse, delay-detect, delay-assess relationships of a facilty. (Hereafter, delay-response will be used as an example. The treatment is analogous for all three). For delay-response components or procedures, two types of information are provided in the component questionnaires. The first type is subjective or qualitative information that refers to evaluating the generic implementation of delay or response components. This information is somewhat component independent. For example, in implementing a barrier for delay, there are a set of questions that help evaluate how well the barrier has been installed. These questions need not refer to the material of which the barrier is made or how much of a delay the barrier provides. The second type of

information is quantitative. It consists of a mean and range of delay times depending upon the type of barrier and type of adversary tools used. Delay-response type components differ from others in this quantitative aspect. With other components and with the "implementation" portion of delay-response components, the basic evaluation concept is one of whether the component "works or doesn't work". With delay and response, however, a "good response" really depends on how much delay is available. Thus a mechanism is needed whereby delay and response times can be compared before an evaluation of a delay-response hierarchy element can be done.

The algorithm provides for such a comparison during the evaluation in the following manner. First, the "qualitative" portion of each delay-response component is scored in the usual manner. Then, at the part of the hierarchy requiring an evaluation of the delay and response, the evaluator is asked to compare the quantitative delay times and response times provided by the system. In theory, this comparison can be done in a variety of ways ranging from using detailed probability models to a subjective preference judgement (Section 3 discusses one way of implementing this comparison using a multiple choice question). The result of the comparison is a score reflecting how well the basic design of the delay-response system works. If actual probabilities could be obtained, this score could be the probability that the delay time is greater than or equal to the response time. From a preference viewpoint, a score between 0 and 1 would be an indication of the utility of the delay-response system relative to some best and worst systems.

The design score or rating described in the last paragraph is then combined with the implementation scores of the delay-response components using one of the possible interaction rules. For example, if the "Lard <u>AND</u>" rule were selected, one interpretation could be that if either the implementation of delay or response fails, (e.g. faulty implementation cruses the barrier to be essentially ineffective), or the basic design fails because the response forces would not arrive in time, the entire delay-response element fails. A "soft AND" rule could be interpreted as providing for such possibilities as an adversary not realizing that a barrier could be negated by a particular tool or the deterrence value of a response force that may cause an adversary to flee rather than attempt access even though the access might be successful.

For the cases of multiple barriers or response forces, a preference interpretation would allow the combination of implementation scores for the like multiple components via a system questionnaire that would specify an aggregation rule depending upon how well the multiple components coordinated with each other. Sums of delay times could then be compared to response times to arrive at a system design score and the computation would proceed as above. A detailed probability interpretation or computation for multiple barriers becomes fairly complex This issue is discussed further in Section 3 of this report.

In summary, the algorithm provides mechanisms for dealing with the delay-response evaluation in a manner that allows for explicit comparison of delay times with response times.

2.5 CAPABILITIES EVALUATION

The previous discussion described the methodology for developing a score for a single capability hierarchy. The disaggregation structure with its five capabilities or objectives directly corresponds to the NRC's Physical Protection Upgrade Rule. As the hierarchies stand, each capability is evaluated independently of the others. Several issues that pertain to evaluating overall safeguards capabilities with respect to this disagregation structure are discussed in the following paragraphs.

Multiple Objectives

In one sense, the structure implicity overdesigns for physical security by evaluating each capability independently of the others. For example, in preventing theft, the only capability that must be arhieved in an absolute sense is, "Permit removal of only authorized and confirmed forms and amounts of SSNM from MAAs." Thus, theoretically it might not matter if the facility fails to "prevent unauthorized access of persons and materials into MAAs and VAs" as long as no unauthorized SSNM is permitted to leave the MAA. However, separate evaluation of capabilities implies that the ability to prevent unauthorized access is important independent of the ability to prevent theft. Thus, the five capabilities represent multiple objectives for a safeguards system rather than a single prevent theft (or sabotage) objective. Multiple objectives can address important policy issues. For instance, in order to decrease the amount of risk as perceived by the public, a facility can be evaluated on how well it prevents unauthorized access to the PA and MAA--not only on how well it ultimately prevents theft and sabotage.

The algorithm that has been developed currently provides separate scores only for each of the five capabilities. The particular disaggregation structure was provided as a "given" with the requirement for independent evaluation. Section 5 of this report discusses recommendations for combining capabilities.

Types of Adversary

An adversary may be an outsider or an insider with respect to certain areas of the facility, where "outsider" implies that either stealth or force would have to be used in order to gain access to that area. For example, a guard at the PA boundary would be an insider with respect to the PA, but an outsider with respect to the MAA (i.e., access not permitted). The nature of the adversary will determine which saf cards can first be expected to be needed. For example, on insider wit' respect to the PA (but not the MAA) whose purpose is theft will have access to the entire protected area. Thus procedures an equipment associated with entry through the PA boundary will be ineffective in stopping an insider with respect to the PA from gaining access to the PA. However, boundary controls at the MAA should be effective, as should procedures and equipment for preventing removal of SSNM from the PA boundary. The importance of the type of adversary and its effect on detection and the timeliness of a response is shown in the adversary path diagram of Figure 2-2.

An aggregation scheme for evaluating overall safeguards can be complicated by consideration of different types of adversaries. For example, to evaluate how well a facility prevents theft by an outsider would seem to require knowing how well it prevented unauthorized access to the PA and MAA. The current disaggregation structure does not make an explicit formal connection between capabilities. Rather, the implication is that of an adversary attempt originating (for all practical purposes) in the particular area addressed by the particular hierarchy with the requirement of preventing the attempt before it succeeds in involving another capability. This condition appears to be reasonable for, say, an MAA insider with respect to moving SSNM. But it becomes awkward to consider the "permit only authorized placement and movement of SSNM within MAAs" hierarchy for the case of an outsider who has arrived there by force. Since the hierarchy treats the attempt as beginning inside the MAA, it is much more natural to view the attempt as coming from an apparent insider and thus have questionnaires tailored to that situation.

The previous discussion serves to describe some of the complexities of safeguards evaluation. It may be desirable to decompose the problem by evaluating separate independent capabilities. This separation may have connotations concerning "overdesigning security"

Figure 2-2. ADVERSARY PATH

Type of T	hreat: Theft
Physical Path of Adversary	Possible Steps Performed by Adversary
PA boundary	A. Unauthorized access to PA
PA	B. Unauthorized activities in PA
MAA boundary	C. Unauthorized access to MAA
МАА	D. Unauthorized activities (non-SSNM) in MAA
SSNM storage (e.g. vault or vorking area)	E. Unauthorized placement and move- ment of SSNM in MAA
	F. Unauthorized removal of SSNM from MAA and facility
Type of Adversary	Points of Detection
l. Outsider	A - F
 Insider with respect to PA only 	B - F
 Insider with respect to PA and MAA 	D - F
 Insider with respect to PA, MAA, and SSNM 	E - F

and "typical threats". While overall safeguards evaluation seems to require combining capabilities, the possible consideration of multiple adversary "modes of attack" (stealth, force, deceit and combinations there of), insider-outsider combinations, adversary tools and pathways can make this a difficult task. Section 5 discusses what types of information can be reasonably expected from a "disaggregation structure-questionnaire" evaluation method in view of the practical constraints on the method's implementation and the complexity of the evaluation.

3.0 IMPLEMENTATION GUIDELINES

3.1 INTRODUCTION

The previous section described the methodology in terms of its capabilities for aggregating safeguard system elements. The algorithms that were developed are flexible enough in terms of the parameters provided to reflect a variety of element interrelationships. This section describes guidelines for the implementation of the methodology in terms of techniques for setting the required parameters. The discussion does not focus on what specific material should be contained in component questionnaires or capability hierarchies. Rather, guidelines are given concerning the general nature of questionnaires and hierarchies and how information in a particular format can be used in a practical manner to assess algorithm parameters. The previous section described the algorithm in a somewhat bottom to top order reflecting the actual computation that would be done. In "starting from scratch," a capabilities hierarchy would be formulated first and probably revised after the lower levels had been defined in a firstcut manner.

3.2 DEVELOPMENT OF HIERARCHIES

The objectives of a safeguard system are defined and the objectives or capability hierarchies developed to indicate how well any particular system achieves those objectives. This kind of evaluation structure is typical of multiobjective evaluation problems. There are some general guidelines for developing these hierarchy structures and these are now discussed.

Typically, hierarchy development proceeds by subdividing a major objective into subobjectives and continuing this process until a fine enough subdivision has occurred so one can evaluate in a specific fashion the lowest level subobjective. The safeguards hierarchies exhibit a natural subdivision of objectives corresponding to different portions of a nuclear facility that continues down until the component level is reached.

Some desirable properties for such hierarchies are the following:

<u>Completeness</u>: All of the essential features of the system are addressed.

- Non-Redundancy: The hierarchy should not double-count by aggregating more than once how a system achieves the same subobjective.
- Reasonable Size: The hierarchy should not proliferate both vertically and laterally to where features are being examined that could more usefully be lumped together or ignored.
- Operational: The structure should have a logical flow from bottom to top so that knowing the bottom levels enables one to proceed easily to evaluate higher levels.

There is not necessarily any unique way of developing a structure for a particular problem. The structure to be used can depend on the evaluation orientation that is desired. In the safeguards evaluation problem, there are at least two orientations that seem useful. One is a fault-tree like orientation from either a facility or an adversary viewpoint. This orientation focuses on sequences that must take place for the safeguard system to succeed or fail in achieving a particular objective. The strength of this approach is in the strong direction it provides in terms of how elements should be aggregated. However, it must be recognized that the nature of the evaluation information may not enable a fault-tree type analysis to be conducted. Analogs to probabilities and conditional probabilities may not be readily obtainable from very qualitative data or it may be too difficult to begin modeling a complex system using probability related computations. A second orientation is one of a checklist of all the features recognized as important to a safeguard system. The strength of this approach is its focus on completeness. Aggregations, however, may need to be done on a more subjective basis since the grouping of elements may not be done with a strong aggregation emphasis in mind. In practice, a hierarchy may contain a blend of both orientations.

3.3 DEVELOPMENT OF QUESTIONNAIRES

A questionnaire can be viewed as a mini-hierarchy that relates fairly specific features that can be assessed by an evaluator to an overall objective of having a component or procedure work as well as possible. As a hierarchy, the same guidelines that were discussed in Section 3.2 apply to questionnaires as well. However, because questionnaires are at the most specific level of the hierarchy, more specific guidelines can be discussed.

Each question on a questionnaire addresses a specific factor related to how well a component works. To make evaluations practical and consistent, each question has a response scale consisting of a set of multiple choices. The questions should be <u>complete</u> in addressing all important factors, <u>non-redundant</u> in not double-counting the same factor effect on the component and <u>minimal in number</u> so that very minor or irrelevant questions are weeded out.

Each specific question response scale should have the following desirable properties:

<u>Comprehensiveness</u>: The score on the scale should adequately reflect the component performance relative to the factor in question. The scale should be applicable in most situations and for most adversary actions.

<u>Operational</u>: The scales should minimize ambiguity by providing a) a sufficient number of possible responses to discriminate between most situations, b) meaningful scale point definitions that include examples for each point on the scale and use specific quantitative units where possible.

The scales may be objective such as "inches of clearance" or subjective such as a series of examples of different features that may be absent or present. Proxy scales may also be used such as "number of drills held per year" as a proxy for response force training.

For algorithm purposes, it is also desirable that response scales be defined so that the following are reasonable approximations:

Linearity of preferences over the scale responses. If two responses are almost equally desirable they should be put as alternatives for the same scale point. Extreme responses that connote an unacceptable facility should not be on a scale but rather should be noted separately for mandatory remedial action.

Utility and Preferential Independence Assumptions Hold (see Appendix Al). These are assumptions underlying the algorithm computations. In general, if the response scales do not include extreme points, these assumptions are more reasonable. The previous discussion on hierarchies and questionnaires provide suggestions for developing structures that are amenable to formal evaluation by the methodology. The following paragraphs discuss techniques for calibrating the parameters of the aggregation algorithms.

3.4 ASSESSING WEIGHTS AND AGGREGATION RULES: INTRODUCTION

The aggregation of any group of "elements", in general, requires two types of parameters to be set. For each element a weight (W_i) can be assigned to indicate its relative importance. Then an "interaction coefficient" (V) is assigned to the group as a whole. Ideally, these parameters would be assessed using formal techniques of utility theor; or subjective probability. However, due to the large number of assessments to be made, some simplified procedures are described both in the remainder of this section and in Appendix A2.

The general procedure for implementing the algorithm is as follows:

- Divide elements into groups such that the elements in any particular group can be aggregated using one rule.
- Assign a weight (this can be a relative weight) to each element.
- 3. Assign aggregation rules for each group of elements and calibrate each group's evaluation function.

Note that in general, the term "element" above could range from being a question on a questionnaire to a higher level hierarchy box. Techniques for assessing weights and aggregation rules for different elements of the hierarchy will now be discussed.

3.5 ASSESSING WEIGHTS AND AGGREGATION RULES FOR QUESTIONS ON COMPONENT QUESTIONNAIRES

Divide the Questions into Groups.

Sections 3.2 and 3.3 have already discussed how a questionnaire can be viewed as a mini-hierarchy. Conceptually, an objectives structure can be developed using the notions of "checklists" and/or "fault-trees." The groupings of questions for assigning an interaction rule need not exactly correspond to the groupings of questions for checklists, although maintaining two sets of mini-hierarchies can be confusing. The Phase I report [1] contains an example of a fault-tree for a hypothetical component questionnaire. In practice, due to the large number of questionnaires and the prospect of their revision after some trial implementations, all questions can be grouped together in one single group for assigning an interaction coefficient. If the ranges on response scales are not too extreme, this approximation can be a reasonable one.

Assign Weights and an Aggregation Rule.

Conceptually, from the multiattribute utility point of view, all weights and the interaction rule are assessed relative to the best and worst levels of each of the questions on a questionnaire. There are techniques for assessing relative weights for different questions as well as an overall "interaction" constant. [2] The use of typical techniques, however, becomes difficult because of the size of the safeguards assessment problem. There are, on the average, about fifteen questions per questionnaire and on the order of one hundred questionnaires. The questionnaires can conceivably be modified further as the algorithm is tested (see Section 5). Also, because component questionnaires are used as input to higher levels of the hierarchy, keeping in mind the

ranges of a myriad of questions when assessing upper level parameters becomes very complex.

To assess parameters in a practical yet systematic manner, techniques to be described shortly have been formulated. They allow relative weights to be assigned quickly and require, one assessment question per questionnaire to consistently link the interaction coefficient with the determination of absolute weights. (When all questions on such questionnaires are assumed to have the same relative weight, this is the only question that needs to be asked to calibrate the entire questionnaire evaluation function.)

The assessment techniques recognize that there are two natural performance level "ranges" to be considered when assessing parameters. The first is the range between the highest and lowest possible responses to a question. The weight W_i must reflect this range for a meaningful parameter calibration. The second is the implicit range that is natural when considering assigning an interaction rule. In this case, one cannot easily keep in mind all the varying question ranges. However, the endpcints of "component is not compromised" (best point) and "component is ineffective" (worst point) provide a somewhat "standardized" range that can be considered in assigning an interaction rule. The assessment techniques to follow connect these two ranges in a logical manner. Question weights are assigned to reflect the relative importance of questions based on their response scale ranges. An interaction rule is assigned based on the best-worst endpoints and is tied into determining the absolute weights for each question.

Assign Relative Weights.

As was discussed earlier, rigorous utility theory techniques for assessing relative weights are not practical for the safeguards problem. A practical procedure for assigning relative weights to individual questions emphasizes distinguishing between relatively important and unimportant questions. The procedure is as follows:

For each question the following heuristic is posed: What is the maximum possible degradation of component quality that can occur as a result of changing from a maximum to a minimum response to the question?

- A severe degradation in quality could occur, rendering the component ineffective in performing its function.
- A moderate degradation in quality could occur, resulting in a likelihood that the component would be ineffective.
- Only a minor degradation in quality could occur, with the component still likely to function properly.
- A very minor degradation in quality could occur, with only a minimal effect on component quality.

Note that the weight is assigned on the basis of the range between the highest and lowest possible responses to the question. For example if a question has five possible responses (a-e) the question should b: assigned a weight on the basis of the relative desirability of a component with a response of "a" versus one with a response of "e."

The content of this question should be such that influence from other questions or components is ignored. For example, if answers to other questions can aggravate an effect, they should be thought of as being at their best levels. If answers to other questions can mitigate an effect, they should be thought of as being at their worst values. It remains to assign the relative W_i to be associated with the response to the above heuristic. Reasonable values might be (1) $W_i = 1$, (2) $W_i = .5$, (3) $W_i = .25$, and (4) $W_i = .1$. Note that technically, at this stage, these W_i are relative and could all get multiplied by a constant depending on the interaction rule assessment.

In practice, one might choose to treat all the questions as being of roughly equal importance. This is more reasonable after relatively minor questions have been deleted. In this case, one can proceed directly to the next step, which is assessing the interaction rule and absolute weight.

Assign Interaction Coefficient and Absolute Weights.

A question for determining the interaction coefficient for a group of questions is as follows:

In general, how do the factors interact with one another?

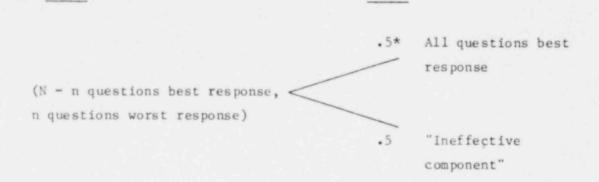
- They interact in a "strongly interdependent" manner, with a weakness on any one factor negating the strength of the other factors. Or, factors interact destructively, tending to degrade each other's performance.
- They interact in an "interdependent manner," with weaknesses accumulating to degrade the overall effectiveness of the other factors.
- They interact in a neutral manner, with the contribution of each individual factor being unaffected by the contributions of others.

- They interact in a redundant manner, with factors acting as layers of depth of defense, or making up each other's deficiencies.
- They interact in a strongly redundant manner such that if any one question gets a high score, the group score should also be high.

These five levels have natural interpretations in terms of the constant V. Specifically, we can assign (5) V = -1 (<u>OR</u> gate), (4) V = -1/2(soft <u>OR</u> gate), (3) V = 0 (average), (2) V = 1 (soft <u>AND</u> gate), and (1) V = + B (AND gate).

Note that in answering this question, there can be a tendency to have a concept of "weakness" in a factor that does not necessarily correspond to the range between best and worst on some of the questions. That is to say, there is an implicit worst point for each factor that makes it intuitively easier to assign an interaction coefficient than to explicitly consider all the best-worst points of each question. To logically connect the explicit and implicit ranges, the following question can be asked:

Given a set of N questions, with n of them having equal W_j , receiving their worst scores, and the other N - n their best, what must n be for the following situations to be about equally preferred:



В

[*This probability can be changed and in general can be set to P where $0 \le P \le 1$.]

Here choice B represents a hypothetical 50-50 gamble between getting a component that scores the best possible on all questions and one which is "ineffective." An "ineffective" component is one which receives a score of 0 using the scoring formula, and the best possible component receives a score of 1.

Appendix A2 shows that the absolute weight W (corresponding to W_j) is computed as follows:

$$W = ((1 + V)/V)[1 - (P + (1 - P)(1 + V)^{-N})^{1/n}]$$
(9)

 $-1 < V < \infty, V \neq 0$

A

For V = 0, W = (1 - P)N/n; for $V \to \infty$, $W \to >1 - P^{1/n}$; for V = -1,

n = N and $W = (1 - P)^{1/N}$

In essence, the calibration above measures how a component is perceived relative to an "ineffective" component given that it scores the worst responses on a certain number of questions. This computation links together the concept of factors making a component ineffective which is used to assign an interaction rule, and the weights W_i that indicate what range of factor "compromise" the questions actually span.

The calibration procedure for W provides a consistency adjustment that connects the original assignment of weights W_i , to the overall interaction coefficient V, so that they are assessed consistently. As an illustration of a typical case of V = I (soft AND gate), several absolute W's are shown as a function of n. Note that for N > 5, the formula for W is well approximated by the following:

 $W = 2[1 - .5^{1/n}]$ when P = .5

n	-	1	W		1	n	=	5	W	=	.26	
n	-	2	W	-	.59	n	-	7	W	=	.19	
n	=	3	W	-	.41	n	=	9	W	=	.15	
n	=	4	W	=	.32	n	=	13	W	=	.10	

3.6 INTERPRETATION OF COMPONENT SCORES AND AGG? EGATION PARAMETERS

Interpretations for the interaction coefficient have already been presented in Section 2. The interpretations concerning the weights invo've interpretations of what is meant by a factor "failing." Having a factor "fail" would ideally be defined by a specific scale point. However, for many factors, such a point is awkward to express in terms of what a facility may be expected to provide. Technically, a W_i represents the probability for which one is indifferent between the following: question has worst response

B

1

1-P factor fails

Because factor "fails" is hard to explicitly define (though easy to work with implicitly when assigning an interaction rule) and because assessing this lottery is difficult enough without doing it fifteen times per questionnaire, the heuristic mentioned earlier referring to overall component quality is used to assess relative W_i 's and an additional assessment is made to assess the absolute W_i 's.

The component score that is finally computed is the "indifference" probability that the component is "not compromised" in its performance. The terms in quotations refer to the fact that the evaluation is considered as a judgemental subjective preference assessment.

A perceived "risk" need not be the same as a calculated one, and yet still have validity in terms of its impact on decisions. Similarly, an "ineffective" component is a judgmental term that need not imply the component fails with probability equal to 1, or that the component is effectively non-existent. It simply connotes a perception of ineffectiveness. If one does equate ineffectiveness with worthlessness and indifference probabilities with actual probabilities, then the component score could represent the conditional probability that the component is effective, given the responses to the questions on the questionnaire.

3.7 ASSESSING AGGREGATION RULES FOR HIGHER LEVEL ELEMENTS OF THE HIERARCHY

Weighting Elements

For higher level elements of the hierarchy, the same basic concepts apply as they did for the component questionnaires. For these higher level elements, however, there is currently no provision for assigning an absolute weight other than 1. One reason for assuming a weight equal to 1 is the fact that an implicit normalization has already taken place in deriving the score for upper level boxes. That is, a score of 0 already connotes an ineffective box. To explicitly weight the box again would be difficult because no easy description could be given as to what the new zero point might mean. Implicit weighting actually occurs when boxes are grouped. As an analogy, consider the case where one component (say tamper protection) feeds into another component (say a sensor). All questions have weights equal to 1 and we soft AND to compute the scores for both components. If the sensor component has one response at its worst level and all others at their best. it would receive a score of about .5 (when N>5). However, if one tamper protection response was at its worst level, the tamper protection element would receive a score of .5, but the sensor would receive a score of .75 if all other responses besides tamper protection were at their best. Thus a tamper protection question implicitly has less effect on the sensor score than a sensor question: This illustrates how the grouping of elements can implicitly give less weight to those factors that affect only a relatively narrow segment of the overall hierarchy. In essence, relative weighting is automatically taking place via grouping of the elements.

Assigning Interaction Rules - System Questionnaires

To assign the interaction coefficient for a group, the same heuristic discussed in section 3.5 can be used. Another approach to assigning an interaction coefficient is that of the system questionnaire discussed in section 2.3. An example of this concept is shown in Figure 3-1. Instead of using a very general heuristic such as that in section 3.5, an evaluator answer: more specific questions such as those shown in Figure 3-1. The system questionnaire in Figure 3-1 is treated just like a component questionnaire. Weights can be assigned and a rule specified for combining question scores. As section 2.3 and Appendix Al explain, the overall score is used to specify the <u>rule</u> for combining multiple sensors. A high score means that the multiple sensors are redundant and provide defense in depth. A low score means that one poor sensor can negate the strengths of the others.

Multiple Access Points and Delay-Response

Section 2.4 disc seed the issue of multiple access points. For implementation purposes, it is simplest to aggregate such points at the lowest possible level of the hierarchy.* For example, if there are several possible entry points, each presenting a tarrier to adversary penetration, it is simplest to aggregate all the barriers using some rule rather than to analyze each barrier-access point individually up to the highest level of the hierarchy and then combine access points at the highest level. Aside from proliferating hierarchy elements, this latter approach tries to define the entire pathway that an adversary might take in trying to gain access to a facility. The algorithm and question aires were not intended for such detailed system modeling.

*NOTE: It is the Sandia authors' view that aggregating each low-level task over all access points and subsequently combining the resultant measures at each level of the hierarchy fails to reflect the essential sequence of events for detection and response functions and fails to identify the locations where these functions are of concern. A test program should help '-solve the question of correctness vs. practicality.

Figure 3-1 System Questionnaire Example: Multiple Sensors

- Will each sensor type be selected to minimize the susceptibility of any two or more sensor types to the same local environmental (natural or man-made) source of nuisance alarms?
- 2. Will each sensor type be selected to minimize the likel hood that two or more sensor types will be affected by the simultaneous occurrence of environmental (natural or man-made) sources of nuisance alarms, e.g., wind and rain?
- 3. What provisions will be made to minimize the likelihood of false or nuisance alarms?
- 4. Will collocated sensors be installed to provide mutual tamper protection for the sensors and processors?
- 5. Will collocated sensors be selected to provide coverage over a wide range of intrusion methods, (e.g., microwave to sense surface intrusion and buried cable to sense tunneling or crawling under the microwave beam or balanced magnetic switch to sense door opening and breakwire system to sense cutting through the door)?
- 6. Will collocated sensors be selected to minimize operational performance incompatabilities?

Figure 3-2 Example Question to Aid in Assessing a Delay-Response Type Score

Direct or Indirect Monitoring

Using data from the questionnaires pertaining to the barrier(s) and the type of monitoring that will be used, how will the adversary boundary penetration and/or introduction of materials time compare with time between monitoring observations?

- Adversary penetration and/or introduction of material
 time will exceed twice the time between observations
- b. Adversary penetration and/or introduction of material time will be less than twice but greater than the time between observations
- c. Adversary penetration and/or introduction of material time will be equal to the time between observations
- d. Adversary penetration and/or introduction of material time will be less than the time between observations

Similarly, it is complex to do scparate delay-response type analyses of every pathway an adversary might pursue. A more subjective evaluation of delay-response is possible as described in section 2.4. For implementation purposes, a multiple-choice question such as the one shown in Figure 3-2 can aid in assessing a delay-response score that considers multiple barriers/access points. (This question can have a weight.) Alternatively, the delay-response score can be directly assigned using whatever means is appropriate. (See those suggested in section 2.3.)

<u>Cuestionnaires</u> Input to Other Questionnaires

The issue of questionnaires being used as input to other questionnaires has already been discussed in terms of component guestionnaires (e.g. tamper protection input to sensors) and system questionnaires (components combined with other information, such as described in section 2.3). Two effects occur when this is done. First, as was described in the beginning of this section, the questions on questionnaires feeding into others generally have implicitly much less weight, Second, a guestionnaire is always the lowest level the hierarchy algorithm can consider. A systems questionnaire must be completed manually with other component scores being input manually (after a first pass computation). If sensitivity analysis is desired or input-type questionnaire responses, a manual update must be done before the algorithm can be run. Furthermore, in order to input one guestionnaire into another, the score of 'he input questionnaire must be discretized to be in the proper mulciple choice format. Because of these effects, we recommend the following:

> Eliminate as much as possible all input-type questionnaire situations. For example, tamper protection has a small influence on the scores of components to which it is input. One question with many multiple choice responses should be used to replace the entire tamper protection guestionnaire, or

else it should be treated as a component that does not feed into another questionnaire.*

• Compose system questionnaires so that they <u>only determine</u> the way in which components combine to perform a function. When other information is mixed in on a system questionnaire, it makes the analysis complex and also "cuts off" the algorithm from tracing results down to components below the system questionnaire level.

Sections 3.6 and 3.7 have described techniques for calibrating the parameters of the algorithm. The following subsections describe the basic steps for exercising the algorithm, interpreting the evaluation results and performing sensitivity analysis.

3.8 INSPECTION (DESIGN) STEPS - DATA REQUIREMENTS

The steps performed in exercising the algorithm are now summarized. The assumptions are that a capability hierarchy is defined and all component and system questionnaires required for the analysis have been developed. In addition, all the fixed interaction coefficients for hierarchy boxes and questionnaires and the weights for questions are assumed to be assigned.

Step 1. Identify all the components to be used for accomplishing lowlevel system tasks (performance characteristics) using the comrenert selection matrices.

*NOTE: The opposing view taken by the Sandia authors is that to evalluate such components as tamper protection, emergency power supplies, etc. independently and then combine results with those from components such as sensors, CCTV, etc. effectively places equal importance on tamper protection as on the combined effect of all the other sensor performance factors.

- Step 2. Answer all of the guestionnaires for the components above. (Questionnaires that feed into others need to be scored first before this can be done. See Step 5.)
- Step 3. Using the component selection matrices, identify those components that are used to perform the same low level system task. Answer any system guestionnaire required to indicate how well these components coordinate with each other.
- Step 4. Compose computer input. (See Section 4 for details.) Select questionnaire input for those components involved and compose File 1. Organize questionnaire responses in File 2. Indicate which questionnaires feed into which hierarchy elements in File 3.

Step 5. Use the algorithm to score all guestionnaires.

Step 6. Assign scores to delay-response hierarchy boxes if they do not have questionnaires determining their scores. (Note whether the same barriers are being used both to help detect and/or assess as well as delay. If so, the delay-response score should take into account the fact that the barrier is doing "double or triple" duty.)

Step 7. Score capability hierarchy.

Step 8. Evaluate results and perform sensitivity analysis.

A capability is scored for a single set of evaluator responses. (Multiple evaluators are not currently handled by the algorithm in terms of such possibilities as averaging evaluator scores or consistency checking them.) The following paragraphs discuss Step 8 in more detail.

3.9 EVALUATION OF RESULTS - SENSITIVITY ANALYSIS

Once a capability is evaluated, the results can be interpreted in a comparative sense. As an illustration of this, suppose the algorithm is used to evaluate systems A and B and system A scores higher than B. The implication is that system A is preferred to system B. If system A is considered a barely acceptable facility, then system B might be considered unacceptable. Whatever score system A received is then considered a threshold score.

Another interpretation can be given in terms of an "ideal" system versus an "ineffective" system. Theoretically, a facility receiving a score of .5 on a capability is considered equally preferred to a 50-50 gamble between an ideal system getting perfect responses on all questionnaires and an "ineffective system" receiving an overall score of 0. (A system receiving the worst response to all questionnaires will probably have a score very close to 0). A score below .5 indicates that one is willing to take such a gamble. This does not indicate much faith in the facility to perform the capability. On the other hand, a score of above .5 indicates that at least one would rather stay with such a facility than take an even chance at "perfecting it" versus losing its effectiveness. (This same interpretation can be applied to individual components and lower level boxes.)

The computer program described in section 4 provides the routines for tracing the capability score computation down through lower level boxes all the way to the component level and to questionnaire responses. It provides displays that help to pinpoint particularly low scores, or situations where many elements are "ANDing" together to produce a lower score than may be desired. The designer or inspector can then see whether improving a component, adding a component or improving the way components coordinate can help improve the facility score.

Sensitivity Analysis

The computer program enables the user to change the following parameters to see how the results are affected:

- guestionnaire weights and responses
- all aggregation rules
- scores for any box or guestionnaire

With these features, a user can examine what "improvements" will cause one facility to be at least as preferred as another. Changing weights and aggregation rules also allows the evaluation to span a range of relatively more conservative assumptions (e.g. larger weights and more AND operators) to relatively less conservative assumptions (e.g. smaller weights and fewer AND operators). In this way, the algorithm is a useful tool that can help analyze what set of assumptions cause a facility to be evaluated as relatively acceptable or not.

In summary, section 3 has discussed how the algorithm can be implemented and used. Section 4 describes the computer program that facilitates this implementation. (Hand calculations and programmable calculators can be used to compute scores for individual elements since the scoring algorithms are straightforward. But the computer program greatly facilitates the handling of hierarchies and large numbers of questionnaires). An example presented in section 4.3 illustrates the algorithm implementation. SAFEGUARDS EVALUATION COMPUTER PROGRAM

4.0

4.1 INTRODUCTION

To implement the algorithm described in previous sections, an evalution computer program has been developed. This program is designed to automate the scoring of evaluation questionnaires and hierarchy elements and to provide maximum flexibility to the user for sensitivity analysis and other changes.

The program uses two basic types of input. The first type provides the structure of the questionnaires and hierarchies, including the number of questions (or inputs to a hierarchy element) weights and the scoring rules to be used. This data is independent of any particular evaluation and can be developed and stored on the computer before an evaluation is done. The second type of input is the responses to the questionnaires. These are the answers to the questionnaires filled out by an evaluator.

To compute the score for a hierarchy, the program first looks at questionnaires. The questionnaire structure (number of questions, weights, lowest values for each question, etc.) is read from one disk file and responses are read off another. The routine then automatically computes and saves the questionnaire score. After the questionnaires have been scored the program can be switched into hierarchy mode. To score a hierarchy element (box) its name is entered. If the scores for all the boxes leading into the box have been computed it scores the box using

the appropriate rule. If not, the program attempts to score lower level boxes, gradually working down in the hierarchy until a box whose score can be computed is found. The program then works back up the hierarchy until the original box's score can be computed. Low level boxes (with component questionnaires) are scored in the same way except that the program assumes that all questionnaires have been scored.

The rest of this chapter describes the structure and operation of the evaluation computer program in more detail. Section 4.2 describes the use of the program and the format of the associated input files. Section 4.3 provides an example of the program's use. Listings of the program are presented in Appendix A3.

4.2 USE OF THE EVALUATION PROGRAM

This section describes how the evaluation computer program can be used to evaluate questionnaires and hierarchies. First the data base is described in detail, then the operation of the program, including the various options available and the flow of the program, is described. The use of the program is demonstrated in section 4.3.

DATA BASE

The input to the program consists of four "files" (sets of data stored on cards or disk):

- 1. Questionnaire structures
- 2. Questionnaire responses
- 3. Hierarchy structure
- 4. Hierarchy initial scores

The content and format of these files is described in more detail below. The program is written in FORTRAN and FORTRAN formats are listed where appropriate.

Questionnaire Structures

This file contains information on the questionnaires to be evaluated. The data for each questionnaire includes:

- Name
- Number of questions
- Number of subgroupings (if any)
- · Weight for each question
- Lowest possible response for each question
- Scoring rules for overall group and subgroups

The specific layout is as follows:

Card (record) 1: Number of questionnaires in file (112 format)

- Card 2: First card number for each questionnaire (i.e. the number of the card where the questionnaire starts). (2014 format) This card is repeated as necessary to specify the first record of all questionnaires in the file.
- Card 3: Questionnaire title. Contains the questionnaire name (maximum of four characters) the number of questions (maximum of 40) and the number of subgroups (counting the overall questionnaire as 1). The inital implementation of the algorithm will not use subgoups, but the program has the ability to process them. Format (1A4,6X,1I2,8X,1I2).

- Card 4: Worst Response. The letter corresponding to the worst response is given for each question. (The best response is always assumed to be "A".) (40(1X,1A1)).
- Card 5: Group Information. Group number. (The group number for the overall questionnaire is always 50). Additional groups are numbered 51,52...etc. The number of questions (and subgroups) to be aggregated and the rule to be used is also given. The codes for rules are HA = hard AND, SA = soft AND, AV = average, S0 = soft OR, OR = hard OR. Format (112,8X,112,8X,1A2).
- Card 6: Group Inputs. The questions (or subgroups) to be aggregated as part of the group are given in (4012) format.
- Card 7: Question Weights. contains the weight (between 0 and 1) assigned to each question. Initially the questions will be equally weighted at 0.5, but the program can accept differential weights. (8F5.3) This card is repeated until a weight is specified for each question. A convenience option allows one weight for all questions to be set by specifying 2. as the first "weight" and the assigned weight for all as the second weight.

Cards 5 and 6 are repeated for each group. Cards 3 through 7 are repeated for each questionnaire.

Questionnaire Reponses

This file contains the responses to the various questionnaires (the results of the evaluation). The format is as follows:

Card 1: The number of questionnaires evaluated. (112) format.

Card 2: The first card number for each questionnaire. (2014) format. The questionnaires must be in the same order as in the questionnaire structures file. This card is repeated as many times as necessary to identify the first record for each questionnaire.

Card 3: The name of a questionnaire. (1A4) format

Card 4: The score for each question on the questionnaire. In alphabetic format (40(1X,1A1)).

Cards 3 and 4 are repeated for each questionnaire.

Hierarchy Structures

This file contains structural data about the organization and scoring of hierarchies. The format is as follows:

Card 1: The number of complete hierarchies in the file. Format (112), maximum value = 5. (Typically, there is only one hierarchy in a file)

Card 2: The first card number for each hierarchy (514) format.

- Card 3: Box Data Card. This card includes the name of a box, the number of subelements to be aggregated and the scoring rule to be used. If the elements to be aggregated are questionnaires instead of boxes then 50 is added to the number of subelements. If a questionnaire is to be used to determine the scoring rule the questionnaire name also appears on the card. The order is: box name, number of elements, rule, questionnaire name (if any). The format is (1A6,4X,1I2,8X,1A2,8X,1A4).
- Card 4: Input Box Data Card. This card contains the name of an input subelement (box or questionnaire) in (1A6) format.

Card 4 is repeated for each input subelement. Cards 3 and 4 are repeated for each hierarchy box having sub-elements. The only restriction of the ordering of the boxes is that a box name must not appear on a number 4 card after it has appeared on a number 3 card, (i.e, go from top to bottom).

Card 5: The last card for each hierarchy is a card with the word "NOMORE" in the first six columns.

Hierarchy Initial Scores

This file contains values for any initial scores to be set for hierarchy boxes. The file is structured as follows:

Card 1: The number of hierarchies in (112) format. (Typically, this number is 1)

Card 2: The initial card for each hierarchy in (514) format.

Card 3: The names of boxes to be set followed by the initial score. If the score is set at -1, the initial score is free. (Otherwise scores must be between 0 and 1). There are no restrictions on the order of the boxes. If a box does not appear its initial score is assumed to be -1. The format is (5(1A6, 1F6.3)).

Card 3 is repeated until all set scores have been read in.

Card 4: Questionnaire scores in (10F8.5) format. Ten cards (they may all be blank) are required for each hierarchy.

> This file is used primarily by the program to store the results of an evaluation so that they need not all be computed each

time the program is run. Initially, the file can be set with the first two cards specified as above, and 18 blank lines following the first two cards.

INTERACTIVE PROGRAM OPERATION

Questionnaires and hierarchy elements are evaluated using an interactive computer program. This program uses the data files described in the previous section as input and provides the user with a wide variety of evaluation and sensitivity analysis options. The following paragraphs describe the relationship of the program elements and data files and the options available to the user.

Input/Output Considerations

The evaluation program is designed to be used interactively at a timesharing terminal. In addition, four disk storage files are needed. These files were described previously. They interface with the program as shown in Table 4-1.

Program (peration: General Features

When the evaluation program is called, it first initializes the major variables and then prompts the user with the following question:

SELECT 1-HIERARCHIES 2-QUESTIONNAIRES 3-STOP --

Typing "1" in response to this question initiates the hierarchy manipulation portion of the program. A second list of options will be printed to allow the user to control the manipulation. These options are described shortly. Similarly, if the user responds with "2", a set of options relating to questionnaires is printed. Typing "3" stops the program. If the user is familiar with the program options described

Table	4-1.	DATA	BASE	DEFINITIONS
			1000 B.L. 100	SALA THAT TOTAD

Fi	1e	Unit	Туре	Record Length (Characters)	Maximum Number of Records
1.	Questionnaire Structures	1	Random Access	80	300
2.	Questionnaire Responses	2	Random Access	80	150
3.	Hierarchy Structure	3	Random Access	80	200
4.	Hierarchy Scores	4	Random Access	80	50

below, any valid option number can be typed and the program will branch directly to that option.

Program Operation: Questionnaire Manipulation

Selecting the questionnaire option causes the following table to be printed.

SELECT ONE:

21-COMPUTE SCORES	22-PRINT SCORES
23-SET SCORES	24-REVISE WEIGHTS
25-REVISE RULES	26-REVISE RESPONSES
29-NO MORE REVI	SIONS

WHICH?

The user simply types in the number corresponding to the desired option and the computer will initiate the option and ask additional questions to enable its completion. The options are described in more detail below.

Option 21 - Compute Scores. This option computes the score for a questionnaire. When the option is selected the prompt "ENTER QUESTIONNAIRE NAME -- " is given. If the name is valid, the questionnaire's information is retrieved from the questionnaire structure and response files and the score is printed and stored. If "ALL" is typed in response to the name prompt, all the currently stored questionnaires are scored and printed. The user is then asked to select another option.

Option 22 - Print Scores. This option prints the data associated with a questionnaire. A name is entered as in Option 21 and the computer prints a table of information for the questionnaire. The information includes the scoring rule and score and a diagram of the questionnaire structure. The structure shows the subgroups (if any) used in scoring the questionnaire, the scoring rules used for the subgroups and the individual questions included in each group along with their associated raw scores, weights and adjusted scores.

Option 23 - Set Scores. This option allows the user to directly specify a score for a questionnaire. In response to a prompt the user enters a questionnaire name. The prompt "SCORE =" is printed and the user may enter any value between 0 and 1.0. This score is saved until the score is recomputed or reset.

Option 24 - Revise Weights. The option allows the user to revise the weight assigned to a given question or questions. After the questionnaire name is entered, the prompt "NUMBER OF QUESTIONS TO BE REVISED =" is given. If the weight has been assigned using the brief form, the common weight assigned to all questions must be revised. For each question to be revised, the prompts "QUESTION NUMBER =" and "WEIGHT =" allow the new weight to be assigned to the appropriate question. After this option is completed the score is recomputed and printed.

Option 25 - Revise Rules. This option allows the user to revise the scoring rule used to score a questionnaire or subgroup. After the questionnaire name is entered, the computer asks for the "group number" to be changed. Group "50" corresponds to the overall questionnaire and 51, 52, etc., correspond to the subgroups (if any). Next the revised rule is requested, using the following abbreviations:

> HA-Hard AND SA-Soft AND AV-AVERAGE SO-Soft OR OR- OR

The revised score is computed after the desired number of changes has been made.

Option 26 - Revise Responses. This option allows the user to revise the responses associated with particular questions. The procedure is similar to that for revising weights, in that the questionnaire name and number of questions to be revised initializes a loop for entering revised responses. For each question, a prompt asks for the question number and then the user is prompted "ENTER REVISED RESPONSE (A to WORST) --". WORST is the letter of the alphabet corresponding to the worst answer on the question. The user enters the letter of the alphabet corresponding to the revised response. After all desired changes have been completed the questionnaire score is recomputed and printed.

Option 29 - No More Revisions. This simply returns the program to the original hierarchy/questionnaire/stop choice. These options represent all of the interactive routines relating to questionnaires. Other changes (e.g., revisions to questionnaire structure) must be made using a text editor on the appropriate files.

Program Operation: Hierarchy Manipulation

When the hierarchy manipulation option of the program is first initiated, the computer requests "ENTER HIERARCHY NUMBER --". The user enters the number of the hierarchy to be manipulated in the current session. The computer then retrieves the data corresponding to that hierarchy from the disk files. Next the following table is printed:

SELECT ONE:

41-COMPUTE SCORES'-PRING DATA43-ASSIGN SCORES44-REVISE DELAY/RESP45-REVISE RULES46-SELECT NEW HIERARCHY47-PRINT BOX NAMES48-FILE HIERARCHY DATA49-CHANGE BOX NAMES50-PRINT HIERARCHY51-NO MORE REVISIONS

4-11

WHICH?

Typing the number corresponding to an option initiates the option. As described below, the computer asks additional questions as necessary to allow completion of the option.

Option 41 - Compute Scores. This option allows the user to compute the score for a hierarchy box. Of course, if the top box of the hierarchy is scored, the overall score will be computed. After the box name is requested and entered, the computer automatically searches as far down in the hierarchy as is necessary (up to a maximum of five levels) to identify boxes which can be scored, (i.e., boxes for which scores are available for each lower level box or questionnaire). Then the computer works back up the hierarchy, scoring higher level boxes until it is possible to compute the score for the requested box. This score is printed. (Note: The scores for all higher level boxes are reinitialized to -1 when a lower level score has been changed.)

Option 42 - Print Data. This option allows the user to obtain a simplified diagram of the hierarchy structure beneath a specified box. Up to four levels of boxes are printed. The information for each box includes the box name, its score, (-1 is shown if the score has not been computed) and scoring rule and scoring questionnaire (if any). The table is printed in outline style, with lower level boxes being indented beneath higher level boxes.

Option 43 - Assign Scores. This option allows the user to assign a score to a specified box. The computer first prompts for the box name and then requests the score, which must be between 0.0 and 1.0. The scores for all higher level boxes are reinitialized to show that a lower level score has been changed.

Option 44 - Revise Delay/Response. This option is not used at the current time.

Option 45 - Revise Scoring Rule. This option allows the user to change the scoring rule associated with a box. The computer first requests the box name and then the rule. The rule is entered using the following abbreviations.

HA-Hard AND SA-Soft AND AV-Average SO-Soft OR OR-OR Q -Scoring rule determined by questionnaire

If Q (Questionnaire Scoring) is entered, the computer will prompt for the questionnaire name.

Option 46 - Select New Hierarchy. This option reinitializes the program by allowing the user to reenter the data for the current hierarchy or another stored in Disk File 3. The only prompt is "EN'L'R HIERARCHY NUMBER".

Option 47 - Print Box Names. This option causes a list of the current box names to be printed.

Option 48 - File Hierarchy Data. This option saves all revisions and scores for the hierarchy (including questionnaire scores) during the current session on Disk File 3 and 4 respectively. The original data is overwritten. This option is done automatically at the termination of a session if Options 45 or 49 have been used.

Option 49 - Change Box Name. The computer first prompts for the original box name and then for a revised name. Names are allowed to be a maximum of 6 characters long. Option 50 - Print Hierarchy. This option is similar to option 42 except that the structure is printed in a more easy to follow graphical form. One to five hierarchy levels are printed starting with a box name entered with the computer prompt. Warning: If you are in a hurry or conserving paper it is best to use Option 42 for viewing hierarchy data.

Option 5. - No More Revisions. This option reverts the program back to the original questionnaire/hierarchy/stop choice.

4.3 F JAMPLE OF ALGORITHM AND COMPUTER PROGRAM USE

To illustrate the algorithm and computer program, hypothetical data was provided by Sandia concerning the capability of "prevent unauthorized access of persons and materials into the MAA" (see Figure 1-1(1)). A set of component questionnaires was filled out corresponding to several, but not all of the low-level system tasks. The weight on each question in every component questionnaire was set equal to .5. Each component's score was computed by using a soft AND operator on the question scores. (Some questionnaires that were input to others were scored in a "first pass".) A total of 38 questionnaires appear directly in the computer program input data which can be found in Appendix A4.

Interaction coefficients were assigned for all hierarchy boxes requiring them. All boxes without appropriate component questionnaire input were assigned arbitrary scores (e.g., a 1 in most cases).

The example is developed in terms of the commands that would be issued ' a user to the computer and the resulting output. Appendix A4 contains computer input and additional computer output for the entire capability. In this example, lowever, the focus is on that segment of the capability hierarchy concerned with "detect access/ introduction of material through remainder of area boundary." (This is the only segment of the example with reasonably complete questionnaire input data.)

4-14

Each questionnaire was given an identifying number and each hierarchy box was given a mnemonic identifier. Appendix A4 lists the component corresponding to each questionnaire number. The mnemonics for this example are shown in Figure 4-1.

In exercising the algorithm, the first command to the computer should be to read in the hierarchy structure (and any initializing information). This step is shown in Figure 4-2. The next step is to score all the questionnaire (if such scores have not been computed and stored previously). The appropriate option for this is 21. Figure 4-3 shows how all the questionnaire scores can be computed. Figure 4-4 shows an example of how a more detailed printout for the annunciator systems component questionnaire can be displayed.

The next step is to evaluate the capability hierarchy. (First, boxes requiring assigned scores should be given these scores using option 43.) The computer is capable of evaluating up to five hierarchy levels down. Therefore, lower level boxes must be evaluated first if the top box has more than five levels beneath it. In this example, the right side of Figure 1-1(1), "Deny Access" (DENACC) can be evaluated directly using option 41. One form of computer display of the hierarchy results is shown in Figure 4-5. By using a series of display commands, the user can focus on different segments of the hierarchy. For example, in Figure 4-5, the DETACC display shows a trace of the scoring down to the questionnaire level for some boxes. The second page of Figure 4-5 illustrates how one can obtain a more detailed look at lower level boxes such as INDM1.

A second way of displaying hierarchy data is shown in Figure 4-6. This type of display does not contain questionnaire scores explicitly, but does give a pictorial view of hierarchy relationships. Figure

4-15

Figure 4-1. MNEMONICS FOR ALGORITHM EXAMPLE

- DETACC Detect Access
- SENSE Sense Attempt
- REPALR Report Alarm
- ASSESS Assess Alarm
- MULTS Multiple Components for Sensing Attempts
- INDIM1 Indirect Monitoring to Sense Attempts
- TSIG Transmit Signal
- ANALRM Anunciate Alarm
- MULTA Single or Multiple Componer s for Assessing Alarms
- INDA1 Indirect Assessment of Al: s
- CASSAS Central and Secondary Alarm Stations Score
- DDS Delay/Detection Score (based on time comparisons)
- GP Guard Patrol Score
- BARR Barriers (multiple access points)
- DDA Delay/Assessment Score (based on time comparisons)
- ALAS Alarm Assessment System Questionnaire (not actually used in capability evaluation)
- PNSS Multiple Sensor Penetration Sensing System Questionnaire (not actually used in capability evaluation).

Figure 4-2. READING IN THE HIERARCHY STRUCTURE

EXECUTION:

SELECT 1- HIERARCHIES, 2- QUESTIONAIRES, 3- STOP -- 1 ENTER HIERARCHY NUMBER -- 1

SELECT ONE:

41- COMPUTE SCORES42- PRINT DATA43- ASSIGN SCORES44- REVISE DELAY/RESP45- REVISE RULES46- SELECT NEW HIERARCHY47- PRINT BOX NAMES48- FILE HIERARCHY DATA49- CHANGE BOX NAME50- PRINT HIERARCHY51- ND MORE REVISIONS

Figure 4-3. COMPUTING QUESTIONNAIRE SCORES

SELECT 41-51	2	1				
ENTER QUESTID	NAIR	E	NAME	AL	1	
QUESTIONNAIRE	4	:	THE	SCORE	=	0.755
QUESTIONNAIRE	6	:	THE	SCORE	=	0.766
QUESTIONNAIRE	10	:	THE	SCORE	=	0.670
QUESTIONNAIRE	47	:	THE	SCORE	=	0.320
QUESTIONNAIRE	57	:	THE	SCORE	=	0.579
QUESTIONNAIRE	1	:	THE	SCORE	=	1.000
QUESTIONNAIRE	2	:	THE	SCORE	=	0.917
QUESTIONNAIRE	3	:	THE	SCORE	=	0.606
QUESTIONNAIRE	11	:	THE	SCORE	=	0.820
QUESTIONNAIRE	14	:	THE	SCORE	=	1.000
QUESTIONNAIRE	16	:	THE	SCORE	=	1.000
QUESTIONNAIRE	21	:	THE	SCORE	=	0.911
QUESTIONNAIRE	22		THE	SCORE	=	0.237
QUESTIONNAIRE	25	:	THE	SCORE	=	0.562
QUESTIONNAIRE	28	:	THE	SCORE	=	0.516
QUESTIONNAIRE	32	:	THE	SCORE	=	0.750
QUESTIONNAIRE	36	:	THE	SCORE	=	0.875
QUESTIONNAIRE	33	:	THE	SCORE	=	1.000
QUESTIONNAIRE	43	:	THE	SCORE	=	1.000
QUESTIONNAIRE	51	:	THE	SCORE	=	0.766
QUESTIONNAIRE	60	:	THE	SCORE	=	0.516
QUESTIONNAIRE	63	:	THE	SCORE	=	0.337
QUESTIONNAIRE	66	:	THE	SCORE	=	1.000
QUESTIONNAIRE	68	:	THE	SCORE	=	1.000
QUESTIONNAIRE	69	:	THE	SCORE	=	0.548
QUESTIONNAIRE	74	:	THE	SCORE	=	1.000
QUESTIONNAIRE	75	:	THE	SCORE	=	1.000
QUESTIONNAIRE	83	:	THE	SCORE	=	0.746
QUESTIONNAIRE	84	:	THE	SCORE	=	0.637
QUESTIONNAIRE	87	:	THE	SCORE	=	0.733
QUESTIONNAIRE	90	:	THE	SCORE	=	0.667
QUESTIONNAIRE	95	:	THE	SCORE	=	0.337
QUESTIONNAIRE	12	:	THE	SCORE		0.338
QUESTIONNAIRE	33	:	THE	SCORE	=	1.000
QUESTIONNAIRE	ALAS		THE	SCORE	=	0.598
QUESTIONNAIRE	PNSS		THE	SCORE	=	1.000
QUESTIONNAIRE	17	:	THE	SCORE	=	1.000
QUESTIONNAIRE	18		THE	SCORE	=	1.000

Figure 4-4. DISPLAYING QUESTIONNAIRES

SELECT 21-29 -- 22 ENTER QUESTIONAIRE NAME -- 4

> QUESTIONAIRE DATA FOR QUESTIONAIRE 4 OVERALL SCORE = 0.75493 RULE : SA

BOX: 50 RULE: SA

Q=	1	RESP=	1.000	4=	0.500	5=	1.000	
Q =	2	RESP=	0.667	$ _{M} =$	0.500	5=	0.833	
Q=	3	RESP=	1.000	W=	0.500	=2	1.000	
Q=	4	RESP=	1.000	ω=	0.500	S=	1.000	
Q=	5	RESP=	1.000	W=	0.500	S=	1.000	
Q=	6	RESP=	1.000	1.1=	0.500	5=	1.000	
Q=	7	RESP=	1.000	iul =	0.500	=2	1.000	
Q=	8	RESP=	1.000	4=	0.500	S=	1.000	
Q=	9	RESP=	1.000	lal =	0.500	5=	1.000	
Q=1	0	RESP=	1.000	4=	0.500	=2	1.000	
Q=1	1	RESP=	1.000	$ _{M} =$	0.500	=2	1.000	
Q=1	2	RESP=	1.000	1.1=	0.500	S=	1.000	
Q=1	3	RESP=	0.833	₄ =	0.500	=2	0.917	
Q=1	4	RESP=	1.000	W=	0.500	=2	1.000	
Q=1	5	RESP=	0.667	id=	0.500	=2	0.833	
Q=1	6	RESP=	0.750	1,1=	0.500	=2	0.875	

WHICH? 42 ENTER BOX NAME -- DENACC HIERARCHY DATH FOR BOX DENACC BOX: DENACC RULE: SA SCORE: 0.436 Q: BDX: DETACC RULE: HA SCORE: 0.335 Q: BOX: SENSE RULE: SD SCORE: 0.702 Q: BOX: MULTS RULE: AV SCORE: 0.671 BOX: INDM1 RULE: SA SCORE: 0.575 0: 0: BOX: REPALR RULE: HA SCORE: 0.868 Q: BDX:TSIG RULE:SD SCORE: 0.940 BDX:ANALRM RULE:SD SCORE: 0.923 0: 0: BOX: ASSESS RULE: AV SCORE: 0.549 Q: BOX: MULTA RULE: AV SCORE: 0.670 0: RULE:SA SCORE: 0.640 0: BOX: INDA1 BOX: CASSAS RULE: AV SCORE: 0.338 Q: BOX: RESACC RULE: HA SCORE: 0.730 Q: BOX: COMRSP RULE: SA SCORE: 1.000 Q: BOX: BETGDS RULE: SCORE: 1.000 61 BOX: GDSSTN RULE: HA SCORE: 1.000 0: 0: SCORE: 1.000 BOX: BETSTN RULE: SCURE: 1.000 BOX: DNDFF RULE: 0: RULE:SA SCORE: 0.730 Q: BOX:RESP BDX:DELRSP RULE:SD SCDRE: 0.790 BDX:EFFRSP RULE:SA SCDRE: 0.706 0: 0: BOX: DRRSP RULE: SCORE: 1.000 0: SELECT 41-51 -- 42 ENTER BOX NAME -- DETACC HIERARCHY DATA FOR BOX DETACC BOX: DETACC RULE: HA SCORE: 0.335 Q: BOX: SENSE RULE: SD SCORE: 0.702 0: BOX: MULTS RULE: AV SCORE: 0.671 Q: QUESTIONNAIRE: 6 SCORE: 0.766 SCORE: 0.579 QUESTIONNAIRE: 57 SCORE: 0.670 QUESTIONNAIRE: 10 BOX: INDM1 RULE: SA SCORE: 0.575 Q: RULE: SCORE: 0.833 0: BOX: DDS 0: RULE: Nº SCORE: 1.000 BOX: GP RULE: SH SCORE: 0.370 0: BOX: BARR BOX: REPALR RULE: HA SCORE: 0.868 Q: BOX: TSIG RULE: SD SCORE: 0.940 0: QUESTIONNAIRE: 47 SCORE: 0.820 QUESTIONNAIRE: 18 SCORE: 1.000 BOX: ANALRM RULE: SD SCORE: 0.923 Q: SCORE: 0.755 QUESTIONNAIRE: 4 QUESTIONNAIRE: 51 SCORE: 0.766 SCORE: 1.000 QUESTIONNAIRE: 43 BOX: ASSESS RULE: AV SCORE: 0.549 Q: BOX: MULTA RULE: AV SCORE: 0.670 D: QUESTIONNAIRE: 10 SCORE: 0.670 BOX: INDA1 RULE: SA SCORE: 0.640 Q: RULE: SCORE: 1.000 Q: BOX: DDA RULE: AV SCORE: 1.000 D: BOX: GP RULE: SA SCORE: 0.370 BGK: BARR Q: BOX: CASSAS RULE: AV SCORE: 0.338 Q: QUESTIONNAIRE: 12 SCORE: 0.338

Figure 4-5. (Continued)

SELECT 41-51 -- 42 ENTER BOX NAME -- INDM1

HIERARCHY DATA FOR BOX INDM1

BOX: INDM1	RULEISA	SCORE	0.575	Q:		
BOX: D	DS RU	.E:	SCORE:	0.833	Q:	
BDX: G	P RU	LE:AV	SCORE:	1.000	Q:	
	QUESTID	NHAIRE	43	SCORE:	1.000	
BOX: B	AFR RU	LE:SA	SCORE:	0.370	Q:	
	QUESTID	NNAIRE	3	SCORE:	0.606	
	QUESTID	NNAIRE:	21	SCORE:	0.911	
	QUESTID	NNAIRE	28	SCORE:	0.516	
	QUESTID	NNAIRE	38	SCORE:	1.000	
	QUESTID	NHAIRE:	68	SCORE:	1.000	
	QUESTID	NNAIRE:	69	SCORE:	0.543	
	QUESTID	NNAIRE	90	SCORE:	0.667	
SELECT 41-						
ENTER BOX	MAME R	ESP				

HIERARCHY DATA FOR BOX RESP

BOX: RESP RULE:SA SCORE: 0.730 9: BOX: PELRSP RULE: SD SCORE: 0.790 Q: BOX: BARR RULE: SA SCORE: 0.370 0: SCORE: 0.606 SCORE: 0.911 QUESTIONNAIRE: 3 QUESTIONNAIRE: 21 SCORE: 0.516 QUESTIONNAIRE: 28 SCORE: 1.000 SCORE: 1.000 QUESTIONNAIRE: 33 QUESTIONNAIRE: 68 QUESTIONNAIRE: 69 SCORE: 0.548 SCORE: 0.667 QUESTIONNAIRE: 90 BOX: GP RULE:AV SCORE: 1.000 Q: QUESTIONNAIRE: 4 SCORE: 1.000 BOX: EFFRSP RULE: SA SCORE: 0.706 Q: BOX: DHSITE RULE: SA SCORE: 0.559 Q: BDX:REODFF RULE:HA SCORE: 0.338 BDX:CONADV RULE:SA SCORE: 1.000 BDX:DFFSIT RULE:SA SCORE: 1.000 Q: 0: Q: BOX:RSPREQ RULE:HA SCORE: 1.000 BOX:ENGADV RULE:SA SCORE: 1.000 0: 0: BOX: DRRSP RULE: SCORE: 1.000 Q: SELECT 41-51 -- 42 ENTER BOX NAME -- EFFRSP

HIERARCHY DATA FOR BOX EFFRSP

BOX: EFFRSP RULE: SA SCORE: 0.706 Q: BOX: DNSITE RULE: SA SCORE: 0.559 Q: BOX: REQOFF RULE: HA SCORE: 0.333 0: QUESTIONNAIRE: 12 SCORE: 0.338 SCORE: 1.000 QUESTIONNAIRE: 16 BOX: CONADY RULE: SA SCORE: 1.000 Q: QUESTIONNAIRE: 16 SCORE: 1.000 QUESTIONNAIRE: 43 SCORE: 1.000 BOX: OFFSIT RULE: SA SCORE: 1.000 Q: BOX: RSPREQ RULE: HA SCORE: 1.000 Q: QUESTIONNAIRE: 16 SCORE: 1.000 BOX: ENGADY RULE: SA SCORE: 1.000 Q: QUESTIONNAIRE: 16 SCORE: 1.00 SCORE: 1.000 Figure 4-6. HIERARCHY GRAPHICAL DISPLAY

• DETACC • •S= 0.335• •RULE: HA• WHICH? 50 ENTER BOX NAME -- DETRCC HIERARCHY INFORMATION FOR BOX DETACC • REPALR • •S= 0.363•• •RULE: HA• • SENSE • •RULE: SD• ******** ******** - ANALRM S= 0.923 -RULE: 30 ٠ ٠ - CASSAS - 3.333 - RULE: AV • INDM1 • ••S= 0.575•• •RULE: SA• • INDA1 • •S= 0.540• •RULE: SA• - MULTA S= 0.570 RULE: AV • MULTS ••S= 0.671 •RULE: AV ŧ ******** ********* ********* ******* ******* * BARR ***S= 0.370 RULE: SA • • BARR • ••••S= 0.370• •RULE: SA• 37 S= 1.000 RULE: AV • GP •••S= 1.000 •RULE: AV• • DDS • •RULE: • ******** ******** ********* ********* ******* ******** ********* ********

.

4-22

4-7 (not produced by the computer) illustrates the level by level evaluation taking place for the entire capability.

The following discussion illustrates a hypothetical analysis of how the computer output might aid an evaluator or designer in recommending upgrade procedures for a facility. Let us assume that it is desired to upgrade the facility capability score from a .4 to a .5 by improving the "deny access" portion of the hierarchy. The upper portions of the hierarchy evaluation involve AND type interactions. From descriptions in Section 2, (and also from formal computations with the algorithm formula in Appendix A1,, the greatest improvement to an overall score per unit improvement of scores directly beneath it comes from improving the worst score when an AND type operation is involved. In looking at Figure 4-7, we first look at improving the DETACC score (the smaller of the two scores immediately below DENACC). We then look at improving the SENSE and ASSESS boxes to at least the same level as the REPALR box.

Figure 4-5 shows that the ASSESS box is evaluated by averaging the scores of the direct and indirect assessment techniques with the alarm station score. The alarm station score (#12) appears quite low. In addition, the techniques only average together reflecting a not especially well coordinated group of elements in performing the assessment task.

The questionnaire for the alarm stations (#12), indicates that the responses to several of the questions are at their worst levels. If these were changed to their best levels, the component would score .802 instead of .338. If, in addition, the ASSESS elements were coordinated so that a soft OR were appropriate, the total ASSESS score would become .838.

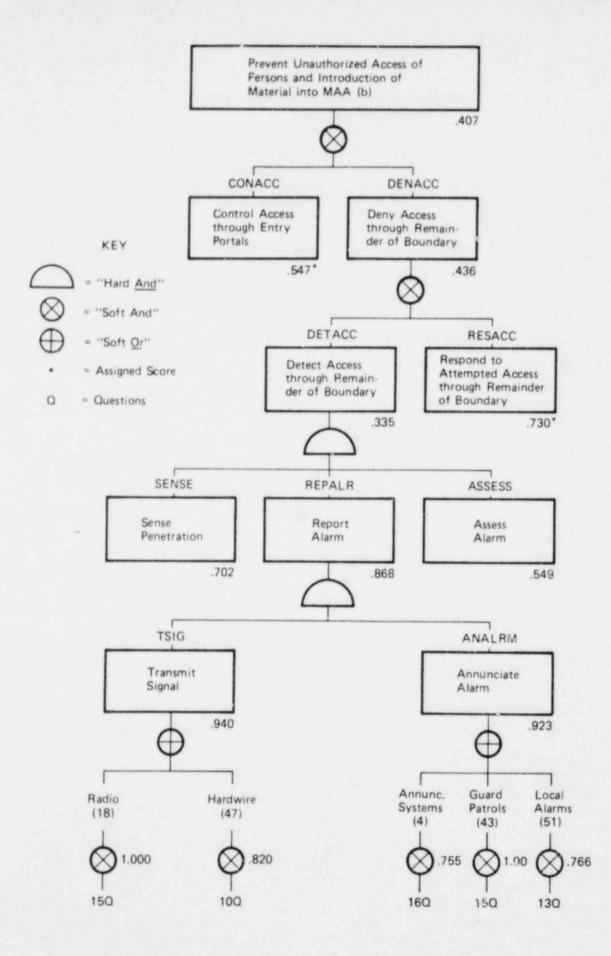


Figure 7. AGGREGATION EXAMPLE

In examining the SENSE box, we note two techniques are used. Since the SENSE box is evaluated using an OR rather than an AND, the most improvement is gained by improving the best technique beneath it rather than the worst. This refers to the box labeled MULTS. In Figure 4-5, MULTS is evaluated by averaging a group of components. This reflects the situation that the components do not especially provide defense in depth; e.g., when one component is active another is not. If these components coordinated together more closely, (e.g., all were always active providing some defense in the depth), so that a soft OR were appropriate, the score for MULTS would become .808. Finally, if the direct and indirect assessment techniques were made to provide completely independent, redundant systems so that a hard OR were appropriate, the total score for the SENSE box would be .918.

Working the improved scores up through the hierarch yields improved scores for the following elements:

SENSE: .918 ASSESS: .838 DETACC: .668 DENACC: .629 Capability: .506

The above discussion and example serves to illustrate how the algorithm is implemented using the computer program and how the resulting output can be analyzed to provide insight into and to explore different safeguard strategies. SUMMARY AND RECOMMENDATIONS FOR FUTURE WORK

5.0

5.1 STRENGTHS OF THE METHODOLOGY

The evaluation methodology presented in this report was designed to have certain desirable properties as outlined in Section 2.1. These strengths of the methodology are now summarized.

- Defensibility of Computational Formulas: The computational rules that are used are not arbitrary. They can be derived formally from a set of assumptions that allow the problem to be decomposed into simpler parts and then logically connected together. Whether the assumptions that justify the use of the computational formulas are valid for a particular component questionnaire or capability hierarchy depends a great deal upon the nature and interpretation of all the elements and factors involved. Because the formulas can be related to specific assumptions, however, the motivation for choosing a particular aggregation rule in a given situation can be explained. The rules have a basis in both probability and utility theory. Both theories provide orientations that appear useful for the safeguards evaluation problem.
- Flexibility of Aggregation Rules: The algorithms are capable of modeling several types of interactions using relatively simple functional forms. The rules were developed to address all the elements of a capability hierarchy including component questionnaires, component combinations, delay-response elements and higher level hierarchy elements.

5-1

- <u>Practicality</u>: The methodology can be implemented in a practical fashion using a framework consisting of multiple choice questionnaires and a hierarchy structure. The effort required to specify the parameters of the algorithms is reasonable. Both interaction rules and weighting factors can be assigned in a practical and systematic manner with a provision for some consistency checking.
- <u>Traceability</u>: The computer program allows a user to trace an overall capability score all the way down to a score on an individual component or question. Thus, the reasons why s facility received a particular score can be specified. The computer program has interactive capabilities that especially facilitate such tracing displays.
- <u>Aid to Evaluators</u>: In addition to traceability, the computer program can aid evaluators via sensitivity analysis. By varying the algorithm parame ers and/or element scores, evaluators can examine the range of assumptions under which a facility receives an overall score within a certain range, or for which it is less "preferred" than another facility. This type of analysis can provide useful insight into the critical elements affecting a facility's overall evaluation.
- <u>Aid to Designers</u>: Sensitivity analysis can also aid designers. By testing out combinations of potential components, or by adding or upgrading components, a designer can gain insight into what the strengths and weaknesses of a design might be and into what changes can most improve the design score.

In summary, the methodology has the potential for providing, in a practical manner, useful evaluations of safeguards facilities to both evaluators and designers.

5.2 LIMITATIONS OF THE METHODOLOCY

The methodology has limitations stemming from the complexity of safeguards evaluation. Limitations of the methodology and approach are now summarized.

- Multiple-Choice Questionnaire, Capability Hierarchy Framework: An approach selected to characterize a facility should be practical to implement and yet still reflect in a reasonable way essential features of the safeguards system. The questionnaire hierarchy framework is an approach that addresses the evaluation problem, not from the standpoint of providing a detailed model or simulation of a safeguards operation, but rather of providing a large "checklist" of aspects that should be examined in inspecting a design cr facility. Information that can be obtained via questionnaires is very diverse in both subject matter and precision. This makes it difficult to handle such questionnaires in a systematic, quantitative fashion. This should be remembered when interpreting an evaluation score. While giving useful insight into a facility, such an evaluation should not be the sole input to a complete safequards capability evaluation. It should not be expected that this particular approach could address all the complex features of a facility and potential adversary actions.
- Methodology Assumptions: In order to quantitatively evaluate a hierarchy capability, several simplifying assumptions were made in developing the aggregation algorithms. It must be recognized that these assumptions are approximate at best. Careful design of questionnaires and hierarchy structure can help make these assumptions more reasonable approximations. The methodology, like the questionnaire-hierarchy framework, has had to consider the compromise between practical implementation and the ability to address complex features of the problem.

 Methodology Implementation: The current capability hierarchies and questionnaires and the methodology itself have had very little testing in terms of sample or hypothetical facility information. The preliminary testing that has been done has served to point out areas for improvement, further implications of setting certain algorithm parameters and potential redundancies or double-counting in both questionnaires and hierarchies. This issue will be discussed further in the paragraphs to follow.

5.3 RECOMMENDATIONS FOR FUTURE WORK

While it is difficult to address all of the limitations discussed above, there are several areas where additional work could improve the framework and methodology presented in this report. These recommendations are discussed below.

- Design Methodology Testing: Further testing of the methodology is strongly recommended. Availability of the computer program will greatly facilitate this testing. The testing should include:
 - Evaluating component questionnaires and deciding what question weights and interaction rule can most reasonably reflect the effectiveness of individual components.
 - Evaluating component combinations and devising, if necessary, improved system questionnaires to deal with this issue.
 - 3) Evaluating higher level system tasks, whether the current hierarchy structures are appropriate and what interaction coefficients should be assigned to higher levels.

Formal assessments and consistency checking of algorithm parameters is recommended as part of this testing.

This evaluation and testing should be carried out under as realistic conditions as possible. Several existing facilities may provide useful data to help evaluate whether computed scores correspond to an intuitive evaluation of the quality of safeguards. After this testing procedure has been completed, an effort should be made to further simplify the use of the methodology for designers and evaluators.

• Extending the Methodology to Evaluate Overall Safeguards Capabilities: As was mentioned in Section 2.5, the current hierarchy structure does not reflect the concept of several safeguard capabilities working together to provide more complete safeguards than an individual capability considered in isolation could provide. There are possibilities for using the questionnaire information assessed for individual capabilities to evaluate overall safeguards. These may involve restructuring hierarchies especially for this purpose and also devising questionnaires that consider how capabilities coordinate with each other. Since some safeguard systems may be designed with an overall concept in mind, it would be useful to be able to evaluate overall safeguards in addition to individual capabilities.

6.0 GLOSSARY

Disaggregation Structure: The capabilities hierarchy that specifies required safeguard capabilities and defines the functions that a system must perform in order to meet these capabilities (see Figure 6-1).

<u>Performance Capabilities</u>: The five major objectives specified by the NRC that form the highest level of the disaggregation structure.

System Functions: The second level of the disaggregation structure. Those functions that contribute directly to a system capability.

System Subfunctions: All disaggregation levels below the system functions (excluding the lowest levels), which identify specific tasks to be performed by the system.

Low-level System Tasks: Those specific tasks or jobs which follow from the system subfunctions and which correspond to the Performance Characteristics in the Component Selection Matrix-

Performance Characteristics: Constrained low-level system tasks.

Component Selection Matrices: Matrices that indicate correlations between specific tasks via performance characteristics and feasible approaches (equipment and procedures).

SSNM: Strategic Special Nuclear Material.

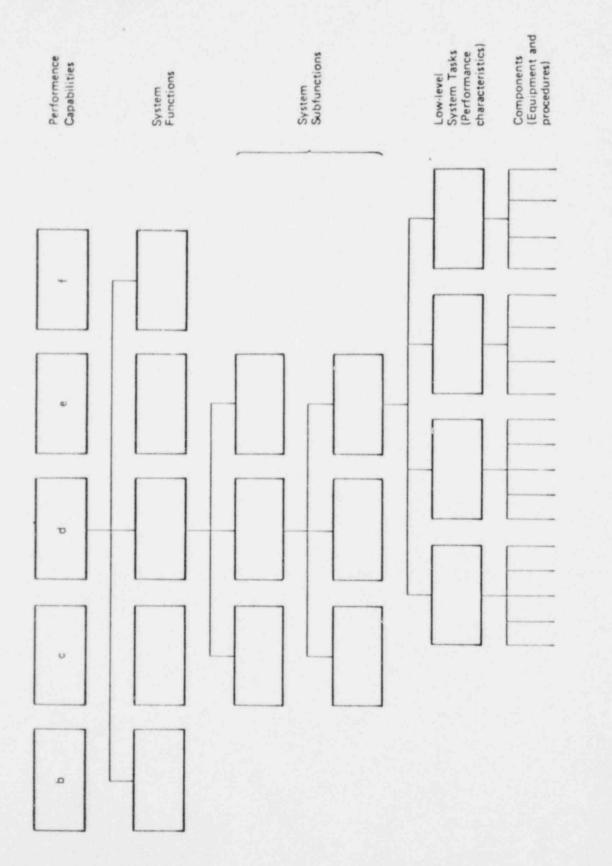


Figure 6-1. DISAGGREGATION STRUCTURE

*

MAA: Materials Access Area. Any area containing SSNM.

VA: Vital Area. Any area containing equipment that could be sabotaged.

<u>PA</u>: Protected Area. Area surrounding a VA and/or MAA, up to the perimeter of the facility.

<u>ETQ</u>: Effectiveness Test Questionnaire. A set of multiple choice questions that provides information used by the algorithm to evaluate an overall component effectiveness score.

IRS: Information Request Sheet. A more expansive description of a component or safeguard system that provides additional information to evaluators. The algorithm uses only the multiple choice responses and direct scoring information provided by the evaluator in its computations.

System Effectiveness Test Questionnaire: A set of multiple choice questions that provides information used by the algorithm to decide what type of computation is appropriate to combine certain component scores. The questionnaire may also provide the information necessary to combine delay-response type scores, treat multiple access points and factor in facility information not covered in any particlar component questionnaire.

Delay-Response Type Elements: Elements that require a synthesis of delay and response/assessment/monitoring capabilities by comparing delay and response/assessment/monitoring times.

6-3

REFERENCES

- Woodward-Clyde Consultants. A Methodology for Evaluating Safeguards Capability for Licensed Nuclear Facilities. Prepared for Sandia Laboratories, December 1978
- Keeney, R.L. and H. Raiffa, <u>Decisions with Multiple Objectives</u>. New York: Wiley, 1976

Appendix Al FUNCTIONAL FORM FOR SAFEGUARDS EVALUATION ALGORITHM

In Section 2.2, the utility function structure used in safeguards evaluation computations was briefly presented. In this appendix, the underlying assumptions made in that structure are described. In addition, further properties of the evaluation function are discussed.

Al.1 Utility Theory

The axioms of decision analysis [1] define a formal logic for evaluating alternatives where the consequences of those alternatives may be uncertain. Specifically, the assumptions utilized in this study imply the existence of a utility function to model the preferences of the evaluators.

Before stating the axioms of utility theory, we define our notation. A <u>simple lottery</u>, written $L(x_1,p,x)$, is a probabilistic event characterized by two possible consequences, which will be designated by x_1 and x_2 , and by their respective probabilities of occurrence, designated by p and 1-p. The symbols >, ~, and < will be read "is preferred to, " "is indifferent to," and "is less preferred than," respectively.¹ Thus, $X_1 \sim L(x_2,p,x_3)$ says that x_1 is indifferent to the lottery which yields either x_2 with probability p or x_3 with probability 1-p.

¹These designations are only for section Al.1.

The axioms stated here which imply the existence of a utility function are only slightly modified from the formulation of Pratt, Raiffa, and Schlaifer [1].

Axiom ul: Existence of Relative Preferences. For every pair of consequences x_1 and x_2 , preferences exist such that either $x_1 \sim x_2$, $x_1 > x_2$, or $x_2 < x_1$.

Axiom u2: Transitivity. For any lotteries L_1 , L_2 , and L_3 , the following hold:

i) $L_1 \sim L_2$ and $L_2 \sim L_3$ implies that $L_1 \sim L_3$

ii) $L_1 > L_2$ and $L_2 \sim L_3$ implies that $L_1 > L_3$, etc.

Since a consequence can be interpreted as a degenerate lottery (i.e., p = 1), axioms ul and u2 together imply the existence of a ranking of the relative desirabilities of the various possible consequences. They do not say that an individual can articulate this, nor do they require that this ranking be stationary over time. Let us designate as x° a prosequence which is not preferred to any of the other consequence for a problem and as * a consequence which is at least as preferrous each of the other consequences. Therefore, one possiblity is that x° and x* designate the least and most preferred consequences, although they may represent hypothetical consequences such that x* > x and x > x° for all possible x.

Axiom u3. Comparison of Simple Lotteries. Given the preference order $x_1 > x_2$, then

i) $L_1(x_1,p_1,x_2) > L_2(x_1,p_2,x_2)$ if $p_1 > p_2$, ii) $L_1(x_1,p_1,x_2) \sim L_2(x_1,p_2,x_2)$ if $p_1 = p_2$. Axiom u4. Quantification of Preference. For each possible consequence x, the evaluator can specify a number $\pi(x)$, where

 $0 \le \pi(x) \le 1$, such that $x \sim L(x^*, \pi(x), x^\circ)$.

Axioms u3 and u4 taken together establish a measure of the relative desirabilities of the various consequences to the evaluator. The $\pi(x)$ value - or indifference probability, as it is called - is that measure.

Clearly, since the standards x° and x^{*} for measuring $\pi(x)$ are somewhat arbitrary, different π functions may be assess or a specific individual in a particular situation. To be consistent with these axioms, however, all possible functions must be positive linear transformations of each other. Any positive linear transformation of π of the form

 $u(x) = a + b\pi(x), b > 0$

is referred to as a utility function. The quantity u(x) is said to be the utility of consequence x. If one accepts the above axioms, one should always prefer alternatives that maximize expected utility. There are no alternative procedures for making decisions consistent with these axioms.

Since maximizing expected utility is equivalent to maximizing the expected value of w, the arbitrary choice of x* and x° has no influence on the actual decision. Utility provides a relative scale analogous to the temperature scales, and two scales which are positive linear transformations of each other are identical for decision-making perposes.

A1.2 INDEPENDENCE ASSUMPTIONS

The theory underlying multiattribute utility functions is presented in detail in [2]. Here we briefly present the theoretical results that underlie the utility functions used in the evaluation methodology presented this report. Suppose $\{X_1, X_2, \ldots, X_N\}$ are the attributes of an evaluation problem. For notational convenience let X_I be any specified subset of $\{X_1, X_2, \ldots, X_N\}$ and \overline{X}_I be the complement of X_I . Then X_I is <u>utility independent</u> of \overline{X}_I if preferences for risky choices (lotteries) over X_I with the value of \overline{X} held fixed do not depend on the fixed value of \overline{X}_I . If X_I is utility independent of \overline{X}_I for all X_I then $\{X_1, X_2, \ldots, X_N\}$ are mutually utility independent.

$$u(x_1, x_2, ..., x_N) = \sum_{n=1}^{N} k_n u_n(x_n)$$
 (A1.1a)

or

$$1+Ku(x_1, x_2, ..., x_N) = \prod_{n=1}^{N} [1+Kk_n u_n(x_n)]$$
 (A1.1b)

where x_n represents a specific value of X_n , u and the u_n 's are utility functions scaled from zero to one, the k_n 's are scaling constants with 0 < k < 1, and K>-1 is a nonzero scaling constant which is the solution to

$$K + 1 = \prod_{n=1}^{N} (Kk_n + 1).$$
 (A1.1c)

Note that in the derivation of the utility function for the safeguards evaluation, some or all of the attributes may be vectors.

In Section 2.2, the expanded expression for the utility function is given with the following relationships:

> $S = u (x_1, x_2, \dots, x_n)$ $C = k_n \text{ for all N attributes}$ $u_n = S_n \text{ for all N attributes}$

Thus, in simplifying the expression for the aggregation algorithm,

$$S = \sum_{i=1}^{N} S_{i}/N \quad \text{or} \quad (A1.2)$$

$$S = \prod_{i=1}^{N} (1 + VS_{i}) -1 \quad (A1.3)$$

$$\frac{i=1}{(1 + V)^{N} -1}$$

A1.3 SYSTEM QUESTIONNAIRES FOR DETERMINING CONDITIONAL UTILITY FUNCTIONS

The parameter V in (Al.3) can be considered to be conditional upon other factors that are assumed fixed for a particular evaluation. If such factors change, it is possible that V might change. A system questionnaire provides the mechanism for assigning V conditional upon how well a group of elements coordinate with one another.

Specifically, in the Phase 1 report in Appendix A3 [3], it was shown how the interaction coefficient was related to the value cf Y that would make a decision maker indifferent between the following: A. Element 1, Element 2 $(S_1=1, S_2=0)$ B. Element 1, Element 2 (S₁=Y, S₂=.5)

It was shown there, that V=(1/Y)-2

A value of Y near 1 would indicate that S_1 and S_2 were redundant or substitutes for each other. A value of Y near 0 would show that a fatal flaw in either one would cause the other to be almost worthless. A system questionnaire provides a mechanism for computing more formally what value of Y a decision maker would feel to be appropriate for the above tradeoff. Specifically, Y is set equal to the score on the system questionnaire. To simplify the algorithm, the following correspondences between Y and V are used:

Y	V	
.8-1.	V=-1	(OR)
.68	V = -1/2	(SOFT OR)
.46	V=O	(AVERAGE)
.24	V=1	(SOFT AND)
02	V≖∞	(AND)

In summary, V is conditional upon the score of the system questionnaire in the manner indcated above. In these instances, (A1.3) with the appropriate value of V is a conditional evaluation or utility function based on howe well the elements coordinate with each other as determined by the system questionnaire.

A1.4 SOME PROPERTIES OF THE FUNCTIONAL FORM

If we take the partial derivative of S with respect to S_i in (A1.2) and (A1.3) we obtain the following:

$$\frac{\partial S}{\partial S_{i}} = \frac{1}{N} \quad \text{for } V=0$$

$$\frac{\partial S}{\partial S_{i}} = \left(\prod_{\substack{j=1\\ j\neq i}}^{N} (1+V S_{j}) \right) * V/K$$
(A1.4)
(A1.5)

In (A1.5) the quality V/K is always positive. In examining (A1.5), when $-1 \le V \le 0$, the product term is largest when the S_i excluded is the largest. This implies that for an OR type aggregation, the greatest gain per unit increase in an S_i is achieved when the largest S_i element is selected. In other words, to upgrade a combination of components which OR together, it would pay to improve the best element further, if possible. When $0 < V < \infty$, the product term is largest when the S_i excluded is the smallest. Thus for an AND type aggregation, it would "pay" to improve the worst element further, if possible.¹

Of course, there are other factors which would influence those elements to be selected for potential upgrading; e.g., cost, feasibility of improvement, etc., But the algorithm's functional form does provide some guidance on this issue.

¹In this sense, the maximum and minimum element scores have the greatest influence respectively on OR and AND type aggregations. But, unlike using a MAX or MIN operator, their influence need not completely override the contribution of the other elements.

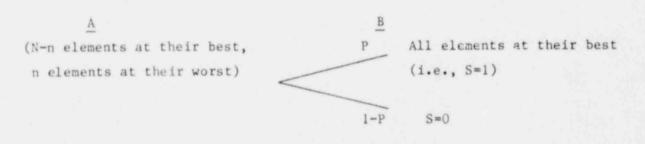
REFERENCES

- Pratt, J.W., Raiffa, H., and Schlaifer., R. 1964. The Foundations of Decision under Uncertainty: an Elementary Exposition. Journal of the American Statistical Association 59:353-357.
- Keeney, K.L. and H. Raiffa, Decisions with Multiple Objectives; Preferences and Value Tradeoffs. New York: Wiley, 1976
- Woodward-Cl/de Consultants, A Methodology for Evaluating Safeguards Capability for Licensed Nuclear Facilities, prepared for Sandia Laboratories, December 1978.

Appendix A2 ASSESSING ALGORITHM PARAMETERS

In Section 3.5, a technique was presented with which the parameters of the algorithm could be assessed in a consistent manner. The derivation of the formulas shown in that section are now presented.

To review, to calibrate the weight W for a group of equally weighted elements in a manner consistent with the selected aggregation coefficient, one should choose n and P for which the followign two situations are equally preferred:



In practice, P is sometimes set to .5, and n is then determined. The formulas for u in terms of N,n and P are now derived.

The basic equations used are the formulas (A1.2) and (A1.3). As appendix Al explained, the expected utility or score for situations A and B above should be set equal to each other since they are equally preferred. In terms of formula (A1.2), the expected utility for situation A is equal to (N - n + n(1-W))/N. The expected utility for situation B is P. Equating these two scores, yields:

W = (1-P)N/n

There is a transformation of formula (A1.3) that is useful in deriving expressions for W. It is a noormalization scheme that has the scores going from -1 to 0 rather than 0 to 1. To make this transformation, the following quantities are defined:

> V' = V/(1+V) + V = V'/(1-V') $S_i' = S_i-1 + S_i = 1 + S_i'$ S' = S-1 + S = 1 + S'

In terms of the new quantities, (using formula (Al.3) and reducing),

$$S' = \prod_{\substack{i=1 \\ 1-(1-V')^{N}}}^{N} (1 + V'S_{i}') - 1$$
(A2.1)

Both formulas (A1.3) and (A2.1) will now be used to derive expressions for W.

The expected utility for B in the new variable system is P(0)+(1-P)(-1) or P-1. The expected utility for A using formula (A2.1) is simply:

$$\frac{(1-V'W)^n - 1}{1-(1-V')^N}$$
 (A2.2) $-\infty < V' \le 1, V' \ne 0$

(S_i' i r a perfect element is 0 in the new system). Equating (A2.2) with P-1 yields the following fo. W:

$$W = (1 - (P + (1 - F)(1 - V')^{N})^{1/n} / V'$$
(A2.3)

The formula in the original variable system is shown in Section 3.5.

For $V + \infty$, V' + 1, and formula (A2.3) becomes $W = 1 - p^{1/n}$.

For V=1, V'=1/2 (soft AND). For N large, the term containing $(1-V')^N$ is multiplied by $(1/2)^N$ and can be ignored in (A2.3) for N>5.

For V=-1 (hard OR), the use of formula (A1.3) yields an expected utility for A of $1-W^N$ for the case where all elements are at their worst. (Even one at its best will yield a utility of 1). Solving for W in this case yields $W=(1-P)^{1/N}$.

APPENDIX A3 COMPUTER PROGRAM DOCUMENTATION

A3.1 GENERAL OVERVIEW

Two versions of the safeguards evaluation computer program were written in FORTRAN IV. This appendix contains listings of both versions. An interactive program design was chosen because (1) it facilitates easy use of the program by providing easy to understand dialogue and (2) it greatly facilitates sensitivity analysis as well as fine tuning of algorithm parameters. The first version has more interactive options than the second, but may require more modifications to implement on non-IBM systems. This version is not strictly 1966 ANSI standard[1]. However, it has been recognized that the 1966 standard is quite limited in certain respects. Specifically, no random access input-output is allowed by the 1966 standards. A new standard (FORTRAN 77)[2] has been published, but as yet is not implemented on most systems.

The second version is less interactive and replaces all the random access input-output statements with sequential input-output statements. This second version is intended to comply with the 1966 standard and should be easy to implement most systems. It has the following limitations: Options 21 and 48 may only be done once in a run and options 22 through 29 are not available.

A3.2 LANGUAGE AND OPERATING SYSTEM CHARACTERISTICS

Both programs do not require large amounts of core, but do utilize four disk files for input and output as explained in Section 4 of the main report. Currently, there is room for 100 questionnaires with up to 40 questions each, and 40 "boxes" per capability. These dimensions can be changed if necessary. With the current dimensions, it is sometimes necessary to evaluate an entire capability in two pieces. This was done for the example discussed in Section 4 and presented in more detail in Appendix A4.

Some features that may require modification when implementing the program on different systems are now discussed. DOUBLE PRECISION statements are necessary to accommodate label variables of more than 4 characters on IBM machines. DOUBLE PRECISION statements should be removed for CDC machines. All random access input-output related statements are usually different for different computer systems. The current fully interactive version is IBM compatible only. The hierarchy graphical display routine utilizes certain carriage control characters that may be different or not present on some systems. The graphical display option 50 should not be used on systems that do not support a control character that allows the printer to remain on the same line with no carriage return and no line feed. Finally, CDC systems require a PROGRAM CARD to appear as the first card in the main program.

The mnemonic QUEAS (Quantitative Evaluation of safeguards) is given to the program files which contain the routines implementing the algorithm. The program routines and data files are described in Table A3-1. The program listings follows the table. The routines with a "B" appended to the file name replace their counterparts to compose version B of the program.

REFERENCES

- American Standard FORTRAN, New York: American Standards Association, Inc., 1966.
- Harry Katzan Jr., FORTRAN 77. New York: Van Nostrand Reinhold Company, 1978.

Table A3-1. PROGRAM ROUTINES AND DATA FILES

QUEASI (FORTRAN): Main Program File

QUEAS1 (FORTRAN): First Subroutine File

SUBROUTINE GETHN : Interactively requests a nierarchy box name and returns the corresponding number

SUBROUTINE ATGET : Accepts a hierarchy box name and returns the corresponding number

SUBROUTINE SCREH : Accepts a box number and returns the box score

SUBROUTINE BOX : Computes score for a box with questionnaires as input

SUBROUTINE MULTEV: First level of recursive scoring routine. Tries to compute box score

SUBROUTINE GETQN : Interactively requests a questionnaire name and returns the corresponding number

FUNCTION YESNO : Accepts an interactive answer to a yes/no question and returns 0 for no, 1 for yes

FUNCTION GETNUM : Accepts a number from within a specific range interactively

SUBROUTINE TXTURE: Combines a group of numbers using a scoring rule

SUBROUTINE MULTE1: Second level of recursive box scoring routine SUBROUTINE MULTE2: Third level of recursive box scoring routine SUBROUTINE MULTE3: Fourth level of recursive box scoring routine SUBROUTINE MULTE4: Fifth level of recursive box scoring routine SUBROUTINE MULTE5: Final level of recursive box scoring routine SUBROUTINE GETR : Reads a questionnaire's responses

QUEAS2 (FORTRAN): Second Subroutine File

SUBROUTINE PRINTH: Prints short version of hierarchy structure starting with specified box

SUBROUTINE SETH : Resets higher score values when a box score is set Table A3-1. (Continued)

FUNCTION NQ : Returns the questionnaire number of a supplied name

QUEAS3 (FORTRAN): Third Subroutine File

FUNCTION SCOREQ : Returns the computed score of a questionnaire SUBROUTINE GETQ : Retrieves questionnaire structure SUBROUTINE PRINTQ: Prints detailed questionnaire data

QUEAS4 (FORTRAN): Fourth Subroutine File

SUBROUTINE HPR : Prints nievarchy diagram SUBROUTINE PNAME : Prints names of hierarchy boxes SUBROUTINE PQNAME: Prints names of questionnaires

- FILE1 (DATA): Questionnaire structure data
- FILE2 (DATA): Questionnaire scores

FILE3 (DATA): Hierarchy structure data

FILE4 (DATA): Initial hierarchy scores

QUEASI	FORTRAN P IDEVCCUR 16.19.58 THURSDAY 6 DECEMBER 1979	PAGE 1
NAT	IONAL CSS. INC. (SUNNYVALE DATA CENTER) SUNY	
C QUEAS	SI A PROGRAM FOR EVALUATION OF SAFEGUARDS QUESTIONNAIRES	QUEDOCIO
C ANC S	TERARCHIES	QUE00020
	COMMON LOCG(100) . OSCORE (100) . LOCR (100)	00500030
	COMMON SCOREH(40) + RULEH(40) + IDEX(40+10) + QDEX(40) + QNAME(100)	QUEDCOAC
	DOUBLE PRECISION BNAME (40) BLANK , NOMO , HNAME (40) . ANAME	PUE00050
	INTEGER LOCHS(5) . ISET(40) .FLAG(10) .LOCH(5) . IRESP(40) . IREST(40)	OUEDCOLC
	REAL SCORE (40) .WEIGHT (40) .RULE (10) .TEXTS(7)	QUEDOC70
	DATA TEXTS/2HHA, 2HSA, 2HAV, 2HSO, 2HOR, 1HQ, 2HDR/	CRODOJUA
	DATA IAA/1HA/.NOMO/6HNOMORE/BLANK/6H /	QUED0090
	OEFINE FILE 1 (300.00.E.19)	QUEDC1CD
	CEFINE FILE 3 (200.80.E.19)	QUEDC110
	DEFINE FILE 2 (150.80.E.19)	QUEOC120
		QUE 0130
	AAA=FLCAT(IAA)	QUEDC140
	DO 399 I=1.40	QUEDD150
395	SCCREH(I)=-1.	QUEDG160
	J= 2	QUECC170
C REAL	QUESTIONNAIRE LOCATIONS AND NAMES	QUEDDIRO
	READ(1*1+9464) NUMG	QUE00190
	READ(1.),9465) (LOCG(11),11=1,NUMG)	QUECC202
	READ(2*J.9465) (LOCR(12), 12=1, NUMQ)	QUED0210
	CC 30 1=1.NUMG	QUE00220
	J=LOCG(I)	QUE00230
30	READ(1*J+9272) QNAME(1)	QUECC240
	DC 10 J=1.100	QUEP3250
10	OSCORE (I)=-1.	QUE00260
	DC 20 I=1.10	QUECO270
20	FLAG(I)=0	QUE DO2EO
C SELE	CT INITIAL OPTION	QUECC290
1000	WRITE(6,910C)	QUESSOC
9100	FORMAT(54H? SELECT 1- HIERARCHIES, 2- QUESTIONAIRES, 3- STOP)QUEDO31C
	1CF=GETNUM(1.+3.+2.)	QUE00320
1001	IF (IDP.EG.3) STOP	QUE00330
	GOTO (4000,2000) ICP	QUECC340
C REVI	EW OPTION SELECTION AND BRANCH TO PROPER OPTION	QUE00350
	1CFT=10P	QUED0360
1100	CONTINUE	QUE 00370
	1F (10PT.6E.1.AND.10PT.LE.3) 6070 1101	QUEDC380
	IF (IOP T. 6E. 21. AND. IOPT.LE. 26) GOTO 1102	QUE 00390
	IF (IOPT.GE.41.AND.IOPT.LE.50) GOTO 1102	CC4002UD
	GCTC (1000,230,1000,402) 10P	QUE00410
1101	ICPEICPT	QUEDD420
	6070 1001	QUEDD431
1102	IOP=INT(FLOAT(IOPT)/10.)	QUEDC440
	IF(IOP+10.EG.ICPT) 60TC 1CC1	QUECDASC
	ICPT=ICPT-ICP+10	GUED0460
	GCTO (1000,201,1000,404) ICP	QUECC470
	GOTO 1000	QUE00480
C PRIN	T MENU AND GET QUESTIONNAIRE OPTION	QUEOC49C
2000	IF (FLAG(2).EQ.1) GOTO 210	QUEDOSOO
230	WRITE(6,9200)	QUEC0510
	WRITE(6,9201)	QUE00520
	WRITE(6,9202)	QUE00530
	WR ITE (6,9203)	QUE00540
	WRITE(6,9204)	QUECOSSO

91	EASI	FORTRAN	P ID=WCCWR	16.19.58	THURSDAY 6	DECEMBER	1979	PAGE 2
		WRITE (6.32	05) 4H SELECT ON 21- COMPUTE 23- SET SCOM 25- REVISE F 27- REVISE F 29- H? WHICH?)					011500560
	200	FORMATCAL	AH SELECT ON	: ./)				QUE 30570
9	201	FCRMAT(51H	21- COMPUTE	SCORES	22. PO THT	STOPES		OUFCOSCO
9	202	FORMAT (51H	23. SET SCO	DEC	24- PEVISE	HEIGHTS		DUEDDESD
9	203	FORMAT (51H	25- REVISE F	UL FS	26- REVISE	PESPONSES	í.	QUEDCEDO
9	207	FORMAT (51H	27- REVISE	NAMES	28- PRINT	NAMES	,	AUF 60610
9	204	FORMAT (51H	29.	NO MORE RE	VISIONS)	QUEC0620
9	205	FCRMAT(/ .9)	H? WHICH?)					QUEC0630
		GCT0 220						QUE 20640
	210	WR ITE (6,92)	06)					0.100650
9	206	FORMAT (18H	? SELECT 21-29))				QUECC660
	221	ICFT=GETNUM	*(21.,29.,2.)					QUEDC670
		IF (IOPT.L	27- REVISE 1 25- H? WHICH?) 06) ? SELECT 21-29 M(21.,29.,2.) T.21.OR.IOPT.6 20	T.29) GOTO	1100			QUECC680
		ICPT=ICPT-	20					QUE00690
	201	FLAG(2)=1	and the second second					QUECC700
C	ERAN	CH TO PROPER	OPTION					QUEDC71C
		ECTO (2001	2002+2003+200	4.2005.2006	.2067.2008.1	10CC) ICPT		QUE00720
				a a dimensionale				QUEOC730
6		FOTO 222 10	PRINT DATA FO	IR A QUESTIO	NNAIRE			QUEDD74C
~	223	0010 222	NUM					00200750
		CILL DETATO	NUMU NUMU					00200760
	221	CONTINUE	ILI FURANCI					QUECOTTO
		50T0 2000						011500700
	222	CALL SETON	ID . ONAME . NUMO	11)				QUEADSAD
		IF (ID.EQ)) GCTO 223					AUFOORIC
		CALL PRINTO	(ID. QNAPE)					QUE00820
		GCT0 2000	H? WHICH?) 06) 2 SELECT 21-25 M(21.,29.,2.) T.21.OR.IOPT.0 20 R OPTION 2002.2003.200 PRINT DATA FO NUMG					QUEDORSO
								QUEDO840
С	69	TION 21 TO	COMPUTE A QUE	STIONNAIRE S	SCORE			QUECCRSO
2	103	GCTO 213						QUEDOS60
	214	DC 211 I=1,	NUMG					QUE00870
1.0		WRITE (6.930	1) GNAME(I)					QUEODBRO
9	101	FORMAT (16H?	QUESTIONNAIR	E +1A4+1H:)				QUECO890
1	221	CALL SCOREG	(1)					ODEC0300
		6010 2:00						QUE00910
	113	CALL BEIGN C	10 .GNAME .NUMG	0)				QUE00920
		IFCID.EG.=1	3 6010 214					QUED0930
		CALL SCURLU	(10)					QUE00940
		6010 2000						QUEDC950
c .		TION 23 TO						GUED0960
23	0.2	CALL GETONI	TD. ONAME. NUMO	INAINE SCURE				QUE DOGOD
- 11		WRITEL6.930	0)	.,				0020098C
93	00	FORMAT (30H?	ENTER QUESTI	NATRE SCORE	1			QUEDICLO
		GSCORE(ID) =	GETAUM(01	1.)				GUE01010
	1.1	GCTC 2000						GUE 31023
								QUE 01030
с -	- CP	TION 24 TO	REVISE QUESTI	ON WEIGHTS .	-			QUEG1040
20	04	CALL GETON .	ID . QNAME . NUMO	U)				QUE01050
		SCORE(ID) =	-1.					QUE01060
		WRITE (6,924						GUED1070
92	40 1	FCRMAT(39H?	NUMBER OF QUE	ESTIONS TO E	E REVISED -	-)		QUEC1080
			1. FLOAT (NUMAL	(.0.)				QUE01090
		rec=rece(1D	,					QUEG1100

QUEAS	I FORTRAN P ID=WCCWR 16.19.58 THURSDAY 6 DECEMBER 1979	PAGE 3
	READ(1*LOC.9500) R.NGGG.NGRP	
		QUEDILLC
	LCC=LOC+2+2+NGRP	QUE01120
	REAC(1 *LOC +9243) (WEIGHT(K) +K=1+8)	QUE01130
	IF (WEIGHT(1).LE.1) GOTO 241	GUE01140
N. Sanda	WRITE(6,9244)	QUE01150
9244	FORMAT (47H? MUST REVISE ALL WEIGHTS. ENTER NEW WEIGHT)	QUECIL60
	WE16HT (2)=6ETNUM (0.,1.,1.)	QUE01170
	WRITE(1*LOC.9243)(WEIGHT(K)*K=1.8)	QUECIIBC
	60T0 2000	QUEC1190
241	LOC=LOC+1	QUE01200
	IF (NUMQU.LE.8) GOTO 243	QUE01210
	READ(1*LOC,9243) (WEIGHT(K),K=9,NUMQU)	QUE01223
242	D0 240 I=1.NUM	QUE01230
	WRITE(6,9241)	QUE01240
9241	FORMAT (20H? QUESTION NUMBER =)	QUE01250
	NUMQ=GETNUM(1	QUE01260
	WRITE(6,9342)	QUEC1270
9242	FCRMAT(11H? WEIGHT # 0	QUE01280
	W=GETNL + (0.,1.,1.)	GUED1290
	WEIGHT (NUMQ) = W	QUE01300
9243	FORMAT (8F5.3)	GUEC1310
	LOC=LOCQ(ID)+2+NGRP+2	GUE1132C
	WRITE(1°LOC+9243)(WEIGHT(K),K=1,NUMQU)	GUE01330
	CALL SCOREG(ID)	QUED1340
	ECTO 2000	QUEC1350
		QUE01360
	CPTICN 25 TO REVISE SCORING RULES	QUE01370
2005	CALL GETON (ID, GNAME, NUMGU)	GJEC1380
	QSCORE(ID)==1.	QUE01390
	LOC=LOCG(ID)	QUEC1400
	READ(1+LOC+950C) R+NGGG+NGRP	QUE01413
	FORMAT(1A2+2(8X+112))	GUE01420
251		GUEC1433
9502	FCRMAT(24H? ENTER GROUP NUMBER)	QUE 1440
	N=GETNUM(50.,FLOAT(NGRP)+49.,0.)	QUE01450
	WRITE(6+9503)	QUEC1460
9503	FORMAT(33H? ENTER RULE (HA+SA+AV+SO+OR))	QUEC147C
254		QUE01480
9501	FCRMAT(1A2)	QUED1490
	DC 252 I=1.5	QUE01500
	IF (R.EG.TEXTS(I)) ECTO 253	GUEC151C
252	CONTINUE	QUE0152C
	WRITE(6,9505)	QUE01530
9505	FORMAT (25H? BAD RULE. TRY AGAIN)	QUE(1540
	60TC 254	GUE01550
253	L=LOC+2	QUE01560
	DU 255 I=1.NGRP	GUE01570
	READ(1*L.9506) 11 RP.M.Z	QUEC1580
9506	FORMAT (2(112.8X).1A2)	QUE 01590
	IF (IGRP.EQ.N) WRITE(1*L.9506) IGRP.M.R	QUEDIGOC
255		QUEDI610
	CALL SCOREG(ID)	QUE01620
	GCTO 2000	QUE 01630
		GUEU1643
C	CPTICN 26 TO REVISE QUESTION RESPONSES	QUECIESO
1.1		

3	CPTICN	26	TO.	REVISE	QUESTION	RESPONSES	
---	--------	----	-----	--------	----------	-----------	--

QUEASI	FORTRAN P IC=WCCWR 16.19.58 THURSDAY 6 DECEMBER 1979		PAGE 4
2404	CALL GETON(ID. GNAME. NUMQU)		QUE 01660
Zeve	QSCORE(ID)==1.		QUE01670
	LOC=LUCR(ID)+1		GUE01680
	READ(2*LOC.960C) IRESP		QUE01690
	FORMAT (40(1x,1A1))		QUE01760
9600			QUE01710
	READ(1*L+9600) IBEST		QUE 01720
	WRITE(6.9240)		QUE01730
	N=GETNUM(U.+40.+0.)		QUE01740
	DC 260 I=1.N		QUE01750
	KRITE(6.9241)		QUE 01760
	NUM=GETNUM (0400.)		QUE01770
261	WRITE(6.9603) IBEST(NUM)		QUE01780
	FORMAT(16H? RESPONSE (A TO+1X+1A1+4H) =)		QUEC1750
	READ(5.9260) IRESP(NUM)		QUE 31800
	FCRMAT(1A1)		QUE 01810
1200	x=1=(AAA=FLOAT(IRESP(NUM)))/(AAA=FLOAT(IBEST(NUM)))		QUE01520
	IF (x.LT.QOR.X.GT.1.) GCTO 261		GUEG1830
	WRITE (2*LOC.96CO) IRESF		QUED1840
	CALL SCOREQ(1D)		QUE01850
	GOTO 2000		QUE01860
			QUE01870
C == []	TICN 27 TO REVISE QUESTIONNAIRE NAMES		QUEC1880
	CALL GETON (ID, GNAME, NUMQU)		GUE01890
E 9 6 1	WRITE(6.9271)		QUE01960
9271	FORMAT(15H? ENTER NAME)		QUE01910
1212	READ(5.9272) QNAME(ID)		QUE01920
9272	FORMAT(1A4)		QUE01930
	GCT0 2000		GUE01940
			QUE01950
C PI	INT AND SELECT HIERARCHY MANIPULATION OPTIONS		QUE01960
	CALL PONAME (QNAME . NUMQ)		QUE 31970
	6010 2000		QUE01980
			QUE01993
0032	1F (FLAG(5).EQ.0) GOTC 4006		QUEC2000
	IF (FLAG(4).EQ.1) 60T0 401		GUE02010
402	WRITE(6.9400)		QUE02320
	WR ITE (6.94 31)		GUE02030
	WRITE(6,9402)		GUE 32 34 3
	WR ITE (6,9403)		QUE 02050
	NRITE(6,9404)		QUE02060
	WRITE(6.9435)		QUE02070
	WRITE(6.9407)		QUE02080
	WRITE(6,9205)		QUE02090
9400	FORMAT (/.14H SELECT ONE: ./)		QUE02100
9401	FORMAT (51H 41- COMPUTE SCORES 42- PRINT DATA)	QUEC2110
9432	FORMAT (51H 43- ASSIGN SCORES 44- REVISE DELAY/RESP)	QUE02120
	FORMAT(51H 45- REVISE RULES 46- SELECT NEW HIERARCHY)	WUE 02130
	FORMAT (51H 47- PRINT BOX NAMES 48- FILE HIERARCHY DATA)	GUE02140
9405	FORMAT(51H 49- CHANGE BOX NAME 50- PRINT HIERARCHY)	QUEC2156
9407	FCRMAT (51H 51- NO MORE REVISIONS)	QUE02160
	GCTC 4C3		QUE02170
401	WRITE(6,9406)		QUE02180
9406	FORMAT(18H? SELECT 41-51)		QUE 02190
463	IOPT=GETNUM(41472.)		GUE02200

QUEASI	FORTRAN P IC=WCCWR 16.19.58 THURSDAY 6 DECEMBER 1979	PAGE 5
	1F(10PT.LT.41.CR.IOPT.GT.51) GOTO 1100	QUE02210
	1CPT=10PT=40	QUE02220
	FLAG(4)=1	QUE 02230
	CH TO PROPER HIERARCHY OPTION	QUE02240
C BRAN	1F(FLAG(5).EQ.() 60TC 4006	GUE 02250
	6010(4001,4002,4003,4034,4005,4006,4007,4008,4009,4002,1000)	
		QUEG2270
C 0	PTION 41 TO SCORE & HIERARCHY BOX	QUE02280
	CALL GETHN(ID .HNAME)	QUE 02290
	CALL SCREH(ID)	QUED2300
	CALL SETH(ID.ISET)	QUE02310
	ISET(IC)=0	QUE02320
	GOTO 4000	QUE02330
		QUED2340
c c	PTICN 42 TO PRINT HIERARCHY DATA	QUE02350
4002	CALL GETHN(ID+HNAME)	QUE02360
	IF(IOPT.EQ.2) CALL PRINTH(ID. HNAME)	QUE02370
	IF (IOPT.EQ.10) CALL HPR(ID, HNAME)	QUES2380
	6010 4000	QUE02390
	the second second and second	QUE02400
	PTICN 43 TO ASSIGN HIERARCHY BOX SCORES	QUE32410 QUE32420
4003	CALL GETHN(ID . HNAME)	QUEC2430
	WRITE(6.943C)	QUE 32440
	CALL SETH(ID, ISET)	QUE02450
9430	FORMAT(10H? SCCRE =)	QUE02460
	SCCREH(ID) = GETNUM(=111.)	GUE 02470
	IF (SCOREH(ID).LT.U) ISET(ID)=0	QUE02480
	IF (SCOREH(ID).GE.O) ISET (ID)=1 GOTO 4000	QUE02490
	6010 4000	QUE02500
C C	PTION 44 TO REVISE DELAY/RESPONSE RULE. NOT CURRENTLY USED	QUE02510
	CALL GETHN(ID, HNAME)	QUE02520
	IF (RULEH(ID).FG. TEXTS(7)) GOTO 440	LUEC253C
	WRITE (6.9440)	QUE02540
9440	FORMAT (28H BOX DOES NOT USE DELAY/RESP)	QUE02550
	GOTO ACOU	QUE32560
440	CCNTINUE	QUE02570
	60T0 4000	QUE02580
		QUEC2590
	PTICN 45 TO REVISE SCORING RULES	QUE 02600
4605	CALL SETHN(ID, HNAME)	QUE02610
	WRITE (6,9503)	GUE02620
45 (READ(5.9504) R	QUE02630
	DO 451 I=1+6	QUE02640 GUE02650
	IF (R.EQ.TEXTS(I)) GOTC 452	GUEU2660
451		QUE02670
	WRITE(6,9505)	QUE02680
	6010 450	QUE02693
452	RULEH(1D)=R	QUE 02760
	SCOREH(ID)==1.	QUE02710
	CALL SETH(ID.ISET)	QUE: 2720
	60T0 4000	QUE02730
£	CPTICN 46 TO ENTER A NEW HIERARCHY INTO THE SYSTEM	GUED2740
	WRITE (6.9469)	QUE02750

.

	FL/G(5)=1	QUE02760
	READ(3*1+9464) NUMH	QUEC2770
	READ(3*2,9465) (LOCH(I), I=1, NUMH)	QUE 3278C
9464		QUE32793
9465		GUE02836
	READ(4 *1.9464) NUMH	QUE02813
	READ(4*2,9465)(LOCHS(I),I=1,NUMH)	QUE J2820
9460	The second with the second sec	QUE C283C
	HNUM=GETNUM(1. +FLOAT(NUMH) +1.)	QUE02840
	CO 465 I=1,40	GUE02850
	ISET(I)=0	QUE02860
	DC 466 11=1.10	GUE 02570
400	ICEx(1,11)=0	QUED28PC
	HNAME (I)=BLANK	QUE02890
465		QUEJ296J
		QUE02910
C	LOC=LOCH (HNUM)	QUE02520
C DEG.	IN BY READING INFO FOR FIRST BOX	QUE02930
	READ(3*LOC.9461) ANAME.NUM.R.Q	QUE02940
2461	FORMAT (146 +4X + 112 + 8X + 1 42 + 8X + 1 44)	QUE02950
	IF (ANAME .E G. NOMO) GCTC 468	60262960
	CALL ATGET (ANAME + ID + HNAME) IF (ID + GT + O) GOTO 464	QUE32970
	L=L+1	QUE 32980
	IC=L	QUE32990
	HNAME (ID) = ANAME	QUESSODS
464	IF (NUM.EQ.0) GCT0 463	GUE03310
404	QDEX(1D)=Q	QJE03020
	RULEH(ID)=R	GUEU3030
461		GUEC3040
40.1	IF (NUM .EQ. 0) GCTO 463	QUEC305.
	IF (NUM. GT. 5C) GOTO 4601	QUED3060
	DC 462 J=1+NUM	QUE03070
	J1=J+1	QUE03380
	LOC=LOC+1	QUE03090
	READ(3*LOC.9462) ANAME	QUE0310.
	CALL ATGET (ANAME , IE , HNAME)	QUED311C
	IF(IE.GT.0) 60T0 462	QUE03120
	L=L+1	QUE03130
	JE =L	QUE03140
	HNAME (L) = ANAME	QUE03155
462	IDEX(ID, J1)=IE	QUE03160
9462	FORMAT(1A6)	GUE 03170
463	LOC=LOC+1	GUE03180
	GCTO 460	QUE 33190
4601	NUM=NUM-50	QUEG32CC
	DC 4662 J=1.NUM	GUE03210
		QUE 03226
		QUE03230
		Q1 F03240
4602		QUEUS250
		QUE03260
		QJE03270
468	CONTINUE	QUE63280 QUE63290
	J=LOCHS(HNUM)	QUE03300

QUEASI	FORTRAN P ID=WCCWR 16.19.58 THURSDAY 6 DECEMBER 1979	PAGE 7
	READ(4*J.9463) ((BNAME(11).SCORE(11)).11=1.L)	QUE03310
	J=J+L/5+1	QUE03320
	READ(4 "J.9480) OSCORE	QUE03330
9463	FORMAT (5(1A6+1F6.3))	QUEC3340
	D0 469 K=1+L	QUED3350
	CALL ATGET (BNAME (K) + ID + HNAME)	GUEC3360
	IF (ID.EQ.U) 6CTO 469	QUE:3370
	SCOREH (ID) = SCORE (K)	QUE 03380
465	CONTINUE	GUEC3390
	00 459 1=1.40	QUED3400
459	IF (IDEX(I.1).EQ.0) RULEH(I)=BLANK	QUE03410
	GCTO A COO	QUE 33423
		QUE23430
C C	PTION 47 TO PRINT CURRENT BOX NAMES	QUE 0344C
4 6 6 7	CALL PNAME (HNAME)	QUE03450
	6CTO 4000	QUE 03460
		QUE03470
	PTICN 48 TO FILE HIERARCHY DATA	GUE03480
4008	LOC=LOCH(HNUM)	QUE 33490
	DC 48C I=1+L	QUEC3500
	IF (HNAME(I).EQ.BLANK) GOTO 480	QUE03510
	WRITE(3*LOC,9461) HNAME(I),ICEX(I,1),RULEH(I),QDEX(I)	QUE0352C
	14=1DEX(1,1)	QUE03530
	IF(I4.EQ.0) 60TO 480 LOC=LOC+1	QUEC3545
	11=MOD(IDEX(I+1)+50)+1	QUE03550 QUE03560
	CC 481 12=2+11	QUEC3576
	13=10E×(1,12)	GUEC3580
	IF (14.LE.50) GCTO 483	QUE03593
	WRITE (3*LOC.9272) GNAME (13)	QUEU3600
	GOTC 481	QUE03613
483	WRITE(3*LOC,9462) HNAME(13)	QUE03620
481		QUE 3363C
480	CENTINUE	QUE03640
	WRITE(3*LOC+9462) NOMO	QUE03650
	J=LOCHS(HNUM)	QUE03660
	#RITE(4*J.9463)(HNAME(13), SCOREH(13), 13=1.L)	QUEC3676
	J=J+L/5+1	GUE 33680
	WRITE(4*J.9480) ASCORE	QUE03690
9480	FORMAT(1CF8.5)	QUE03700
	GCT0 4000	QUE03710
		QUE03720
	PTICN 49 TO CHANGE NAME OF BOX	QUE 33730
4 6 0 9	CALL GETHN(ID+HNAME)	QJE03740
	WRITE(6,9271)	QUE03750
	RE 40 (5+9462) HNAME (ID)	QUEC3760
	GOTC 4000	GUE 03770
	ENC	QUE03783

A3-11

*

	FORTRAN P IC=WCCWR 16.20.08 THURSDAY 6 DECEMBER 1979 IONAL CSS, INC. (SUNNYVALE DATA CENTER) SUNY	PAGE 1
		QUECODI
	SUBROUTINE GETHN (ID . HNAME)	0000002
C SUEF	OUTINE GETHN INTERACTIVELY REQUESTS & BOX NAME AND RETURNS ITS	QUECOC3
	UMPER.	QUESC34
	DOUBLE PRECISION ALL HNAME (40) .A	QUEDCOS
	DATA ALL/6HALL /	QUEDOUG
	IO=1	QUECCO7
10	WRITE (6.9000)	QUECOOS
3008	FORMAT (20H? ENTER BOX WAME)	QUEDCOS
	READ(5,9001) A	QUECOID
9001	FORMAT(1A6)	QUED011
	IF (A.EQ.ALL) RETURN	QUEDC12
	CALL ATGET (A.ID.HNAME)	GUEGU13
	IF(ID.EQ.0) 6010 20	QUE0014
	RETURN	QUECC15
20	WRITE(6,9002)	QUE 3516
	FORMAT(11H? BAD NAME)	QUED017
	CALL PNAME (HNAME)	QUECC18
	GCTC 1C	QUECC19
	END	QUECO2C
		QUESC21
	SUBROUTINE ATGET (ANAME . ID . HNAME)	QUEG022
C SUER	OUTINE ATGET CHECKS & BOX NAME AGAINST THOSE CURRENTLY IN	QUEDD23
C THE	SYSTEM AND RETURNS ITS ID NUMBER (O IF NOT VALID).	QUEDC24
	DOUBLE PRECISION ANAME HNAME (40)	QUES025
	DC 10 I=1.40	QUED026
	IF (ANAME.EQ.HNAME(I)) GOTO 20	QUE0027
10	CONTINUE	QUED028
	IC=C	GUE0029
	RETURN	GUE SO3C
20	10=1	QUECC31
	RETURN	QUECC32
	ENC	QUEU033
		GUE0034
	SUBROUTINE SCREH(ID)	QUEC035
SUBR	OUTINE SCREH IS THE STER SUBROUTINE FOR SCORING BOXES	QUECC36
	COMMON L1(100) +QSCORE(100) +L2(100) +SCOREH(40) +RULEH(40)	QUELO37
	COMMON IDEX(40+10)	QUECC38
	CALL MULTEV(ID)	QUECC39
102	WRITE(6,9000) SCOREH(IC)	QUE004C
9000	FORMAT(13H THE SCORE IS+1F10.5)	QUEDD41
	RETURN	QUEC042
	ENC	QUEDCA3
		QUE3044
	SUERCUTINE BCX(ID)	GUE0045
SHER	DUTINE BOX SCORES BOXES WHICH HAVE QUESTIONNAIRES AS INPUT.	QUED0460
	COMMON L1(100),QSCORE(100),L2(100),SCOREH(40)	QUEDCA7
	COMMON RULEH(4C), IDEX(40,10)	QUE CC48
	DIMENSION INDEX(40), RESP(100)	4UE0049
	NAT=10Ex(10,1)-49	QUEDOSO
	DO 101 1=2 .NAT	QUEDC51
101	INCEX(I)=IDEX(ID,I)	QUE0052
	INCEX(1)=NAT-1	QUED653
	DC 102 I=1,100	QUED054
	RESP(I)=QSCORE(I)	6UE0055

A3-12

QUEASI	FORTRAN P ID=WCCWR 16.20.08 THURSDAY 6 DECEMBER 1979	PAGE 2
	RULE=RULEH(ID)	QUEDOSES
	CALL TXTURE (ID , RULE , INDEX , RESP , SCORE)	QUEODE70
	SCOREH(ID)=SCORE	OUECOESC
	RETURN	QUE00590
	ENC	CJ3703UA
		QUE00610
	SUBROUTINE MULTEV(ID)	QUECO620
C SUER	DUTINE MULTEV IS THE HIGHEST LEVEL OF THE	QUECJ630
C NESTI	ED BOX SCORING ROUTINE	QUEDO640
	COMMON LL(300) .SCOREH(40) .RULEH(40) . IDEX(40,10)	QUEDO65C QUEDO660
	DIMENSION INDEX(40)	QUESC673
	NUMAT-IDEX (ID,1)	QUEDS680
	IF (NUMAT.EQ.C) RETURN	GUESCERO
	IF (NUMAT.GT.SU) GOTO 40	QUECO750
	CO 10 I=1.NUMAT I1=I+1	GUELC710
	L=IDEX(ID,I1)	QUEDG720
	IF(SCOREH(L).6E.0) 60TO 10	GUEGC730
	CALL MULTEI(L)	QUEDO740
10	CONTINUE	QUEDC750
	J1=NUMAT+1	QUEG0760
	CO 30 J=1, J1	GUEGC776
30	INDEX(J)=IDEX(ID,J)	QUECO780
	CALL TXTURE (ID + RULEH (ID) + INDEX + SCOREH + SCORE)	QUEDO790
	SCCREH(ID)=SCORE	GUECOSOC
	RETURN	QUE06810 QUE06820
40	CALL BCX(ID)	QUESCA30
	RETURN	GUEG0840
	ENC	GUECO850
	SUEROUTINE GETEN(ID, QNAME, NUMG)	QUEDOB60
	OUTINE GETON INTERACTIVEL Y ACCEPTS & QUESTIONNAIRE NAME	QUEDCE70
	RETURNS ITS AN NUMBER	QUEDOBBC
C ANL	COMMON LOCG (10C)	CC8333UQ
	CIMENSION QNAME(100)	QUEDC9CU
	WRITE(6,9060)	GUE00910
9000	and a second state state that the second state is a second state of the second state stat	GUES0920
	READ(5+9001) A	QUED0930
9101	FCRMAT (1A4)	QUEGC940
	IC=NG(A)	QUE06950 QUE06960
	1F(ID.EQ.0) GOTO 10	QUE03970
	IF(ID.LT.O) RETURN	QUECC990
	I=LOCG(ID)	QUE00990
	READ(1*1.9002) NUMG	QUED1000
9002	FORMAT(10X,112)	GUE01C10
	RETURN CALL PQNAME(QNAME+NUMG)	QUED1020
10	GCTO 40	QUE01030
	RETURN	GUE01040
	ENC	QUE01050
		QUE D1 060
	FUNCTION YESNO(2)	QUE 01070
C FUN	CTION YESNO RETURNS THE ANSWER TO A YES-NO QUESTION	QUEDICSO
CCF	CR NC, 1 FOR YES.	QUE01093 QUE01160
	DATA X/1HN/	00201100

DATA X/1HN/

	UEASI	FORTRAN P ID=WCCWR 16+20+08 THURSDAY 6 DECEMBER 1979		PAGE 3
		A.T. V.I.V.		
		DATA Y/1HY/		QUE01110
		YESNO=0.		QUE 01120
		RE 40 (5,9000) A		QUED1130
	9000			QUE01140
		IF (X.EQ.A) RETURN		QUES1150
		IF (Y.NE.A) GOTO 10		QUE01160
		YESNC=1.		QUE01170
		RETURN		GUE01180
	10	WRITE(6,9001)		QUE01190
	9001	FERMATIZOH? TYPE YES CR NO)		QUE31200
		6010 20		QUEC1213
		ENC		QUEC1220
				QUE 01230
~	FULF	FUNCTION GETNUM (ALOW, AMIGH, TYPE)		QUE01240
C C	FUNC	TICN GETNUM RETURNS A NUMBER ENTERED INTERACTIVELY		QUE01250
C	AFILI	R CHECKING FOR THE APPROPRIATE RANGE AND TYPE.		QUE01260
10	10	REAL GETNUM		QUE01270
	10	READ (5, ., ERR=20) GETNUM		QUE01280
		IF (TYPE.EQ.O.AND.GETNUM.NE.AINT(GETNUM)) GOTO 50		QUE01290
		IF (GETNUM.GE.ALOW.AND.GETNUM.LE.AHIGH) RETURN		QUECIZOO
		IF (TYPE.EG.2) GOTO 40		QUE 31310
	30	WRITE(6,100) ALOW, AHIGH		GUE 1320
	100	The second secon)	
		60T0 10		QUE01340
		IF (TYPE.NE.2) GOTO 3C		GUEC1350
	40	CONTINUE		GUEC1360
		RETURN		GUEC1370
		WRITE(6,101) ALOW, AHIGH		GUE01385
	101	FERMAT(26H? ENTER AN INTEGER BETWEEN.G10.5,4HAND .G10.5,3H) GOTO 10	÷.,	
		ENC		QUE01400
		Enc		QUE01410
		CHEDONITING THINGS AND DUE THOSE ADDA ADDA		GUE0142C
~	SUDDO	SUEROUTINE TXTURE (NUM, RULE, INDEX, RESP, SCORE) NUTINE TXTURE COMPUTES A SCORE USING A SPECIFIED		GUED1430
		ING RULE.		QUEU1440
c	SCONT	COPHON LL(100), GSCCRE(100), LL1(580), GDEX(40)		QUEC1450
		DIMENSION INDEX(1), RESP(1), TEXTS(6)		QUE01460
		DATA TEXTS/*HA*, *SA*, *AV*, *SC*, *OR*, *G*/		QUE 01470
		M=INDEx(1)		GUE01480
		N=M+1		QUE01490 GUE01500
		CC 5 I=1.6		
		IF (RULE .EQ .TEXTS(1)) GOTO 6		QUE01510 QUE01520
		CONTINUE		
		WRITE (6,900)		QUED1530
	000	FORMAT(38H BAC RULE ENCOUNTERED COMPUTING SCORE)		QUESI543
		RETURN		QUEC1550
		GOTO (10.20.30.48.50.70) I		QUE01560 QUE01570
		SCORE=1.		QUEC1580
	-	CO 11 I=2.N		
		SCCRE=SCORE + RESP(INDEX(I))		QUE01590 QUE01600
		RETURN		GUE01610
		SCORE=1.		GUE01620
		DC 21 I=2+N		QUE01630
		SCORE=SCORE = (KESP(INDEX(I))+1.)		
		C=1./(2.**M-1.)		QUE01640 QUE01650
				00501000

QUEASI	FORTRAN P ID=WCCWR 16.20.08 THURSDAY 6 DECEMBER 1979	PAGE 4
	SCORE=C+(SCORE-1.)	QUE01660
	RETURN	QUED1670
		QUEC168C
30	SCORE=0.	QUE01693
	D0 31 1≈2•N	QUE01700
31	SCORE=SCORE+RESP(INDEx(I))	QUEC1710
	SCORE=SCORE+(1./M)	QUE01720
	RETURN	QUEC1730
40	SCCRE=1.	GUE 01740
	D0 41 1=2+h	QUE 01750
41	SCORE=SCORE + (15+RESP(INDEx(I)))	QUE31760
	SCCRE=(2.**M/(2.**M-1.))*(1SCORE)	QUE01770
E C	CONTINUE	QUE01761
	RETURN	GUE01790
50	SCCRE=1.	QUEDILGO
	DC 51 I=2+N	QUED1813
51	SCCRE=SCORE*(1RESP(INDEX(I)))	QUEC1825
	SCCRE=1SCORE	QUEC1830
	RETURN	GUE01840
70	Q=QDEX(NUM)	QUECIBLO
	IC=NQ(Q)	QUEC1860
	A=6SCORE(ID)	QUE01870
	B=C.2	GUEC1880
	DC 65 J=1+5	QUE01890
	IF(A.LT.B) GCTC 66	00201900
	B=B+Q.2	QUEC1910
	CONTINUE RULE=TEXTS(J)	QUE01920 QUE01930
96	I=J	GUE 61940
	GCTO 6	QUE01950
	END	QUE01960
		QUE01973
	SUBROUTINE MULTE1(ID)	QUE01980
C SUPR	OUTINE MULTER IS THE SECOND LEVEL OF THE BOX SCORING	GUE01990
	EM. IT IS A CLONE OF MULTER. AS ARE MULTE2. MULTE3. AND MULTE4.	
	CCMMON LL(300), SCOREH(40)RULEH(40), IDEX(40,10)	QUE02010
	CIMENSION INDEX(4C)	QUE02020
Sec. 3	~NUMAT=IDEX(ID,1)	GUE02030
	IF (NUMAY.EQ.D) RETURN	GUE02040
	IF (NUMAT.GT.50) GOTO 40	QUE02050
	DO 10 I=1+NUMAT	QUE02060
	11=1+1	QUE02070
	L=IDEX(ID,II)	GUEC2080
	IF(SCOREH(L).GE.0) GOTC 10	QUE 32090
	CALL MULTEZ(L)	QUEJ2100
10		QUE02116
	J1: NUMAT+1	00202120
	D0 30 J=1,J1	QUE02130
30	INJEX(J)=IDEX(ID,J)	QUEC2140
	CALL TXTURE(ID, RULEH(ID), INDEX, SCOREH, SCORE)	QUE02150
	SCCREH(ID)=SCORE	QUE02160
	RETURN	QUE02170 QUE02180
40	CALL BOX(ID)	QUE02190
	RETURN END	QUE02200
	ENU	00002200

QUE	AS1 FORTRAN P ID=	WCCWR 16.26.08	THURSDAY 6 DECEMBER 1979	PAGE 5
				QUE 02210
		2/103		QUE02220
	SUER OUTINE MULTE		THE BOX SCORING SYSTEM	QUE 32230
C SI		COREH(40)RULEH(40)		QUE02240
	DIMENSION INDEXC		10224404107	QUEC2250
	NUMAT=IDEX(10.1)	407		GUE 32260
	IF (NUMAT.EQ.0) RI	TUPA		QUEC2273
	IF (NUMAT.GT.50)			QUE02280
		6010 40		QUE02290
	DC'10 I=1.NUMAT 11=1+1			GUE02300
	L=10EX(10,11)			QUE J2310
	IF (SCOREH(L).GE.	D) 6070 10		QUE 2320
	CALL MULTE3(L)	or dere 10		QUE02336
1110	10 CONTINUE			QUEC2340
1.1	J1=NUMAT+1			QUE02350
	DO 36 J=1.J1			QUE02360
1.1	30 INDEX(J)=IDEX(ID)	• J)		GUE 02370
1.1		ULEH (ID) . INDEX . SCOP	EH.SCORE)	QUE02380
	SCCREH(ID)=SCORE			QUE 02390
	RETURN			GUE024CO
	AC CALL BOX(ID)			QUE02410
	RETURN			QUE02420
	ENC			QUEU2430
				QUE 32440
	SUEROUTINE MULTES	3(10)		QUEC2450
C SU	UBROUTINE MULTES IS	THE FOURTH LEVEL OF	THE BOX SCORING SYSTEM.	QUEG2460
	COMMON LL(300) .SO	COREH(40)RULEH(40)	IDEX(40+10)	QUE02470
	DIMENSION INDEX(453		QUE0248C
	NUPAT=IDEX(ID+1)			QUE 02490
	IF (NUMAT.EQ.0) RE	ETURN		GUE02500
	IF (NUMAT.GT.50)	GOTO 40		QUE02510
	DC 10 I=1, NUMAT			QUE02520
	I1=I+1			GUE02530
	L=IDEX(ID,11)	and here and the second second		QUE02540
	IF (SCOREH(L).GE.	O) GOTO : G		GUE 02550
	CALL MULTEA(L)			CUE02560
1	10 CONTINUE			QUE02570
	JI=NUMAT+1			QUED2580
	DO 30 J=1, J1			QUE02595
	30 INDEX(J)=IDEX(ID	The second se		QUE02660
		ULEH(ID) . INDEX . SCOP	(EH . SCORE)	QUE02610
	SCOREH (ID) = SCORE			QUE02625
	RETURN			QUE02630
1.1	CALL BOX(1D)			QUE02640
	RETURN			QUE 32650
	ENC			QUE02660
				QUE02670 QUE02680
	SUPROUTINE MULTE		THE BOX SCORING SYSTEM.	QUE02690
c su		COREH(40)RULEH(40)		QUE 52700
	DIMENSION INDEX(4			QUE02710
	NUPAT=IDEX (ID.1)			QUE02720
	IF (NUMAT.EQ.0) RE	THRA		QUE02730
	IF (NUMAT.GT.50)			QUE 02740
	DO 10 I=1.NUMAT			QUE02750
	00 10 1-19HORAT			

9	UEAS1	FORTRAN P ID=WCCWR 16.20.08 THURSDAY 6 DECEMBER 1979	PAGE 6
		11=1+1	QUE02760
		L=IDEx(ID,11)	QUE02776
		IF(SCOREH(L).GE.D) GOTO 10	QUE02780
		CALL MULTES(L)	QUE02790
	11	CONTINUE	QUE02800
		JI=NUMAT+1	QUE02810
		DC 30 J=1.J1	QUE02020
	30	INCEX(J)=IDEX(ID,J)	QUE02830
		CALL TXTURE (ID, RULEH (ID) . INDEX . SCOREH . SCORE)	QUES2840
		SCCREH(ID)=SCORS	QUE02850
		RETURN	QUEC2860
		CALL BOX(ID)	GUE 02870
		RE,URN	QUE02880
		END	QUE:2890
			QUED29CU
			QUE02913
		SUEROUTINE GETR(ILOC+RESP)	GUE 2920
		OUTINE GETR RETRIEVES THE RESPONSES FROM A SPECIFIED QUESTIONNAIR	
0	LOCA		QUEC2940
		INTEGER RESP(40)	QUED2950
		J=ILOC+1	QUE02960
		READ(2*J.9000) RESP	GUEC2973
	9000	FORMAT(40(1X+1A1))	QUE02980
		RETURN	QUEC2990
		ENC	00203000
			GUEC3610
		SUBROUTINE MULTES(IC)	QUE 33320
(SUBR	OUTINE MULTES SIMPLY PRINTS AN ERROR MESSAGE AND RETURNS.	QUE03030
		WRITE(6,900)	QUE 33040
		WRITE(6,901)	QUE 33050
		FCRMAT(54H YOU HAVE HIT THE LOWEST LEVEL IN THE SCORING ROUTINE	
	501	FERMAT (45H PLEASE TRY SCORING LOWER LEVEL BOXES FIRST)	QUE 030 70
		RETURN	QUEC3C80
		END	QUESSOPL

	2 FORTRAN P ID=WCCWR 16.20.19 THURSDAY 6 DECEMBER 1979 TIONAL CSS. INC. (SUNNYVALE DATA CENTER) SUNY	PAGE 1
		0115 0 0 0 0 0
	SUERCUTINE PRINTH(ID +HNAME)	QUECCC10
	ROUTINE PRINTH PRINTS & GRAPHICAL DESCRIPTION OF THE HIERARCHY	QUE 00020 QUE 00030
L CLLY	COMMON LXX(100), QSCORE(100), LXXX(100), SCOREH(40), RULEH(40)	GUEGODAD
	COMMON IDEX (40,10) . GDEX (40) . GNAME (100)	QUEC:050
	COUELE PRECISION HNAME (46)	QUESCOED
	DATA Q/1HQ/	QUECCO70
	WRITE(6,9000) HNAME(10)	GUECCOED
9000	FORMAT(1,25H HIERARCHY DATA FOR BOX +146+/)	QUEDCODG
	11=1DEx(10+12 **	QUECCIGO
	"1=MOD(I1,50)	GUESS110
	WRITE(6,9001) HNAME(ID), RULEH(ID), SCOREH(ID), GDEX(ID)	QUEDC120
	FORMAT (5H BOX: 1A6 .7H RULE: 1A2.8H SCORE: 1F6.3.	QUE00130 QUED0140
	15H G: 144) IF (11.EQ.1) RETURN	QUEDCISS
	DC 10 12=2.K1	GUEDC16C
	WRITE (6,9002)	GUE00170
9602	FORMAT(7H?)	QUEÓCIBO
	13=10Ex(10,12)	QUECC190
	IF(K1.EQ.I1) GCTO 15	QUECO2CO
	WRITE(6,9003) GNAME(13), GSCORE(13)	60206210
9003	FORMAT(17H QUESTIONNAIRE: +144+8H SCORE:+1F6.3)	GUEG0220
	6070 10	QUE00230
15		QUESC240
	14=1DEX(13+1)+1	QUECC250
	K2=MOD(I4+50)	QUEC0260
	IF(I4.EG.1) GOTO 10	QUE00270 QUE00280
	D0 40 15=2+X2 WRITE(6+9002)	QUE00290
	WRITE(6,9002)	GUED0300
	18=10Ex(13.15)	QUE00313
	IF (K2.EG.I4) 6(TO 25	QUEDC320
	WRITE(6,9003) QNAME(18), QSCORE(18)	QUECC330
	GCTO 4C	QUEG0340
25	WRITE(6,9001) HNAME(18),RULEH(18),SCOREH(18),GDEX(18)	QUEDJ350
	16=IDEx(18,1)+1	QUE00360
	K3=MOD(16,50)	QUE00370
	IF(I6.EQ.1) GOTO 45	QUEDU385
	DC 30 17=2 • K3	QUECC390
	DC 11 J1=1,3	GUEDC400
11	WRITE(6,9002) J9=IDEX(18,17)	QUE00410 QUE00420
	IF (K3.EQ.I6) GCTO 35	QUEOC43C
	WRITE(6,9003) GNAME(J9), GSCORE(J9)	QUEC0440
	GCTO 30	GUEGO450
35	WRITE (6.9001) HNAME (J9). RULEH (J9). SCOREH (J9). QDEX (J9)	QUE 30460
30	CONTINUE	QUESO470
45	CONTINUE	QUE00480
40	CONTINUE	QUE06490
10	CONTINUE	GUECOSCO
	RETURN	QUE00510
	END	QUECC523
		QUECC530
	SUBROUTINE SETH(ID, ISET)	QUE00540 QUE06550
C SUBK	OUTINE SETH IDENTIFIES WHAT BOX SCORES WILL BE CHANGED	00000000

QUEAS 2	FORTRAN P ID=WCCWR 16.20.19 THURSDAY 6 DECEMBER 1979	PAGE 2
C WHEN	A LOW LOVEL BOX'S SCURE IS CHANGED AND REINITIALIZES THEM. CCMMON LXX(300),SCOREH(40),RULEH(40),IDEX(40,10) DIMENSION ISET(40) IE=ID	QUE00560 QUE00570 QUE00560 QUE00560
	DO 26 13=1+10	QUECOSOC
	ICK=0	QUESS610
	CC 40 11=1.40	QUEJ0620
	12=IDEx(11,1)+1	GUE 0 C630
	IF (12.EQ.1.OR.12.GE.50) GOTO 40	GUE03640
	DC 10 14=2.12	QUECCESO
	15=IDEx(11,14)	QUEC3660
	IF(I5.XE.IE) G(TO 1C SCOREH(I1)=-1.	QUEJC679
	IE=11	GUEDGGBD
	6070 20	GUEC0690 GUEC070C
10	CONTINUE	QUE00710
	CCNTINUE	QUEDO720
4.	IF (ICK.EQ.0) GOTO 50	GUEDC730
20	CONTINUE	QUEC 0740
	RETURN	QUEOC750
	END	QUEGO76C
		GUEC0773
	FUNCTION NG(ANAME)	QUECO780
C FUNC	TICK NG RETURNS THE NUMBER OF THE PASSED QUESTIONNAIRE NAME.	GUEG0790
	COMMON LL(820) + GNAME(100)	QUEGOSCO
	DATA ALL/9HALL /	QUECOELO
	DC 10 1=1,100	QUECC820
	IF (ANAME.EG.GNAME(I)) GOTO 20	GUEJC830
10		QUECP64C
	IF (ANAME.NE.ALL) GOTO 30	3UE0085C
	NG = -1	QUESSA60
	RETURN	GUEDDR70
30	WRITE(6,9003)	QUEJJ380
9003	FORMAT(10H BAC NAME)	QUE 6 0 8 9 0
	NGED	QUECCOCC
20	RETURN NG=1	QUEGG410
20	RETURN	GUE00920
	END	QUECO93C
	ENU	GUE30940

	FORTRAN P ID=WCCWR 16.20.27 THURSDAY 6 DECEMBER 1979 IONAL CSS, INC. (SUNNYVALE DATA CENTER) SUNY	PAGE 1
		000300
		QUEGOO
	SUBROUTINE SCOREG(ID)	QUECOL
CUD DI	DUTINE SCORES QUESTIONNAIRES.	GUECOO
SUCH	COMMON LOCG(100), QSCCRE(100), LOCR(100)	GUEDDO
	DIMENSION IRESP(40), RESP(40), RULE(10), IWRST(40), WEIGHT(40)	QUESSO
	CIMENSION SCORE (40) + SCC(40) + INDEX(40)	QUEDDO
	INTEGER QDEX(1C.40)	GUEDDO
	DATA IBEST/IHA/	QUECOL
	BEST=FLOAT(IBEST)	QUEJOI
	CALL GETG(LOCG(ID), RULE, IWRST, WEIGHT, NUMO, NUMOP, QDEX)	QUECO1
	CALL GETR(LOCR(ID) .IRESP)	QUEDOI
	DO 10 I=1.NUMG	QUEGUI
	IF (IWRST(1).EG.IBEST) GOTO 10	QUESSI
	RESP(I)=(BEST-FLOAT(IRESP(I)))/(BEST-FLOAT(IWRST(I)))	QUEC 01
	RESP(1)=1WEIGHT(I)*RESP(I)	QUECOI
10	CCATINUE	QUEDOI
	DC 150 1=1.10	QUECCI
150	SCORE(1)=-1.	QUEUDI
	DC 140 11=1.10	GUESDS
	FLAG=0.	QUELSE
	CO 1CO I=1.NUMEP	QUEDOS
	J1=QDEX(1,1)+1	GUE 002
	DC 110 J2=2.J1	QUECCA
	J3=QDEX(I,J2)	QUECOZ
	1F (J3.GE.100) EOTO 115	QUESCA
	IF (SCORE (J3).LT.0) GCTC 120	QUEDOS
115	CONTINUE	GUEDDI
110		QUECOS
	DC 130 J4=2.J1	QUESS
	J5=QDEX(1, J4)	QUEDJ
	1F(J5.LT.100) COTO 160	CUEDO:
	J5=J5-100	QUECO
	SCC(J4)=RESP(J5)	GUE30
	6010 170	QUEDS
160	SCC(J4)=SCORE(J5)	QUEDS
170	INDEX(J4)=J4	QUEDO
130	CONTINUE	QUE 0 0
	FLAG=1.	QUESS
	INCEX(1)=QDEX(I,1)	QUEDD
	CALL TXTURE(I,RULE(I), INDEX, SCO, SCORE(I))	QUEDO
	IF(1.EQ.1) GOTO 145	GJE00
120	CONTINUE	QUEDO
100	CONTINUE	QUE03
	IF (FLAE.EQ.0) 60T0 145	QUEDD
	1F(SCORE(1).GE.0) €CTO 145	QUEDO
140	CONTINUE	GUECO
145	QSCORE(ID) = SCORE(1)	QUEGO
	WRITE(6.9000) OSCORE(ID)	QUEDO
000	FORMAT(13H THE SCORE =+1F6.3)	GUESC
	RETURN	QUECCS
	ENC	QUEDOS
	그는 것은 이번 것 같은 것 같아요. 동안 집 것 같은 것은 것 같이 가지 않는 것 같이 하는 것 같이 했다.	QUESOS
	SUEROUTINE GETGELOC, RULE, IWRST, WEIGHT, NUMQU, NUMGP, QDEX)	QUEDOS

QUEAS	3 FORTRAN P ID=WCCWR 16.20.27 THURSDAY 6 DECEMBER 1979	PAGE 2
	INTEGER ADEVIIA AND TEAVE AND THREETAND	
	INTEGER QDEX(10,40), ISAVE(40), IWRST(40) DIMENSION RULE(10), WEIGHT(40)	QUE00560
		QUE00570
		QUE0058C
0000	READ(1*J,9000) X.NUMQU,NUMGP	QUE00590
9000	FORMAT(1A4.6X.112.8×.112) J=J+1	QUEDOGOD
		00200610
9002	READ(1+J,9002) IWRST FORMAT(40(1x+1A1))	QUEC2620
	FORMAT(8F5.3)	QUEDDESO
2001	00 50 12=1.NUMGP	QUE00640
	J=J+1	GUE30650 QUE00660
	READ(1+J,9003) NAME, NUM, R	QUE00670
1010	FORMAT (2(112,8X),1A2)	QUEDGG80
	NAME=NAME - 49	QUEC3690
	QCEX(NAME, 1)=NUM	QUEOC7C3
	RULE(NAME)=R	QUEUC710
	J=J+1	QUEDO720
	REFD(1*J.90(4) (ISAVE(.2).J2=1.NUM)	GUEC0730
	DC 50 I=1.NUM	QUECO74C
	11=1+1	QUE 00750
	IF(ISAVE(I).LT.50) QDEX(NAME.II)=ISAVE(I)+100	QUECO760
	IF (ISAVE (I).GE.5.) QDEX(NAME.II)=ISAVE (I)-49	QUE00770
50	CONTINUE	QUECO780
	J=J+1	GUE JC790
	READ (1*J,9001)(WEIGHT(12),12=1.8)	QUEDDSCC
	IF (WEIGHT(1).GT.1) GCTC 70	QUEGOSIC
	IF (NUMGU.LE.8) RETURN	QUED0820
	J1=J+1	GUECC836
	READ(1 + J1+9001)(JEIGHT(12)+12=9+NUMQU)	QUECOSAS
	RETURN	QUEGO850
70	DO 100 J2=1.NUMQU	QUEDC860
100	WEIGHT(J2) = WEIGHT(2)	QUEDOB70
9004	FCRMAT(4012)	QUECEBBO
	RETURN	GUEDC890
	END	QUEDODOO
		GUE: 3913
		JUED0920
-	SUEROUTINE PRINTG(ID, GNAME)	QUECC930
C SUE	ROUTINE PRINTO PRINTS THE STRUCTURE FOR A QUESTIONNAIRE.	QUE00940
	COMMON LOCG(103),QSCORE(102),LOCR(103)	QUEUL950
	INTEGER QDEX(10,40)	QUEC0960
	DIMENSION RULE(10) , RESP(40) , QNAME(100) , IWRST(40) , WEIGHT(40)	QUEDD970
	DIMENSION IRESP(40) + X(40) + Y(40)	QUE00980
	DATA IBEST/1HA/	QUECO990
	CALL GETQ(LOCG(ID), RULE, IWRST, WEIGHT, NUMQU, NUMGP, QDEX)	QUEDIDCC
	EEST=FLOAT(IBEST) CALL GETR(LOCR(ID), IRESP)	GUE01010
	WRITE(6.8999)	QUE01620
	WRITE(6,9000) (NAME(ID)	GUE 61030
8999		QUE 31243 QUE 61050
9000		QUE01050
1000	WRITE(E, 9002) QSCORE(IC).RULE(1)	GUE01070
9002		QUEC1080
1002	DC 5 I=1.40	QUE01090
	IF (IWRST(I).NE.IBEST) GOTO 4	QUECIICO

QUEASS	FORTRAN P ID=WCCWR 16.20.27 THURSDAY 6 DECEMBER 1975	PAGE 3
	x(1)=0.	QUE 01110
	60T0 6	GJE01120
	X(I)=1-(BEST-FLOAT(IRESP(I)))/(BEST-FLOAT(IWRST(I)))	QUE01130
	Y(I)=1.~WEIGHT(I)+(1-X(I))	QUEG1143
5	CONTINUE	GUE01150
- C	I = 1	QUE 01160
	J1=50	QUEC1170
	WRITE (6.9004) J1.RULE (1)	QUE01180
9004	FORMAT(7H BOX: .112.8H RULE: .142)	QUEC1190
	11=QDEx(1,1)+1	QUE01200
	IF(11.EQ.1) GOTO 10	GUES1210
	00 20 12=2.11	QUE01220
	SRITE (6,9005)	QUE01230
9005	FORMAT(7H?)	GUE01240
	I3=4DEx(1,12)	QUE01250
	IF(13.6E.100) GOTO 30	QUED1260
	J1=I3+49	QUEC1273
	WRITE(6,9004) J1, RULE(13)	00201280
	14=QDEX(13+1)+1	QUE01290
	IF(I4.E0.1) GOTO 20	QUE 01300
	DC 40 15=2,14	QUE01310
	WRITE (6,9005)	QUE 01320
	WRITE (6,9005)	QUE01330
	18=QDEx(13,15)	GUEC1340
	IF(18.GE.100) GOTO 50	QUE 01350
	J1=18+49	QUE 3136C
	#RITE(6,9004) J1,RULE(18)	QUE 01370
	16=QDEx(18:1)+1	QUE01360
	1F(16.EQ.1) ECTO 45	QUE 01390
	00 60 17=2.16	QUEC14CO
	DO 11 J2=1,3	QUE01410
11	WRITE(6,9005)	QUEC1+20
	J9=QDEx(18,17)	QUE01430
	1F(J9. EE.100) EOTO 7C	QUE01440
	J1=J9+49	QUE 31450
	WRITE(6,9004) J1.RULE(J9)	QUED1460
	60T0 60	QUE01470
7 (J1=J9=100	QUE01480
9006	WRITE(6,9006) J1,X(J1),WEIGHT(J1),Y(J1)	QUEC1490 QUEC1500
	FORMAT(4H @=+112.7H RESP=+1F6.3.3H N=+1F6.3.3H S=+1F6.3)	QUEJ1510
60	CONTINUE GCTO 45	QUE01520
		QUE01530
50		GUE 0154C
45	WRITE(6,9006) J1,X(J1),WEIGHT(J1),Y(J1)	QUEC1550
*0	CONTINUE	QUE01560
40	60T0 2C	QUEC1570
30	J1=I3-100	QUECIERO
50	WRITE(6,9006) J1,X(J1),WEIGHT(J1),Y(J1)	QUE 01590
20	CONTINUE	QUE01660
	RETURN	QUE01610
10	END	QUE01620

QUEASA	FORTRAN P ID=WCCWR 16.20.36 THURSDAY 6 DECEMBER 1979 IONAL CSS. INC. (SUNNYVALE DATA CENTER) SUNY	PAGE 1
	SUBROUTINE HPR (ID, HNAME)	QUEDPOIC
r SUPR	OUTINE MPR PRINTS A PICTURE OF A PORTION OF THE HIERARCHY.	
C DOCH	COMMON LXX (300), SC CREH (40), TULEH (40), IDEX (40, 10)	QUECCO20 QUECCO30
	DIMENSION DOT(20,5), IPR(20,5)	QUEDCO4C
	DOUBLE PRECISION HNAME (40)	QUEDODAD
	DATA STAR/4H · · · / BLANK/4H /	QUECCOED
	CC 1 1=1+20	QUEDCO70
	DC 1 J=1.5	GUEDDOSD
	1PR(1.J)=0	QUESCOSS
1	DOT(I,J)=BLANK	QUECC100
	LEV=1	QUECCIIO
	11=10Ex(ID+1)+1	QUEDC120
	IPR(LEV,1)=ID	QUEGO133
	IF (11.EQ.1.CR.17.GT.50) GOTO 5	QUEDO145
6	DC 10 12=2.11	QUECC15C
	13=10Ex(10,12)	QUECO160
	IPR(LEV+2)=13	QUECC17C
	I4=IDEx(I3,1)+1	QUEDC180
	IF (I4.EQ.1.OR.I4.GT.50) GOTO 15	GUECC190
16	CC 20 15=2.14	QUE30200
	16=10Ex(13,15) 1PR(LEV,3)=16	QUE05210
	I7=I0Ex(I6+1)+1	QUEC0220 QUED0230
	IF(IT.EQ.1.0R.17.GT.50) 60T0 25	GUEU0240
26	DC 30 18=2.17	GUELC250
	19=IDEx(16.18)	QUECC26C
	IPR(LEV.4)=19	QUEDG270
	110=IDEx(19,1)+1	QUECC280
	IF(I10.EQ.1.OR.I10.GT.50) GOTO 501	QUEDC290
	DO 500 111=2.110	QUECO3CO
	IPR(LEV,5)=IDEx(19,111)	QUEC0310
500	LEV=LEV+1	GUEOL32L
	GOTO 3C	GJE00330
501	LEV=LEV+1	QUEC034C
30	CONTINUE	QUEGC35C
	GOTO 20	QUE00360
25	LEV=LEV+1	QUEOC370
20	GOTO 10	QUECO380
15		QUECO39C
10	LEV=LEV+1 CONTINUE	QUECCADC QUECCADC
	CONTINUE	QUEC0420
~	OC 60 I=1+19	GUEC0430
	D0 60 J=1+4	QUEC0440
	11=1PR(1.J)	QUED0450
	IF(11.EQ.0) \$010 60	GUEDD460
	12=IDEX(11,1)+1	QUEDC470
	IF (12.LE.2.0R.12.61.50) GOTO 60	QUECO480
	15=1+1	QUEOC490
	DO 61 13=15+19	QUEDOSCO
	J1=J+1	QUE00510
	CCT(13+J)=STAR	QUEC0520
	IF (IPR (13, J1).EQ.IDEX(11,12)) 60T0 62	GUE00530
61	CONTINUE	GUE 6 3540
62	CONTINUE	QUECC550

OUEASe	FORTRAN P ID=WCCWR 16.20.36 THURSDAY 6 DECEMBER 1979	PAGE 2
60		QUEDOS60
	WRITE(6,9000) HNAME(ID)	QUE00570
9000	FORMAT(32H HIERARCHY INFORMATION FOR BOX +14++//)	QUEDOSEO
	FORMAT(11H?********)	QUESUS9C
	FORMAT(3H? * 1146+2H *)	QUECCEDC
5036	FORMAT (4H?+S=+1F6+3+1H+)	QUEDC610
9004	FORMAT(6H?*RULE: +1A2+1H*)	QUE GE20
9005	FORMAT (3H?++)	QUESC630
9006	FCRMAT(1H?1A1)	QUESO640
9007	FCRMAT(3H?)	QUEDJ650
9608	FORMAT(2H?)	CUEDOEEO
9009	FORMAT(11H?)	QUESC670
9011	FORMAT (3H? +1A1+2H)	QUECOSAC
9612	FORMAT(',2H?)	QUECC690
	I=LEV-1	QUEDU7CO
	WRITE(6,9008)	GUECC71C
	D0 40 LEV=1+I	QUECO720
	LE=LEV+1	QUEDC730
	PAX=1	QUECC740
	DO 101 III=1.5	GUE00750
101	IF (IPR (LEV.III).NE.C.OR.DOT (LEV.III).EG.STAR) MAX=III	QUECO760
	DC 41 II=1+MAX	GUE 30770
	IF (IPR (LEV.II).NE.0) GOTO 39	QUE00780
	WRITE(6,9009)	QUEDG790
	GCTU 41	QUECOBGO
35	WRITE(6,9001)	QUECOB10
41		QUED0820
	WRITE (6,9012)	QUEDOS30
	CC 42 12=1.MAX	GUECO840
	13=1PR (LEV.12)	QUEDC850
	IF(13.EQ.0) 6010 43	QUECCASE
	WRITE(6,9002) HNAME(13)	QUEJOS7C
	GOTO 38	QUECOSBO
43	WR ITE (6.9009)	QUECC890
	WRITE (6.9011) DOT(LEV.12)	QUEJC90C
44	CONTINUE	QUE00910
42	CONTINUE	00200300
	WRITE(6,9012)	QUEDD930
	CC 66 11=1+4	QUED0940
	IF (DOT (2.11).EG. STAR) DOT (1.11)=STAR	GUE00950
66	IF (DOT (1.11).EG.STAR.AND.MAX.LT.I1) MAX=11	QUESSIG
60	DO 45 I2=1.MAX	QUE00970
	13=1PR (LEV +12)	QUEDU980
	IF(I3.EQ.D) GOTO 47	QUEDD990
	IF(12.GT.1) WRITE(6.9005)	GUE01000
	WRITE(6.9003) SCOREH(13)	QUEDICIC
		GUE01020
	IF (12.EQ.4) GOTO 45	QUE01030
		QUECIDAD
	IF (IPR (LEV.IA).GT.D) GCTO 46	QUE01050
	WRITE(6,9007)	QUE01060
	60T0 65	QUE01070
46	WRITE(6,9005)	
	WRITE(6,9006) STAR	QUE01080 QUE01090
	GOTO 45	
47	WRITE(6,9009)	QUEDIIDO

QUEASA	FORTRAN P ID=WCCWR 16.20	0.36 THURSDAY 6 DECEMBER 1979	PAGE 3
	WRITE(6.9007)		QUE01110
	IF(12.GT.1) WRITE(6.9007)		QUECIIZO
65	WRITE(6.9006) COT(LEV.12)		QUE01130
	CONTINUE		QUE01140
45	WRITE(6.9012)		QUE01150
	DC 49 12=1 . MAX		QUE01160
	13=1PR (LEV.12)		QUEC1170
	IF (13.EQ.0) 60TO 50		QUEC1180
	WRITE(6.9004) RULEH(13)		QUE01190
	6070 51		QUE01200
50	WRITE(6.9009)		QUE01210
	WRITE (6,9011) COT(LE,12)		GUE01220
	CONTINUE		QUE01233
	WRITE(6.9012)		QUED124C
	DC 52 11=1.MAX		QUE01250
	IF (IPR (LEV, 11) .NE. 0) GCTO 53		GUEC1260
	WP ITE (6.90 09)		QUEC127C
	SCTO 55		QUE01280
53	WRITE (6,9061)		QUE01290
55	WRITE(6.9011) COT(LE.11)		GUE01300
52	CCNTINUE		QUE 31310
	WRITE(6,9012)		QUEC1320
	CC 54 11=1.4		QUE01330
10 A.S.	WRITE(6,9009)		QUEG1345
54	WRITE(6,9011) COT(LE.11)		QUE01350
1.1	WRITE(6,9012)		GUEC1360
40	CCNTINUE		QUED1370
	RETURN		QUE01360
	END		QUE01390 QUE01400
	SUBROUTINE PNAME (HNAME)		QUE 31410
	OUTINE PNAME PRINTS THE NAME OF	THE CURRENT HITPARCHY PONES.	QUE01420
C SUCH	DCUBLE PRECISION HNAME (4C)	THE CORRENT HIERANCHT ECKES.	QUE01430
	WRITE (6.947C)		QUE01440
9475	FORMATISEN THE CURRENT HIERAR	CHY INCLUDES BOXES . ()	QUE01450
	DO 470 I=1.5		QUE01460
	11=8+(1-1)+1		QUEC147C
	12=11+7		QUE01480
470	WRITE (6.9471) (HNAME (13).13=11.	12)	QUE01490
9471	FCRMAT(8(1X+146+1X))	해야 한 것이 같은 것이 같은 것이 같은 것이 같이 없다.	QUE01500
	RETURN		QUE01510
	END		QUE01520
			QUE01530
	SUBROUTINE PONAME (QNAME . NUMQ)		QUE01540
C SUER	OUTINE PONAME PRINTS THE NAMES	OF THE CURRENT QUESTIONNAIRES.	QUE01550
	DIMENSION GNAME(100)		QUE01560
	WRITE(6,9000)		QUE01570
9000	FORMAT(33H THE CURRENT QUESTI	ONNAIRES ARE: ./)	QUE01590
10	WRITE (6,9001) (QNAME (12),12=1,N	UMG)	GUE01590
9001	FCRMAT (2X+10(144+2X))		QUE01600
	RETURN		QUE01610
	END		QUE01620

CHEAS	TIB FORTRAN P ID=WCCWR 12.48.56 THURSDAY 13 DECEMBER 1979	
NI	TIONAL CSS. INC. (SUNNYVALE DATA CENTER) SUNY	PAGE 1
r		
	ASI A PROGRAM FOR EVALUATION OF SAFEGUARDS QUESTIONNAIRES	QUEDOUIO
CANL	HIERARCHIES	QUED0020
	CCMMON LCCO(10C), USCCRE(10C), LOCR(10J)	QUECCC30
	COMMON SCOREH(40) + RULEH(40) + IDEX(40+10) + ODEX(40) + GNAME(100)	GUEDCC40
	DOUBLE PRECISION BRAME (40) BLANK NOMO HNAME (40) ANAME	GUECCISO
	INTEGER LOCHS(5) ISET(43) FLAG(10) LOCH(5) INESP(43) IBEST(40) REAL SCOKE(40) WEIGHT(40) RULE(10) TEXTS(7)	
	DATA TEXTS/2HHA+2HSA+2HAV+2HSO+2HOR+1H0+2HOR/	QUELSC70
	CATA IAA/1FA/.NUMO/EFNOMORE/ELANK/EH	QUEDGOES
	AAA=FLCAT(IAA)	QUECJU90 QUECCICJ
	DC 399 I=1.40	QUE02110
399		QUE10120
	J=2	QUELO130
C REA	C QUESTIONNAIRE LOCATIONS AND NAMES	GUEDC14C
	REAC(2, 5464' NUMQ	QUELCISC
	REAC(2,9465) (LCCR(12),12=1,NUMQ)	QUEDDIEC
	REAC(1.9464) NUK3	QUESC170
	REAC(1,9465) (LCCQ(11),11=1.NUMC)	QUEPLIEL
	CC 2C J=1+NUMJ	OUEC(190
30	READ(2+9727) GNAME(J)	QUESCOL
9727	FORMAT(144/)	QUEOC210
	REWIND 2	QUEDC22C
	REAC(2+9464) NUMG	QUECC230
	REAL(2,9465) (LCCK(12),12=1,NUMG)	QUECL245
	DC 10 I=1+100	QUEDC250
10	QSCORE(I)=-1.	QUECC26?
	DD 20 I=1+10	QUEDD273
	FLAG(1)=0	QUEJO260
	ECT INITIAL OPTION	0056:29:
	WPITE(c.910L)	GUESL300
9166	FERMAT(54H SELECT 1- FIERAPCHIES, 2- GUESTIONAIRES, 3- STOP	JOUED 310
	1CF=6ETHUM(13.,2.)	QUESC32C
1001	IF (IOF.EG.3) STOP	GUECC330
	GOTE (40(0,200) 10P	QUEJJ340
C REV.	IEW OPTION SELECTION AND BRANCH TO PROPER OPTION	QUEDC350
	IOPTEIOP	GUESC360
1100		QUECC37C
	IF (IOFT.GE.1. AND. IOFT.LE.3) GOTO 1101	QUECC380
	IF (IOFT.JE.21.AVD.ICPT.LE.26) GOTO 1102	QUEDC39C
	IF (IOFT.GE.41.AND.IOPT.LE.50) GOTC 1102	GUENSASS
1101	GOTO (1900+23(+1_0(+422) IOP IOP=IOPT	QUEJC410
1101		QUEDC420
1100	ICF=INT(FLCAT(IOPT)/10.)	QUE00430
1102		QUECCANO
	IF(IOP+10.EG.ICPT) GOTC 1001 ICPT=ICPT-10P+10	QUED0450
	GOTO (1000.201.100404) ICP	QUEJ0460
	GCTO 1000	QUECC470
C PRIM	T MENU AND GET QUESTIONNAIRE OPTION	QUECL48C
	IF (FLAG(2).E7.1) GUTO 210	QUECC490 QUE00500
230	WRITE(6,9200)	
200	WRITE(6.92:1)	QUE00510 QUE00520
	WRITE(6,9202)	QUE00530
	WP ITE (6.92 03)	GUEOC540
	WRITE(6.9204)	QUECASSS

GUEASI	B FORTRAN P	ID=WCCWR	12.48.56	THURSDAY 13 DECEMBER	1979	PAGE 2
	WRITE (6.9205					QUEDOSEC
9200	FORMAT(/.14H		: ./)			QUECOE71
9201	FORMATISTH	21- COMPUTE	SCORES	22- PRINT SCORES)	QUE JC580
9252	FORMAT (51H	23- SET SCCA	FC	24- REVISE .EIGHTS)	
9202	FCRMAT (51H FCRMAT (51H	25- BEVISE R	ULES	26- REVISE RESPONSES	i	QUEDC600
9207	FORMATISIH	27- REVISE N	AMES	26- PRINT NAMES)	
	FORMATISTH	29-	NO MORE RI	EVISIONS)	
	FERMAT (/.9H		no none n			QUEL 3630
	GOTO 220					QUEDDE40
210	WRITE (6.9206	.)				QUEDUESD
	FERMAT(18H)			QUED0660
220	10PT=6ETNUM	(21 29 2.)				QUEDL670
	IF (IOPT.LT.			1100		QUECOESC
	10PT=10PT-21					QUECOE90
201	FLAG(2)=1					GUECCTUP
C ERAM	CH TO PROPER					QUESC710
	GCTC (2001,2)	100.20.00.2000	.2102.2050	.2000,2006,1000) IOPT		GUECN72L
						QUEC0730
C (PTICN 21 TO I	COMFUTE A QUE	STIONNAIRE	SCORE		QUECC740
	CONTINUE					GUEDJ75.
214	DC 211 1=1.	NUMG				QUEOC7E0
	WRITE(6.93U					QUESC770
100 100 100 100 100 100 100 100 100 100	FORMAT (16h		E .144.1H:)		QUECO780
211	CALL SCORE	(1)				QUE00790
	GCT0 2000					ULEUDECO
	NUMBER OF A DESCRIPTION OF	an languar saran al lan				00500810
	FORMAT(33H	ENTER RULE (F	14 . SA . AV . SU	, OR))		QUE SCE20
	FCKMAT(1A2)	and the second second second	a basadi da	and the second		QUEDOR3C
	FORMATIZSH			,		QUECOS40 QUECO850
	FORMAT(15H	ENTER NAME	•)			30236860
	FCRFAT(1A4)			101 0071000		GUESCARC
				ION OPTIONS		GUESCHES
2008	CALL PONAME	CUNAME INUMUS				GUECCB9C
	GCT0 2000					QUEDCAGO
	IF (FLAG(5)	5 . A. C. T.	nr.			QUE00910
		EQ.1) GOTO 41				QUEOC920
402			• •			QUEDC930
462	WHITELE.940					GUED2940
	#RITE (6.940					GUEDC950
	WRITE (QUE00960
	WRITE (6.940					QUEDO970
	WRITE (E. 940					QUE0J980
	WEITE (6.940					QUE00990
	WRITE (6.920					GUEO1000
9400		H SELECT ON	E: ./)			QUE01010
9461	FCRMAT (51H	41- COMPUTE		42- PRINT DATA)	QUED1220
9402	FCRMAT (51H	43- ASSIGN		44 - REVISE DELAY/RES	P)	QUEGIL3?
9403	FCRMAT (51H	45- REVISE		46- SELECT NEW HIERA	RCHY)	QUE01040
9404	FCRMAT(51H	47- PRINT B		48 - FILE HIERARCHY D	ATA)	QUE01050
9405	FORMATISIH	49- CHANGE		50- PRINT HIERARCHY)	QUEC1º60
9407	FCRMAT (51H		51- NO	MORE REVISIONS)	QUE01070
	GCTC 4C3					QUE C1080
401	WRITE (6.940	6)				QUED1093
9406	FORMAT(18H	SELECT 41-5	1)			QUEC1100

OUEAS	IB FORTRAN P ID=WCCWR 12.48.56 THURSDAY 13 DECEMBER 197	9 PAGE 3
	ICPT=GETNUM(41.,47.,2.)	QUE01116
400	IF (IOPT.LT.41.CR.IOPT.CT.51) GOTO 1100	
	IOPT=10PT=40	QUESI120
	FLAG(4)=1	QUE01130
	ACH TO PROPER HIERARCHY OPTION	QUE01140
C DAM	IF (FLAG(5).EQ.C) GCTC 4006	QUE01150 QUE01160
	GOTO (4001,4602,4003,4004,4065,4006,4007,4008,4009,4002,1000)	
		QUESIIRO
· · · ·	CPTICN 41 TO SCOPE A HIERARCHY BOX	QUE01190
	CALL GETHN (ID HNAME)	QUE01200
	CALL SCREH(ID)	GUE01210
	CALL SETH(ID+ISET)	QUE01220
	ISET(ID)=0	QUE 01230
	GCT0 4:00	QUED124C
		QUE01250
()	CFTION 42 TO PRINT HIERARCHY DATA	GUE01265
	CALL GETHN(ID . HNAME)	QUE01270
	IF(IOFT.Eg.2) CALL PRINTH(ID, HNAME)	QUE01260
	IF (IUPT.EQ.10) CALL HPR(ID.HNAME)	QUE01290
	ECTC 4(D)	QUE01300
		QUE01310
	CPTICN 43 TO ASSIGN HIERARCHY EOX SCORES	QUE01320
4003	CALL GETHN(ID .HNAME)	QUE 31330
	WRITE(6,9430)	QUE01340
	CALL SETH(ID, ISET)	QUE 61350
9430	FORMAT(10H SCORE =)	QUEC1360
	SCCREH(ID) = GETNUM(-11.)	QUE(137)
	IF(SCOREH(ID).LT.D) ISET(ID)=0	GUEU1380
	IF (SCOREH(ID).GE.C) ISET(ID)=1	QUE01390
	GCTO 4.00	QUED14CC
4.11.1		QUE01410
4004	CALL GETHN(ID, FNAME)	
4004	IF (RULEH (ID).EG.TEXTS(7)) ACTO 442	QUE 01430
	WRITE(6.9440)	QUE01440
9441	FORMAT (28H BOX DOES NOT USE DELAY/RESP)	QUE01450
	GUTO 4:00	QUE01463 QUE01470
44:	CONTINUE	QUE31480
	6CT0 4C00	QUE01490
	이 방법에 해외했다. 이 가지 않는 것은 것은 것은 것은 것은 것은 것을 하는 것을 수 있다.	QUED15CO
C C	PTICN 45 TO REVISE SCORING RULES	GUEC1510
4105	CALL GETHN(ID . HNAME)	GUEG1520
	WRITE(6,9503)	QUE01530
	RE (5 + 9504) R	QUEC1540
	DC 451 I=1+6	QUE01550
	IF (R.EG.TEXTS(I)) GOTC 452	QUE 01560
451	CONTINUE	QUED1570
	WRITE(6,9505)	QUED1580
	6010 450	QUE01590
452	RULEH(ID)=R	QUE01600
	SCCREH(ID)==1.	QUEL1619
	CALL SETH(ID,ISET)	QUE 01620
	GOTO 4000	QUE01630
		QUEC1640
((PTICN 46 TO ENTER A NEW HIERARCHY INTO THE SYSTEM	QUE01650

QUEASI	B FORTRAN P 1D= WCCWR 12.48.56 THURSDAY 13 DECEMBER 1975	PAGE 4
	WRITE(6,9460)	QUESIEES
4000		QUE01670
	FLAG(5)=1 READ(3+9464) NUMH	QUECIESU
		QUE01690
0.477	READ(3,9465) (LOCH(I),I=1,NUMH)	QUED1700
	FCRMAT(112)	QUE01710
9460	FCRMAT(2014)	QUE01720
	READ(4+9464) NLMH	QUE01730
	READ(4.9465)(LCCHS(I),I=1.NUMH)	QUE31740
9462	FORMAT(28H ENTER HIEPARCHY NUMBER)	QUE01750
	HNUM=GETNUK(1. +FLCAT(NUMH)+1.)	QUE01760
	CC 465 I=1.40	GUE0177C
	ISET(I)=0	QUEC17EJ
	DC 466 11=1.1.	QUE 01790
466		QUEC1500
	HIAME(I)=BLANK	QUE01815
465	SCCREH(I)==1.	
	La	GUE01820 GUE01830
	LCC=LCCH(HNUM)	QUE01840
	N BY READING INFO FOR FIRST BUX	
465		QUE31850
9461	FCRMAT(1A6+4X+112+8X+1A2+8X+1A4)	QUESIEES
	IF (ANAME.EG.NUMC) GOTO 468	QUEC1870 QUE01880
	CALL ATGET (ANAME +10+HNAME)	QUE01890
	1F(10.6T.0) 60TO 464	20201903
	L=L+1	
		QUE01913 QUE01920
	HLAME(10)=ANAYE	GUE 01930
40.4	IF (NUM.EQ.U) GCTU 463	QUEC1940
	QDEX(10)=Q	
	RULEH(ID)=R	QUE01950
461	ICEX(IC+1)=1.0*	QUE01960
	IF (NUK.EQ.C) SCTO 463	QUE01970
	IF(NUK.GT.50) GOTO 4EC1	GUED19RD
	DC 462 J=1+NJM	QUE01990
	01=0+1	GUE 62000
	LOC=LOC+1	QUE02010
	READ(3.9462) ANAME	QUE02:20
	CALL ATGET (ANAME + IE + HI, AME)	QUE02330
	IF(IE.GT.0) GOTO 462	QUE32043
	L=L+1	QUEJ2CED
	IE=L	QUE02063
	HNAME(L)=ANAME	GUEC2070
	ILEX(ID, JI)=IE	QUED2CEC
9462	FCRMAT(1A6)	QUE 52293
462	LCC=LOC+1	QUE32100
	GOTO 460	QUEC2110
4661		QUE02120 QUE02130
	DC 4602 J=1.NUM	
	J1=J+1	QUE02140
	LCC=LOC+1	QUE02150 QUE02160
	RE #D(3+9272) ANAM	QUEC2170
4602		
	LOC=LOC+1	QUE02183
1.1.1.1.1.	GCTO 4EU	QUE 32190
468	CONTINUE	QUE02200

JELOCHS(HNUM) QUE 02210 READ(4,9463)((ENAME(I])*SCORE(I])*+I1=1+L) QUE 02220 JSUG(A) QUE 02230 READ(4,9463)(SCORE QUE 02230 CONSTRUCTION QUE 02230 DC 469 KE1+L QUE 02230 CO 469 KE1+L QUE 02230 CALL ATGET (RAFE(K)*, ID+HNAME) QUE 02230 F (ID-LG, C) 6CTO 465 GUE 02230 GC 455 IE1+C QUE 02323 GUE 02300 QUE 023243 GUE 02301 QUE 023243 GUE 02302 QUE 023243 GUE 02303 QUE 023243 GUE 02304 QUE 023243 GUE 02305 QUE 023243 GUE 02305 QUE 023243 GUE 02305 QUE 02325 GUE 1 CALL PAAPE(HNUME) QUE 02245 GUE 02305 QUE 02256 GUE 1 CALL PAAPE(HNUME) QUE 02256 GUE 1 CALL PAAPE(HNUME) QUE 02257 KE 1 ND 4 QUE 02257 <th>GUEAS</th> <th>IE FCRTRAN</th> <th>P IC=#CC#R</th> <th>12.48.56</th> <th>THURSDAY</th> <th>13 DECEMPER</th> <th>1975</th> <th>PAGE 5</th>	GUEAS	IE FCRTRAN	P IC=#CC#R	12.48.56	THURSDAY	13 DECEMPER	1975	PAGE 5
RED(4,9463)((0), MARE(1)), SCORE(1)), 11=1, L) DUE(223) UUE(223) RED(4,9460) USCORE GUE(223) 9462 FORMAT(511A6,1F6.3)) GUE(223) C3LL ANGET(6), AFE(X), 10, HNAME) GUE2270 C3LL ANGET(6), AFE(X), 10, HNAME) GUE22250 C3LL ANGET(6), AFE(X), 10, HNAME) GUE22250 C459 GCORE GUE22250 SCORE(L), 153CCRL(X) GUE22250 C455 FICDEX(1,1,1,53, C) GUE22250 C456 GCIL (A,100 GUE2230 C410 GUE22350 GUE22350 C4110 GUE22350 GUE22350 C4111, AT TO FAINT CURRENT BOX NAMES GUE22350 GUE2250 GUE2250 GUE2250 REAC13,9660 NUPH GUE		JELOCHSCHN	UM)					0115 02210
JBUGL /51 QUE 2233 9453 FORMATISTIALA, IFG. 3)3 QUE 02243 9453 FORMATISTIALA, IFG. 3)3 QUE 02256 CALL ATGT (BNAPE(K), ID, HNAME) QUE 02266 CALL ATGT (BNAPE(K), ID, HNAME) QUE 02266 SCCREH (LD)=SCCRL(K) QUE 02266 SCCREH (LD)=SCCRL(K) QUE 02266 GO 459 FI (DE K(L)) QUE 02267 QUE 02266 GO 459 FI (DE K(L)) QUE 02267 QUE 02267 GO 459 FI (DE K(L)) GO 459 GUE 02276 GO 459 FI (DE K(L)) GO 450 GUE 02276 GO 450 GUE 02372 GO 450 GUE 02372 GO 450 GUE 02372 GO 50 FI (DE K(L)) GO 120277 GUE 02372 GO 6070 GUE 02372 GO 70 GUE 02372 GUE 02472 GUE 02473 REAC(3, 9464) MUPH GUE 02470 GUE 02473 REAC(3, 9464) MUPH GUE 02460 <td></td> <td></td> <td></td> <td></td> <td>11=1-11</td> <td></td> <td></td> <td></td>					11=1-11			
REACT+996D3 USCORE QUE 02250 9461 FORMAT (551464)F6.33) QUE 02250 CALL ATAGT (BAPE(K), LD, HNAME) QUE 02250 1F (LD, GG. D) GCTO 469 QUE 02250 465 CCNTIAUE QUE 02230 b0 455 IF.40 QUE 02230 d1 455 IF.40 QUE 02330 d2 455 IF.40 QUE 02330 d2 456 CCNTIAUE QUE 02330 d2 457 CTL 4.47 TO FRIAT CURRENT BOX NAMES QUE 02330 d2 CPTILN 47 TO FRIAT CURRENT BOX NAMES QUE 02350 d2 CPTILN 47 TO FRIAT CURRENT BOX NAMES QUE 02350 d2 CPTILN 46 TO FILE HIERARCHY DATA QUE 02350 d2 CPTILN 46 TO FILE HIERARCHY DATA QUE 02350 d2 CPTILN 46 TO FILE HIERARCHY DATA QUE 02350 d2 CPTILN 46 TO FILE HIERARCHY DATA QUE 02350 d2 CPTILN 46 TO FILE HIERARCHY DATA QUE 02350 d2 CPTILN 46 TO FILE HIERARCHY DATA QUE 02350 d2 CPTILN 46 TO FILE HIERARCHY DATA QUE 02350 d2 CPTILN 46 TO FILE HIERARCHY DATA QUE 02450 REACCS, 9663 NUMH QUE 02450 REACCS, 9663 NUMH QUE 02450 REACCS, 96663 NUMH <td></td> <td></td> <td>of the stand the stand</td> <td></td> <td></td> <td></td> <td></td> <td></td>			of the stand the stand					
9462 FORMATISSIAG.IS.) QUE02250 CALL ATAGIT (BNAPE(K), ID, HNAME) QUE02270 IF (ID, GG, C) 6CTC 469 QUE02270 SCCREH(ID)=SECRE(K) QUE02230 GCT QUE02250 SCCREH(ID)=SECRE(K) QUE02230 GCT QUE02300 GCT ASS F(IDEXII)=SECRE(K) QUE02300 GCT ASS GCT ASSS GCT ASSS GCT ASSSS GCT ASSSSS GCT ASSSSSSS GCT ASSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS			C) USCORE					
D0 469 KE1+L QUE02270 IF (10.60.0) 6CT0 469 QUE02270 IF (10.60.0) 6CT0 469 QUE02270 459 CCMTINUE QUE02230 450 CCMTINUE QUE02300 451 IF (10EX(1+1)+EQ.C) RULEH(I)=BLANK QUE02300 455 IF (10EX(1+1)+EQ.C) RULEH(I)=BLANK QUE02300 456 CCMTINUE QUE02300 457 IF (10EX(1+1)+EQ.C) RULEH(I)=BLANK QUE02300 458 CCTC 4400 QUE02300 C CPTICN 47 TO FRINT CURRENT BOX NAMES QUE02340 4107 CCLL PNAME(CHRAME) QUE02350 GT - 400 QUE02350 GT - 400 QUE02350 GT - 400 QUE02350 GE - CPTICN 46 TO FILE HIERARCHY DATA QUE02350 GE - CPTICN 46 TO FILE HIERARCHY DATA QUE02350 REACTS 9463 NUMH QUE02420 REACTS 9464 NUMH QUE02420 REACTS 9464 NUMH QUE02420 REACTS 9464 NUMH QUE02420 REACTS 9464 NUMH QUE02460 GT + THE (1)-GU-PLAKK) EOTO 460 QUE02450 GT + THAME (1)-GU-PLAKK) EOTO 460 QUE02450	9463							
CALL ATABET (BAPE(RX), ID, HNAME) QUE 2230 IF (ID, GG, O) ECTO 469 QUE 2230 SCCREM(ID) ESCCRE(K) QUE 2230 GL 455 IF: (ID, FALLER) QUE 2230 GL 455 IF: (ID, FALLER) QUE 2230 GL 4105 QUE 22350 GL 1 CALL PNAME (HHILRS) QUE 22350 GL 1 CALLE PNAME (HHILRS) QUE 22350 GL 1 CALLER NAME (D) QUE 22350 GL 2 CALLER NAME (D) QUE 22400 REACT 3.94650 NUPH REACT 3.94650 NUPH REACT 3.94650 NUPH								
IF (10.EG.C) ECTO 469 GUEC2260 SCREHGILDSCRELK) GUEC2300 469 CONTINUE GUEC2300 D0 459 I=1,+0 GUEC2300 459 IF(IDEXCL:).D.EG.C) RULEH(I)=BLANK GUEC2300 GCTL 4100 GUEC2300 GCTL 4101 GUEC2300 GCTL 4101 GUEC2300 GCTL 4102 GUEC2300 GCTL 4101 GUEC2400 REACC+94640 NUMH GUEC2400 REACC+94640 NUMH GUEC2400 REACC+94641 NUMH GUEC2400 GCTL 4401 GUEC2400 GUEC2400<				INAME)				
SCCREHILD:=SCCRE(K) QUEC2300 469 CONTINUE QUE02319 459 IF (IDEX(I,I).EQ,C) RULEH(I)=BLANK QUE02319 6CIL 410C QUE02330 9CIL 410C QUE02331 9CIL 410C QUE02333 9CIL 410C QUE02353 9CIO CALL PNAME(HNLKE) QUE02373 9CIO CALL PNAME(HNLKE) QUE02420 REACCL-POCH(HNLM) QUE02420 REACCL-POCH(HNLM) QUE02420 REACCL-POCH(HNLM) QUE02420 REACCL-POCH NUMH QUE02420 REACCL-POCH NUMH QUE02420 REACCL-POCH NUMH QUE02420 REACCL-POCH NUMH NELLONCKI-LID.								
445 CONTINUE QUE02300 D0 455 IF (10Ex(1+1).E3, C) RULEH(I)=BLANK QUE02310 655 IF (10Ex(1+1).E3, C) RULEH(I)=BLANK QUE02323 GC10 4100 QUE02343 C CPTILN 47 TO FRINT CURRENT BOX NAMES QUE02343 QUE02350 GCT0 4(30 QUE02350 C CPTICN 4c TO FILE HIERARCHY DATA QUE02350 4(36) CCLCLOCH(NAM) QUE02350 GCT0 4(30 QUE02350 QUE02350 C CPTICN 4c TO FILE HIERARCHY DATA QUE02350 A(36) CCLCLOCH(NAM) QUE02400 REAL(30, 9464) NUPH QUE02420 REAL(30, 9464) NUPH QUE02420 REAC(4, 9		SCCREH(1L)	=SCCRE(K)					GUE G2290
#55 IF(IDEx(I,1),EG,C) RULEH(I)=BLANK GUE2322 GUE2323 GC1C 4106 GUE22353 C CFTICN 47 TO FRINT CURRENT BOX NAMES GUE22353 GCTD 4107 GUE10400 GUE22353 GUE22353 GCTD 4108 GUE22353 GCTD 4108 GUE22353 GUE22353 GUE22353 GUE22354 GUE22353 GUE22355 GUE22353 GUE22353 GUE22353 GUE22354 GUE22353 GUE22355 GUE22353 GUE22353 GUE22353 GUE22353 GUE22353 GUE2260 GUE22533 GUE2260 GUE2260 REATC3,9463 NUMH GUE22423 REAC(4,9464) NUMH GUE22423 GUE2442 GUE22423 GUE22453 GUE22453 GUE141,13 GUE22453 GUE22453 GUE22453 GUE22453 GUE22453 GUE22453 GUE22453 GUE22453 GUE22453 GUE22453 GUE22453 GUE22553 IF(14,12,1),E0)+1	469	CONTINUE						QUE 22300
GC1C 4:06 GUEC2333 GUED2343 C CPTICN 4:7 TO PRINT CURRENT BOX NAMES GUED22350 GUED2350 GUED22360 GCTO 4:00 GUED22350 C CPTICN 4: TO FILE HIERARCHY DATA SUED2350 GUED2350 4:06 LOC=LOCH(HNUM) GUED24251 REWIND 3 GUED24252 REWIND 3 GUED24252 REAC13,9463 NUMH GUED24253 REAC13,9465 (LOCH(1),1=1,NUMH) GUED24253 REAC13,9465 (LOCH(1),1=1,NUMH) GUED2453 REAC14,9465 (LOCH(1),1=1,NUMH) GUED2453 LOC=LOC+1 GUED2453 IF (HNAME(1),10EX(1,1),SUEL(1,1),90EX(1) GUED25450 IF (14,4E2,0) GOTO 460 GUED2550 LOC=LOC+1 GUED2550 IF (14,4E3,9462) NAME(13) GUED2550 IF (14,4E3,9462) NAME(13) GUED2550 IF (14,4E3,9462) NAME(13),SCOREH(13),13=1,L) GUED22653 GUED2550 GUED2663 GUED26653 VRITE(4,9463) (NNAME(13),SCOREH(13),13=1,		Dù 459 I=1	+40					GUE02310
C CPTICN 47 TO PRINT CURRENT BOX NAMES GUE22350 4C07 C1LL PNAME(HNAME) C770 4 GUE C770 4 GUE	455	IF (IDEX(1.	1).EQ.C) RULEN	(I)=BLANK				QUE02320
C CPTICN 47 TO FAINT CURRENT BOX NAMES GUE02353 4107 CALL PRAME(HNAME) GUE02350 67 CPTICN 42 TO FILE HIERARCHY DATA GUE02350 4506 LOCELOCHHNUM) GUE02350 RE.IND 4 GUE02350 RELIND 3 GUE02240 RELIND 4 GUE02420 RELOCS,9%64) NUPH GUE02440 REACCS,9%65) (LOCH(I),I=I,NUMH) GUE02440 REACCS,9%65) (LOCH(I),I=I,NUMH) GUE02440 GUE02440 GUE02440 REACCS,9%65) (LOCH(I),I=I,NUMH) GUE02440 GUE02440 GUE02440 REACCS,9%65) (LOCH(I),I=I,NUMH) GUE02440 GUE02440 GUE02440 IF (H,YAPC(I).EC.ELALAK) GOTO 4E0 GUE02440 GUE02440 GUE02440 <t< td=""><td></td><td>GC1C 4100</td><td></td><td></td><td></td><td></td><td></td><td>QUEC2330</td></t<>		GC1C 4100						QUEC2330
4107 CALL PAAMÉ(HNIAKE) QUÉD2360 0070 QUÉD2360 QUÉD2370 1 CELOCALOCHANLMO) QUÉD2370 4106 DCELOCALANNON QUÉD2370 1 RELINO 3 QUÉD2400 RELINO 4 QUÉD2410 QUÉD2410 REACCA,9463) NUMH QUÉD2450 QUÉD2450 REACCA,9463) NUMH QUÉD2450 QUÉD2450 REACCA,9465) (LOCHID,1]=1,NUMH) QUÉD2450 QUÉD2450 LOCALOCHANNA GOTO 460 QUÉD2450 LOCALCA,1) NUMME(1),IDEX(1,1),QUÉX(1) QUÉD2450 LOCALCC+1 QUÉD2550 QUÉD2550 LCCASINANC QUÉD2550 <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>QUE02340</td>								QUE02340
0(TD 4(00) QUE02370 f == CPTICN 4c TO FILE HIERARCHY DATA == QUE02370 4(u6 LOC=LOCH(HNUM) QUE02400 RE.IND 3 QUE02410 RE.IND 4 QUE02420 READ(3,9464) NUPH QUE02420 READ(2,9465) (LOCH(I).I=1.NUMH) QUE02420 READ(2,9465) (LOCH(I).I=1.NUMH) QUE02420 READ(2,9465) (LOCH(I).I=1.NUMH) QUE02450 READ(2,9465) (LOCH(I).I=1.NUMH) QUE02450 GUE02450 QUE02450 IF(HVAME(I).EG.BLANK) EOTO 4E0 QUE02450 GUE02560 QUE02450 IF(14.9465) (LOCH(I).I).IDEX(I.I).RULEH(I).QDEX(I) QUE02450 GUE02550 QUE02550 IF(14.26.0) GOTO 480 QUE02550 ICC4L0C+1 QUE02550 ISTEDX(I,I) QUE02550 ISTEDX(I				BOX NAMES				QUE 02350
<pre>f CPTICN 46 TO FILE HIERARCHY DATA GUE02360 f uc CDC=LOCH(HNUM) RELIND 3 RELIND 3 RELIND 4 RELIND 4 RELC(3,9464) NUMH READ(3,9464) NUMH REAC(4,9464) NUMH REAC(4,94</pre>	4607		(HNLME)					QUE02360
f -= CPTICN 40 TO FILE HIERARCHY DATA QUE02390 4EU6 LOCELOCH(HNUM) QUE02410 RE.IND 3 QUE02410 RE.IND 4 QUE02420 RELIND 3 QUE02420 RELIND 4 QUE02420 RELIND 3 QUE02420 RELIND 4 QUE02430 REACC3.9464) NUMH QUE02440 REACC4.9464) NUMH QUE02440 REACC4.9464) NUMH QUE02440 REACC4.9464) NUMH QUE02450 QUE02450 QUE02450 GUE0440 QUE02450 QUE02450 QUE02450 GUE0460 QUE02450 QUE02450 QUE02450 GUE0440 QUE02450 QUE02450 QUE02450 QUE02450 QUE02450 QUE02450 QUE02450 QUE02450 QUE02450 QUE02450 QUE02450 QUE02450 QUE02550 IF (14+E0-0) GOTO 460 QUE02550 IF (14+E0-0) GOTO 463 QUE02550 IF (14+E0-0) GOTO 463 QUE02550 IF (14+E0-0) GOTO 463 QUE02550 IF (14+E0-0		CCTO 4100						
4(36 L0C=L0CH(HNUM) QUE22400 REWIND 3 QUE22401 REWIND 3 QUE22402 READC3.9964) NUMH QUE22402 READC3.9965) (LOCH(1),I=1,NUMH) QUE22402 REAC(4,965) (LOCH3(I),I=1,NUMH) QUE22402 REAC(4,9665) (LOCH3(I),I=1,NUMH) QUE22402 REAC(4,9665) (LOCH3(I),I=1,NUMH) QUE22402 REAC(4,9665) (LOCH3(I),I=1,NUMH) QUE22402 REAC(4,9665) (LOCH3(I),I=1,NUMH) QUE22402 If HNAPE(I),EU,BLAKN) EOTO 480 QUE22403 If HNAPE(I),EU,BLAKN) EOTO 480 QUE22403 If (14,20,0) GOTO 480 QUE2250 If (14,120,0) GOTO 480 QUE2								
REwIND 3 QUE 2410 RE.IND 4 QUE 2420 REELC3,9464) NUPH QUE 2420 REACC3,9464) NUPH QUE 2420 REACC4,9464) NUPH QUE 2420 REACC4,9465) (LOCH SCI),1E1,NUPH) QUE 2420 DC 480 I=1xL QUE 2420 IF 4NAME(I),EG.PLAAK) COTO 480 QUE 2421 IF 4NAME(I),IDEX(I,1),RULEH(I),GDEX(I) QUE 2421 IF 4NAME(I),IDEX(I,1),RULEH(I),GDEX(I) QUE 2420 QUE 2250 QUE 2250 IF 4NAME(I,1) QUE 2250 IC 421 I22,II QUE 2250 IF (IA,LE,50) ECTO 483 QUE 2250 IF (IA,LE,50) NOND QUE 2250 WRITE(IA,9462) KNAME(I3),SCOREH(I3),I3=1,L) QUE 2260 WRITE(IA,9460) GSCORE				IY DATA				
RE.IND 4 GUE02420 REACC3.99461 NUPH GUE02450 GUE02450 REACC4.99461 NUPH GUE02450 GUE02450 REACC4.99461 NUPH GUE02450 GUE02450 REACC4.94665 (LOCHI).IEI.NUMH) GUE02460 GUE02470 GUE02470 IF (HNAME(I).EG.BLANK) EOTO 4E0 GUE02470 GUE02470 GUE02470 IF (IA.60.0) GOTO 4E0 GUE02500 IF (IA.60.0) GOTO 480 GUE02500 IF (IA.60.0) GOTO 480 GUE02500 IF (IA.60.0) GOTO 480 GUE02500 ISIDEX(I,I). GUE02500 ISIDEX(I,I). GUE02500 ISIDEX(I,I2) GUE02550 ISIDEX(I,I2) GUE02550 ISIDEX(I,I2) GUE02550 WAITE(3,9462) HNAME(I3),SCOREH(I3),I3=1,L) GUE022600 JSICCHSCHNUN) </td <td>4100</td> <td></td> <td>NUM)</td> <td></td> <td></td> <td></td> <td></td> <td></td>	4100		NUM)					
REAC(3,9464) NUPH GUE 02433 REAC(3,9464) NUPH GUE 02443 REAC(4,9464) NUPH GUE 02445 REAC(4,9464) NUPH GUE 02450 REAC(4,9464) NUPH GUE 02450 REAC(4,9464) NUPH GUE 02450 REAC(4,9464) NUPH GUE 02450 CC 480 I=1+L GUE 02450 If (HVAME(I).EG.BLANK) 60T0 480 GUE 02460								
REAC(3,9465) (LOCH(I),I=1,NUMH) GUE02440 REAC(4,9465) NJMH GUE02450 REAC(4,9465) (LOCHS(I),I=1,NUMH) GUE02460 DC 4R0 I=1,L GUE02470 IF (HVAME(I),EG,BLANK) E0TO 4E0 GUE02480 * 01TE(3,9461) HNAME(I),IDEX(I,1),RULEH(I),QDEX(I) GUE02490 14=IDEX(I,1) GUE02490 IC(4,00,0) G0TO 480 GUE02290 IF (14,Eq.0) G0TO 480 GUE02290 IF (14,Eq.0) G0TO 480 GUE02500 IF (14,E0,0) G0TO 480 GUE02500 VRITE(3,9462) HNAME(I3) GUE02500 VRITE(3,9462) HNAME(I3) GUE02600 VRITE(4,9463) (HNAME(I3),SCOREH(I3),I3=1.L) GUE02600 VRITE(4,9463) (HNAME(I3),SCOREH(I3),I3=1.L) GUE02660 VRITE(4,9463) (HNAME(I3),SCOREH(I3),I3=1.L) GUE02660 VRITE (4,9463) (HNAME (I0),SCORE GUE02600 <td></td> <td>the state of the s</td> <td>A ATTRA</td> <td></td> <td></td> <td></td> <td></td> <td></td>		the state of the s	A ATTRA					
REAC(4,9464) NJMH GUE 02450 REAC(4,9465) (LDCHS(I),I=1,NUMH) GUE 02460 DC 460 I=1+L GUE 02460 IF (HYAME(I).EG.BLANK) E0TO 460 GUE 02480 .4 ITE (3,9461) HNAME(I).IDEX(I,1).RULEH(I).GDEX(I) GUE 02490 IF (HYAME(I).EG.BLANK) E0TO 460 GUE 02490 .4 ITE (3,9461) HNAME(I).IDEX(I,1).RULEH(I).GDEX(I) GUE 02490 IF (14.EG.60) GOTO 460 GUE 02490 LGC=LUC+1 GUE 02500 ISIDEX(I,10.50)+1 GUE 02550 DC 481 I2=2.11 GUE 02550 IF (14.LE.50) GCTO 483 GUE 02550 IF (13.9462) HNAME(I3) GUE 02550 GCTU 481 GUE 02560 WR ITE (3,9462) NOMO GUE 022600 J=LCCHS(HNUM) GUE 022600 WR ITE (4.9463) (HNAME (I3).SCOREH (I3).I3=1.L) GUE 022600 J=LCCHS(HNUM) GUE 022600 WR ITE (4.9463) GSCORE GUE 022600 948 (FORMAT (I0F 8.5) GUE 022600 GUE 02				NUMBER				
REAC(4,9465) (LOCH3(I),I=1,NUMH) QUE02460 CC 4R0 I=1,L QUE02470 IF (HNAME(I),EG.BLANK) E0T0 4E0 QUE02470 GUE02470 QUE02470 IF (HNAME(I),EG.BLANK) E0T0 4E0 QUE02470 GUE02490 GUE02490 IF (IA,S,9461) HNAME(I),IDEX(I,1),RULEH(I),SDEX(I) GUE02490 IF (IA,EG.0) GOTO 480 QUE02500 LC=LCC+1 GUE02500 ISIDEX(I,10) GUE02500 DC 481 I22211 GUE02550 IF (IA,LE.50) GCTO 483 GUE02550 GOTO 481 GUE02550 GOTO 481 GUE02550 4F2 WITE(IS,9462) HNAME(I3) GUE02560 GOTO 481 GUE02560 WR ITE(IS,9462) HNAME(I3),SCOREH(I3),I3=1.L) GUE02660 GOTO 4600 GUE02670 GUE02670 GOTO 4600 GUE02670 GUE02670 GOTO 4600 GUE02670 GUE02670 GUE02670 GUE02670 GUE02670 GUE02670 GUE02670 GUE02670 GUE02670 GUE02670 GUE02670 GUE02670 GUE02670 GUE02670 GUE02710				(NUNN)				
DC 4R0 I=1.L QUE02470 IF (HNAME(I).EG.BLANK) 60T0 460 GUE02480 *TTE(3.9461) HNAME(I).IDEX(I.1).RULEH(I).GDEX(I) GUE02490 I4=IDLX(I.1) GUE02480 IF (I4.20.0) GOTO 460 GUE02500 LC=LC(1) GUE02500 II=MOC(IDEX(I.1).E0)+1 GUE02500 DC 481 I2=2.11 GUE02500 IS=IDEX(I.12) GUE02500 GCTU 481 GUE02500 WR ITE(3.9462) HNAME(I3) GUE02601 J=JCCKS(HNUK) GUE02620 WR ITE(4.9463) (HNAME(I3).SCOREH(I3).I3=1.L) GUE02620 J=JL/5+1 GUE02600 WR ITE(4.9463) GSCORE GUE02600 948(FCRMAT(10F6.5) GUE02600 GUT0 4C00 GUE02710 GUE02720 WR ITE(6.99271				1.NIIKH1				
IF (HVAME(I).EG.BLANK) COTO 460 GUE02483 •SITE (3,9461) HNAME(I),IDEX(I,1).RULEH(I),QDEX(I) GUE02493 I4=IDEX(I,1) GUE02503 IF (14.E0.0) G3T0 480 GUE02513 L0C=LCC+1 GUE02530 I3=IDEX(I,1),E0)+1 GUE02530 OC 481 12=2,I1 GUE02553 IF (14.LE.50) GCTO 483 GUE02550 IF (14.LE.50) GCTO 483 GUE02550 GCTU 481 GUE02550 GCTU 481 GUE02550 GCTU 481 GUE02550 UF (14.LE.50) GCTO 483 GUE02550 GCTU 481 GUE02550 UF (14.LE.50) GCTO 483 GUE02560 WRITE (3.9462) HNAME (I3) GUE02560 GUE 22610 GUE02560 WRITE (3.9462) HNAME (I3),SCOREH(I3),I3=1.L) GUE02620 J=J+L/5+1 GUE02660 WRITE (4.9480) GSCORE GUE02660 9480 FCRMAT(L0F8.5) GUE02670 GOTO 4C00 GUE02670 GUE02660 GUE02660 9480 FCRMAT(L0F8.5) GUE0270 GOTO 4C00 GUE02710 WRITE (6.9271) GUE02710 WRITE (6.9271)				- I THOPHIN				
<pre># RITE(3,9461) HNAME(1),IDEX(1,1),RULEH(I),GDEX(I) GUE02493 I4=IDEX(I,1) GUE02500 IF(14+E0+0) GJT0 480 UUE2513 UUE2E13 UUE02540 II=MOD(IDEX(I,1),50)+1 GUE02540 GUE02540 GUE02540 GUE02540 GUE02540 GUE02540 GUE02540 GUE02540 II=MOD(IDEX(I,1),50)+1 GUE02540 GUE02740 GU</pre>				TO 480				
I4=IDEx(I,1) GUE02500 IF(I4+E0+0) GOTO 486 GUE02510 LU(=LUC+1 GUE02520 I1=MDC(IDEX(I,1),50)+1 GUE02530 DC 481 I2=2+I1 GUE02550 IF(I4+LE-50) GCTO 483 GUE02550 IF(I4+LE-50) GCTO 483 GUE02550 IF(I4+LE-50) GCTO 483 GUE02550 IF(I4+LE-50) GCTO 483 GUE02550 IF(I4+G-9462) HNAME(I3) GUE02550 GOTU 481 GUE02550 GOTU 481 GUE02550 WRITE(3+9462) HNAME(I3) GUE02550 WRITE(3+9462) NOMD GUE02660 JELCCHS(HNUM) GUE02620 WRITE(4+9463) (HNAME(I3)+SCOREH(I3)+I3=1+L) GUE02660 JELCHS(HNUM) GUE02660 WRITE(4+9463) GSCORE GUE02660 948(FCRMAT(10F6.5) GUE02660 GOTO 4C00 GUE02660 GUE02660 948(FCRMAT(10F6.5) GUE02670 GOTO 4C00 GUE02670 GUE02710 WRITE(6+9271) GUE02710 GUE02720 READ(6+9462) HNAME(ID) GUE02730 GUE02730 GOTO 4L00 GUE02730 GUE0					EH(1).90F	(1)		
1F (14.EG.0) GOTO 480 QUEC2510 LUCELUC+1 QUEC2520 11=MOD (IDEX(I,1),50)+1 QUEC2530 DC 481 12=2,11 GUE02540 I3=IDEX(1,12) GUE02550 IF (14.LE.50) GCTO 483 QUE02560 WRITE (3,9462) HNAME (I3) GUE02570 GOTU 481 GUE02580 4F3 WRITE (3,9462) HNAME (I3) GUE02560 4F4 UCCTINUE GUE02560 WRITE (3,9462) HNAME (I3) GUE02600 4F4 UCCTINUE GUE02600 WRITE (4,9463) KNM3 GUE02600 WRITE (4,9463) KNM3 GUE02600 WRITE (4,9463) GSCORE GUE02660 9480 F CRMAT(10F6.5) GUE02660 9480 F CRMAT(10F6.5) GUE02660 GOTO 4000 GUE02670 GUE02660 GUE02670 GUE02670 GUE02660 9480 F CRMAT(10F6.5) GUE02670 GOTO 4000 GUE02670 GUE02670 GUE02670 GUE02670 GUE02670 GUE026710 GUE02670 GUE0269710 GUE02670 GUE02710 GUE02720 READ(5,9462)								
LUC=LUC+1 0UE02520 11=MUD(IDEX(I,1),50)+1 0UE02530 DC 481 12=2,11 0UE02530 DC 481 12=2,11 0UE02530 IF(I4.LE.50) GCT0 483 0UE02550 IF(I4.LE.50) GCT0 483 0UE02560 % HITE(3,9272) GNAME(I3) 0UE02590 4F3 WRITE(3,9462) HNAME(I3) 0UE02590 4F1 LUC=LUC+1 0UE02610 4F3 CCNTINUE 0UE02610 WRITE(3,9462) NOMD 0UE02610 URE02620 J=LCCHS(HNUM) 0UE02610 WRITE(4,9463) (HNAME(I3),SCOREH(I3),I3=1.L) 0UE02620 J=LCHS(HNUM) 0UE02640 VKITE(4,9480) GSCORE 0UE02650 WKITE(4,9480) GSCORE 0UE02660 948C FCRMAT(10F6.5) 0UE02660 948C FCRMAT(10F6.5) 0UE02670 GOTO 4C00 0UE02680 0UE02680 0UE02680 0UE02680 0UE02680 0UE02670 GUT0 4C00 0UE02720 READ(6,9462) HNAME(ID) 0UE02730 0UE02740		IF (14.EQ.0) GOTO 480					
DC 481 12=2.11 GUE02540 13=1DEx(1,12) GUE02550 IF (14.LE.50) ECTO 483 GUE02560 WRITE(3.9272) GNAME(13) GUE02580 GCTU 481 GUE02580 4F2 WRITE(3.9462) HNAME(13) GUE02560 4F3 WRITE(3.9462) HNAME(13) GUE02650 4F4 WRITE(3.9462) NOMD GUE02660 JELCCHS(HNUM) GUE02620 GUE02620 JELCCHS(HNUM) GUE02630 GUE02640 JELCCHS(HNUM) GUE02640 GUE02640 JELCCHS(HNUM) GUE02640 GUE02660 JELCCHS(HNUM) GUE02660 GUE02660 VRITE(4.9480) GSCORE GUE02660 GUE02670 GUE02660 GUE02660 GUE02660 9480 FCRMAT(10F8.5) GUE02670 GOTO 4000 GUE02670 GUE02680 GUE02660 GUE02670 GUE02700 GUE02670 GUE02710 GUE02700 VL5 GUE02710 GUE02710 KRITE(6.9271) GUE02720 GUE02730 GUE02730 GUE02730 GUE02730 GUE02730 GUE0		LUC=LUC+1						QUE02520
13=1DEX(1,12) GUE02550 IF (14+LE+50) GCT0 483 GUE02560 KhITE(3,9272) GNAME(13) GUE(2570 GCT0 481 GUE02580 4F2 WRITE(3,9462) HNAME(13) GUE02580 4F3 LCC=LOC+1 GUE02560 4F4 CCNTINUE GUE02600 URITE(3,9462) NOMD GUE02610 URITE(3,9462) NOMD GUE02620 URITE(4,9462) NOMD GUE02620 URITE(4,9463) (HNAME(I3),SCOREH(I3),I3=1.L) GUE02620 URITE(4,9460) GSCORE GUE02670 9481 FCRMAT(10F8.5) GUE02670 GOTO 4000 GUE02670 GUE02670 9481 FCRMAT(10F8.5) GUE02670 GOTO 4000 GUE02670 GUE02670 9481 FCRMAT(10F8.5) GUE02670 GOTO 4000 GUE02700 GUE02700 GUE02660 GUE02700 GUE02700 GUE02710 GUE02710 GUE02710 READ(5,9462) HNAME(ID) GUE02730 GOTO 4100 GUE02730		I1=MUD(IDE	X(I+1)+50)+1					QUE02530
IF (14.LE.50) GCTO 483 GUE 02560 WR ITE (3.9272) GNAME(13) GUE 02580 GCTU 481 GUE 02580 4F3 WR ITE (3.9462) HNAME (13) 4F4 GUE 02590 GCTU 481 GUE 02580 4F4 GUE 02590 GCTU 481 GUE 02590 4F4 GUE 02590 GUE 02560 GUE 02590 GUE 02560 GUE 02560 GUE 02560 GUE 02560 GUE 02560 GUE 02560 GUE 02560 GUE 02560 GUE 02560 GUE 02670 GUE 02670 GUE 0270 GUE 0270 GUE 0270 GUE 02710 GUE 0270 GUE 02710 GUE 02720 GUE 02720 GUE 02720 GUE 02720 GUE 02720 GUE 02730 GUE 02730 GUE 02730 GUE 02730 GUE 02740 GUE 02740 <td></td> <td>DC 481 12=</td> <td>2.II</td> <td></td> <td></td> <td></td> <td></td> <td>QUE02540</td>		DC 481 12=	2.II					QUE02540
kkite(3,9272) GNAME(13) GUE(2570 GOTU 401 GUE02580 483 write(3,9462) HNAME(13) GUE(2650 481 LCC=LOC+1 GUE(2660 48.1 LCC=LOC+1 GUE(2610 WRITE(3,9462) NOMO GUE02620 J=LCCHS(HNUM) GUE02620 WRITE(4,9463)(HNAME(13),SCOREH(13),13=1+L) GUE02620 J=J+L/5+1 GUE02650 WKITE(4,9480) GSCORE GUE02660 948(FCRMAT(10F6.5) GUE02660 948(FCRMAT(10F6.5) GUE02680 GUE02690 GUE02600 GUE02670 948(FCRMAT(10,HNAME) GUE02680 948(FCRMAT(10,HNAME) GUE0270 GUE02710 GUE02710 GUE02710 GUE02710 GUE02710 GUE02720 GUE02710 GUE02720 GUE02720 GUE02720 GUE02720 GUE02730 GUE02730 GUE02730 GUE02730 GUE02740 GUE02740 GUE02740								GUE02550
GCTU 481 GUED2580 483 WRITE(3,9462) HNAME(I3) GUED2560 481 LCC=LOC+1 GUEC2660 482 CCNTINUE GUED2610 WRITE(3,9462) NOMD GUED26210 J=LCCHS(HNUM) GUED2620 WRITE(4,9463) (HNAME(I3),SCOREH(I3),I3=1+L) GUED2640 J=J+L/5+1 GUED2660 948(FCRMAT(10F6.5) GUED2670 GUTD 4 CDU GUED2670 GUED2660 948(FCRMAT(10F6.5) GUED2670 GUTD 4 CDU GUED270 GUED270 GUTD 2 CALL GETHN(ID, HNAME) GUED2710 WRITE(6.9271) GUED2720 GUTD 4 L00 GUED2730 GUTD 2740 GUED2740								QUE02560
483 WRITE(3,9462) HNAME(I3) QUEC2550 481 LCC=LOC+1 QUEC2600 480 CCNTINUE QUEC2610 WRITE(3,9462) NOMO QUEC2610 J=LCCHS(HNUM) QUE02620 WRITE(4,9463)(HNAME(I3),SCOREH(I3),I3=1.L) QUE02650 J=J+L/5+1 QUE02660 948(FCRMAT(10F6.5) QUE02670 GOTO 4 COD QUE02680 C CPTICN 49 TO CHANGE NAME OF BOX QUE02710 40U5 CALL GETHN(ID,HNAME) QUE02710 WRITE(6,9271) QUE02730 GOTO 4 COD QUE02730 QUE02730 QUE02740			72) GNAME(13)					GUE (2570
481 LCC=LOC+1 QUEC2660 480 CCNTINUE QUEC2610 WRITE(3,9462) NOMO QUEC2620 J=LCCHS(HNUM) QUEC2620 WRITE(4,9463)(HNAME(I3),SCOREH(I3),I3=1.L) QUEC2650 J=J+L/5+1 QUEC2660 WKITE(4,9480) GSCORE QUEC2660 948(FCRMAT(10F6.5) QUEC2670 GOTO 4 COD QUEC2680 C CPTICN 49 TO CHANGE NAME OF BOX QUEC2710 4005 CALL GETHN(ID,HNAME) QUEC2710 WRITE(6,9271) QUEC2720 GOTO 4(D0) QUEC2730 QUEC2740 QUEC2740		and the second						QUED2580
4FJ CCNTINUE QUE22610 WRITE(3,9462) NOMO QUE02627 J=LCCHS(HNUM) QUE02630 WRITE(4,9463)(HNAME(I3),SCOREH(I3),I3=1+L) QUE02640 J=J+L/5+1 QUE02650 WKITE(4,9480) GSCORE QUE02660 948(FCRMAT(10F6.5) QUE02670 GOTO 4CD0 QUE02680 C CPTICN 49 TO CHANGE NAME OF BOX QUE0270 4005 CALL GETHN (ID, HNAME) QUE02710 WRITE(6,9271) QUE02720 READ(5,9462) HNAME(ID) QUE02730 GOTO 4L00 QUE02740			62) HNAME(13)					
WR ITE(3,9462) NOMO GUED2627 J=LCCHS(HNUM) GUED2630 WR ITE(4,9463) (HNAME(I3),SCOREH(I3),I3=1+L) GUED2640 J=J+L/5+1 GUED2660 WK ITE(4,9480) GSCORE GUED2660 948(FCRMAT(10F6.5) GUED2680 GOTO 4CD0 GUED2680 C CPTICN 49 TO CHANGE NAME OF BOX GUED2680 4005 CALL GETHN (ID, HNAME) GUED2710 WR ITE(6,9271) GUED2720 GOTO 4LD0 GUED2720 GOTO 4LD0 GUED2720								
J=LCCHS(HNUM) GUED2630 WRITE(4,9463)(HNAME(I3),SCOREH(I3),I3=1.L) GUED2640 J=J+L/5+1 GUED2650 WKITE(4,9480)GSCORE GUED2660 948(FCRMAT(10F8.5) GUED2670 GOTO 4C00 GUED2680 C CPTICN 49 TO CHANGE NAME OF BOX GUED2690 4005 CALL GETHN(ID,HNAME) WRITE(6,9271) GUED2720 GOTO 4(00 GUED2730 GOTO 4(00 GUED2740	44.0							
WRITE(4,9463)(HNAME(I3),SCOREH(I3),I3=1.L) GUE02640 J=J+L/5+1 GUE02650 WKITE(4,9480) GSCORE GUE02660 948(FCRMAT(10F8.5) GUE02670 GOTO 4C00 GUE02680 C CPTICN 49 TO CHANGE NAME OF BOX GUE02690 4005 CALL GETHN(ID,HNAME) GUE02700 WRITE(6,9271) GUE02720 READ(5,9462) HNAME(ID) GUE02730 GOTO 4(00 GUE02740								
J=J+L/5+1 GUE02650 WK ITE (4,9480) GSCORE GUE02660 948(FCRMAT(10F6.5) GUE02670 GOTO 4C00 GUE02680 C CPTICN 49 TO CHANGE NAME OF BOX GUE02690 C CPTICN 49 TO CHANGE NAME OF BOX GUE02700 4005 CALL GETHN (ID, HNAME) GUE02710 BEITE (6,9271) GUE02720 GOTO 4(00 GUE02730 GOTO 4(00 GUE02740								
WK ITE (4,9480) GSCORE QUE 02660 948(FCRMAT(10F8.5) GUE 02670 GOTO 4C00 GUE 02680 C CPTICN 49 TO CHANGE NAME OF BOX QUE 02690 4005 CALL GETHN (ID, HNAME) QUE 02700 WR ITE (6,9271) GUE 02720 READ (5,9462) HNAME (ID) GUE 02730 GOTO 4(00 GUE 02740				SCUREM(15) +1	13=1+11			
948(FCRMAT(10F6.5) GUE02670 GOTO 4CD0 GUE02680 C CPTICN 49 TO CHANGE NAME OF BOX GUE02690 4005 CALL GETHN(ID, HNAME) GUE02700 WRITE(6,9271) GUE02720 READ(5,9462) HNAME(ID) GUE02730 GOTO 4(00 GUE02740								
GOTO 4CDU GUED2680 C CPTICN 49 TO CHANGE NAME OF BOX GUED2690 4005 CALL GETHN (ID, HNAME) GUED2700 WRITE(6,9271) GUED2720 READ(5,9462) HNAME(ID) GUED2730 GOTO 4LD0 GUEC2740	9440							
C CPTICN 49 TO CHANGE NAME OF BOX QUE02690 4005 CALL GETHN(ID, HNAME) QUE02700 WRITE(6,9271) QUE02720 READ(5,9462) HNAME(ID) QUE02730 GOTO 4000 QUE02740								and the second life from our second second
C CPTICN 49 TO CHANGE NAME OF BOX QUE02700 4005 CALL GETHN (ID, HNAME) QUE02710 WRITE(6,9271) QUE02720 READ(5,9462) HNAME(ID) QUE02730 GOTO 4000 QUE02740								
40U5 CALL GETHN (ID + NAME) GUE02710 WRITE(6+9271) GUE02720 READ(5+9462) HNAME(ID) GUE02730 GOTO 4L00 GUE02740	(C	PTICN 49 TO	CHANGE NAME O	F BOX				
WRITE(6,9271) GUE02720 READ(5,9462) HNAME(ID) GUE02730 GOTO 4L00 GUE02740								
READ(5,9462) HNAME(ID) QUE02730 GOTO 4100 QUE02740		LRITE (6.92)	11)					
GOTO 4100 QUEC2740		READ (5.9462	HNAME (ID)					
END QUE02750		GOTO 4LDO						
		END						QUE02750

		QUE 3 0010
	SUBROUTINE GETHN(ID, HNAME)	QUECCC20
C SUBR	OUTINE GETHN INTERACTIVELY REQUESTS & BOX NAME AND RETURNS ITS	QUED0030
C ID N	UMEER.	QUELJC40
	DOUBLE PRECISION ALL .HNAME (40) .A	QUECOCSO
	DATA ALL/6HALL /	QUESSES
	ID=1	QUELCC70
	WRITE(6.900C)	QUECIOSO
9000	FERMAT(20H ENTER BOX NAME)	GUEDCO90
	READ(5,9001) A	GUEC0100
9001	FORMAT(1A6)	QUECO110
	IF (A.EG.ALL) RETURN	GUELC120 GUEC0136
	CALL ATGET (A, ID, HNAME)	QUECJ140
	IF(ID.EG.D) GOTO 2C	QUECC150
	RETURN	QUESCIES
	WRITE(6,9002)	QUESC170
9602	FERMAT(11H BAC NAME)	QUEDD180
	CALL PNAME (HNAME) GCTO 10	QUE00190
	END	QUECOZOC
	Enc	QUEC 521?
	SUBROUTINE ATGET (ANAME . ID . HNAME)	BUEUL220
C SUE	OUTINE ATGET CHECKS & BOX NAME AGAINST THOSE CUPRENTLY IN	QUE00236
	SYSTEM AND RETURNS ITS ID NUMBER (C IF NOT VALID).	QUEDO240
	DOUBLE PRECISION ANAME HNAME (40)	GUE36250
	DO 10 1=1.40	QUE00263
	IF (ANAME.EQ.HNAME(I)) GOTO 20	QUEST270
10	CONTINUE	QUE00280
	10=0	GUEDJ290
	RETURN	QUEC:300
20	1D=1	QUE00310
	RETURN	GUE (- 3323
	ENC	QUE?0330
		QUEDO340
	SUPROUTINE SCREH(ID)	QUECC35C
C SUE!	ROUTINE SCREH IS THE MASTER SUBROUTINE FOR SCORING ECKES	QUECO360
	CCPMON L1(100)+QSCOPE(100)+L2(100)+SCOREH(40)+RULEH(40)	QUEJC370
	COMMON IDEX (40.10)	QUESO383
	CALL MULTEV(IC)	QUECC390
102		QUEC0400
9000	FORMAT(13H THE SCORE IS:1F10.5)	QUEJ0410
	RETURN	QUE30420
	END	QUED0430
	이 같은 것 같아요. 그는 것 같아요. 아이는 것 같아요. 이 있 않 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?	GUEDG44C
1.5.1.6	SUEROUTINE BOX(ID)	QUE63450
C SUB	ROUTINE BOX SCORES BOXES WHICH HAVE QUESTIONNAIRES AS INPUT.	QUELO460
	CCPMON L1(100) .QSCORE(100) .L2(100) .SCOREH(40)	QUED0470
	COMMON RULEH(40) . IDEX(40.10)	QUE00480 QUE00490
	DIMENSION INDEX(40), RESP(100)	QUECO500
	NAT=IDEX(10,1)-49	QUECC510
	DO 101 I=2.NAT	QUECO520
101		QUE20530
	INCEX(1)=NAT-1	QUECO54C
	DC 102 I=1,10C RESP(I)=QSCCRE(I)	QUECC550
102	MEDILIT-ADEONE(1)	

QUEASIB FORTRAN P ID=WCCWR 12.27.28 WEDNESDAY 12 DECEMBER 1979 PAGE 1 NATIONAL CSS. INC. (SUNNYVALE DATA CENTER) SUNY

.

.

QUEAS18	FORTRAN P ID=WCCWR 12.27.28 HEDNESDAY 12 DECEMBER 1979	PAGE 2
	RULE=RULEH(ID)	QUEDES
	CALL TXTURE (ID .RULE . INDEX . RESP .SCORE)	QUECCS
	SCCREH(ID) = SCORE	QUECUS
	RETURN	QUEDOS
	END	QUECOS
		QUEDDE
	SUBROUTINE MULTEV(ID)	QUE036
	UTINE MULTEV IS THE HIGHEST LEVEL OF THE	QUEDOG
	D BOX SCURING ROUTINE	QUEDOG
	COMMON LL(300) .SCOREH(40) .RULEH(40) .IDEX(40.10)	QUEDDE
	DIMENSION INDEX(40)	QUEDOG
	NUFAT=IDEX(ID,1)	GUE 306
	IF (NUMAT.EQ.C) RETURN	QUEDCO
	IF (NUMAT.GT.50) GOTO 4C	QUECCE
	DC 10 1=1.NUMAT	QUED07
	11=1+1	QUECC7
	L=10£X (10, 11)	GUEDO7
	IF(SCOREH(L).GE.D) GOTO 10	QUE JO7
	CALL MULTEI(L)	QUEDC7
	CONTINUE	QUEDO7
	J1=NUMAT+1	GUECU7
	DO 30 J=1,J1	QUE 307
	INCEX(J)=IDEX(ID+J)	QUESC7
	CALL TXTURE(ID,RULEH(ID),INDEX,SCOREH,SCORE)	QUECC7
	SCCREH (ID) = SCORE	QUEDOS
	RETURN	QUEDES
		QUEDDA
	CALL BOX(ID) Return	QUESCE
	END	GUEJOS
	IND	
1 . A		QUESCB
	SUEROUTINE GET ON (ID, ONAME, NUMG)	QUEDGE
	TINE GETON INTERACTIVEL Y ACCEPTS A QUESTIONNAIRE NAME	QUECOS
	ETURNS ITS ID NUMEER	QUESOS
	COMMON LOCO(100)	BUESCA
	DIVENSION GNAME(10)	QUEC 39
	RITE(6,90C0)	GUEDO9
100.001.001.001	CRMAT(29H ENTER QUESTIONAIRE NAME)	QUEUOS
	EAD(5,9301) A	QUECOS
	CRMAT(1A4)	QUECOS
	IC=NG(A)	QUECUS
	(F(10.EQ.0) GCTO 10	QUE039
	F(ID.LT.D) RETURN	QUECOS
and the second se	=LOCG(ID)	aUE009
	WIND 1	QUE 0 C 9
	AC(1,9002) NUMG	QUEDIO
	CRMAT(10x,112)	QUE C10
	ETURN	QUE 313
	ALL PENAME (QNAME , NUMO)	QUE010
	CTO AC	QUEC10
F	ETURN	QUEDIO
E	ND	QUEDIC
		QUEDIC
Ŧ	UNCTION YESNO(Z)	QUEDICI
FUNCTI	CN YESNO RETURNS THE ANSWER TO A YES + NO QUESTION	QUEDIC
	NC. 1 FOR YES.	QUEDII

01	UEASIE	FORTRAN	P ID=WCCWR	12.27.28	WEDNESDAY 12	CECEMBER 1975	PAGE 3
		DATA X/1HN					AUEC111C
		CATA Y/1HY YESNO=U.	'				QUE01120 QUE01136
		READ(5.900					GUE31143
1.0	9000	FORMATCIAL					QUE 21152
		IF (Y.NE.A)					QUE01160 QUE01170
		YESNO=1.	0010 10				GUEC1160
		RETURN					QUE 31193
	10	WRITE (6,90	6.1.1				QUE01200
	9001		TYPE YES OR	NO 1			QUE01210
	7001	GCTO 20	1111 123 04	NO /			QUE61220
		ENC					QUE :1230
		e no					QUE01240
		FUNCTION G	ETNUMALONAH	IGH. TYPE)			QUE01250
C	FUNC				INTERACTIVELY		QUE01260
		and the second and the second s	FOR THE APPRO		and the second se		QUE01270
		REAL GETHU	IM				QUE01280
	10	READ (5	ERR=26) GETNU	IM			QUE01290
		IF (TYFE.E	G.O. AND. GETNU	M.NE.AINT(GET	NUM)) GOTO 5:		20501300
		IF (GETNUM	.GE . ALOW . AND .	GETNUM .LE . AHI	GH) RETURN		QUE 1310
		IF (TYPE.E	G.21 GOTO 40				QUE 01320
	35	WRITE (6.10	O) ALO. AHIGH	e			GUE 01330
	100	FCRMAT (26H	ENTER A NUM	BER BETWEEN	+1610.5.4HAND	+1610.5.3H)	QUE01340
		GCTJ 10					QUEC1350
			E.2) GOTO 30				QUE01360
	4.	CCATINUE					QUEC137C
		RETURI	a da fata da cara da cara da f				GUE01360
))) ALON AHIGH		المحدة المتعدلات	12	QUE01390
	101		I ENTER AN IN	TEGER BETWEEN	N. G10.5.4HAND	•610.5.3H==)	
		6070 10					QUE21410 QUE01420
		ENC					GUE01430
		CHERONTING	TATURE (NUM		19679.003		QUE01440
6	CHER	and the second sec	a construction of the second	a so the set of a set of the set of the set	A SPECIFIED		QUE01450
		ING RULE.	The controles ,	A SCOLE USING	A SPECIFICO		QUED1460
	Seen	The second se	(100).OSCORE (1	1003-11 1(580)	ODFX (40)		QUE01470
			INDEX(1) .RESP		ACCALLE?		QUE014E0
			2HHA . 2HSA . 2H		149/		QUE01490
		M=INDEX(1)					QUED1500
		N= P + 1					GUE01510
		DC 5 1=1.0	6				QUE01520
		IF (RULE.E.	G.TEXTS(1)) 6	CTO 6			QUEC1530
	E	CONTINUE					QUE01540
		WRITE (6.9	(00)				QUE01550
	900	FCRMAT(38)	H BAD RULE E	ACCUNTERED CO	MPUTING SCORE)	QUEC156C
		RETURN					GUEC1570
			26.30.40.50.7	0) I			GUE01560
	10	SCCRE=1.					2UE01590
		DC 11 I=2					QUE01600
	11		RE .RESPLINDEX	(1))			QUE01610 QUE01620
		RETURN					QUE01630
	20	SCCRE=1.					QUE01640
		CO 21 I=2	the second				QUE01650
	21	SCERESCO	RE*(RESP(INCE	x 1) / * 1 * 1			00000000

•	UEASI	B FORTRAN P ICHWCCWR 12.27.28 WEDNESDAY 12 DECEMBER 1975	PAUL 4
		(-1.//D. ++H-1.)	QUE01660
		C=1./(2.**H-1.)	QUE 01676
		SCORE=C*(SCORE=1+)	QUEDIEBO
		RETURN	GUEC1690
			QUE01760
	30	SCCRE=0.	QUEL1713
		DC 31 I=2+N	QUEC1720
	31	SCORE = SCORE + RESP(INCEX(I))	GUE 01730
		SCCRE=SCORE*(1./M)	QUE01740
	100	RETURN	QUE(1750
	40	SCORE=1.	QUE01763
	100	DC 41 I=2.N	QUES1770
	41	SCCRE=SCORE * (1 ** • 5 * RESP(INDEX(1)))	QUE01780
	4.8	SCCRE=(2.**M/(2.**M-1.))*(1.=SCORE)	QUE 31793
	60	CCATINUE	QUESIEDE
		RETURN	QJE0181C
	50	SCORE=1.	QUEC1820
			QUE01830
	51	SCCRE=SCORE+(1RESP(INDEX(I)))	QUE 01840
		SCCRE=1.=SCORE	QUE01850
	7.0	RETURN GEGDEX (NUM)	GUE51860
	10	ICaNG(G)	QUE 51670
		A=GSCORE(ID)	GUECISSC
		B=C.2	QUE01890
		D0 65 J=1+5	QUEC1900
		IF (A.LT.P) GOTC 66	QUE01910
		E=E+0.2	QUE01920
	15	CCNTINUE	GJE01936
		RULE=TEXTS(J)	QUE01940
	DC.	1=0	QUE01950
		GCTO 6	QUE01960
		ENC	90501970
		ENC	QUE01980
		SUFROUTINE MULTE1(10)	QUE01990
~	SULE 0	CUTINE MULTEI IS THE SECOND LEVEL OF THE BOX SCORING	QUECEDED
ř	CVCT	EF. IT IS A CLONE OF MULTEV, AS ARE MULTE2, MULTE3, AND MULTE4.	
	9191	COPMON LL(300) . SCOREH(40) RULEH(40) . IDEX(40.10)	QUES2020
		DIMENSION INDEX(40)	QUE 02030
		NUMAT = IDEX (ID, 1)	GUEG2040
		IF (NUMAT.EQ.C) RETURN	QUE02050
		IF (NUMAT.GT.50) GOTO 40	QUED2C60
		CC 10 I=1. NUKAT	QUEG2C7C
		i1=I+1	GUEC2086
		L=IDEx(ID,II)	QUE02090
		IF (SCOREH(L).GE.0) GOTO 10	QUE 62100
		CALL MULTE2(L)	QUEC2110
	11	CCNTINUE	QUE02120
		J1=NUMAT+1	QUE02130
		DO 30 J=1, J1	QUE0214 3
	3.0	INDEX(J)=IDEX(ID,J)	QUE02150
	50	CALL TXTURE(ID,RULEH(ID),INDEX,SCOREH,SCORE)	QUE 02160
		SCCREH(ID)=SCORE	GUE02170
		RETURN	QUEC? LBC
	40		QUED. 90
	2.4	RETURN	QUEDZLOD

QUEASIB FORTRAN P IC=UCCUR 12.2	27.	12	IC=WCCWR	P	FORTRAN	QUEAS18
---------------------------------	-----	----	----------	---	---------	---------

	ENC		QUE02210
			QUEC2220
	SUPROUTINE I	ULTE2(ID)	QUE02230
C	SUEROUTINE MULTE:	IS THE THIRD LEVEL OF THE BOX SCORINE SYSTEM	QUEU2240
	COMMON LL (3)	DU) + SCOREH(40) RULEH(40) + 10EX(40+10)	QUEC2250
	DIFENSION I	NDEX(4C)	QUE02260
	NUMAT= IDEX (10.1)	QUE .2273
	IF (NUMAT.EQ.	O) RETURN	QUEC2260
	IF (NUMAT.GT.	.50) GOTO 4C	QUE 0:290
	DO 10 I=1.N	JMAT	00202300
	I1=I+1		QUE02313
	L=IDEX(ID,I	1)	QUE02320
	IFISCOREMIL).GE.0) 60TC 10	QUED233C
	CALL MULTES	(L)	QUE02340
	10 CONTINUE		QUE 12350
	J1=NUMAT+1		GUE 2360
	DO 30 J=1.J		QUE02370
	3C INDEX(J)=ID		QUEC238C
	CALL TATURE	(ID +RULEH(IC) + INDEX + SCOREH + SCORE)	QUEC2390
	SCOREH(ID) =	SCORE	QUE024CC
	RETURN		QUE02410
	AC CALL BOX(ID)	QUEC242C
	RETURIX		QUE32430
	ENC		QUE 02440
		for all sectors and the sector of the sector	QUES2450
	SUBROUTINE		QUE32465
C		3 IS THE FOURTH LEVEL OF THE BOX SCORING SYSTEM.	QUED2470
		CO) + SCOREH (40) RULEH (40) + IDEX (40+10)	QUE02480
	DIMENSION I		QUE02495
	NUMAT=IDEX(I		QUE02500
	IF (NUMAT .EQ		QUE02510
		.50) GCTO 4C	QUES2525
	CO 10 I=1.N	UKAT	QUES2530
	I1=I+1		GUE02540
	L=IDEX(ID,I		GUE62550
).GE.0) GOTO 10	QUE02560
	CALL MULTEA	(L)	QUEC257C
	10 CONTINUE		QUE02580 QUE02590
	J1=NUMAT+1		QUE02600
	DO 30 J=1.J		QUE02610
	3C INDEX(J)=ID		QUE02620
		(ID,RULEH(ID),INDEX,SCOREH,SCORE)	QUE32630
	SCCREH(ID) =	SLUKE	GUE02640
	RETURN		GUE02650
	40 CALL BOX(ID	·	QUEC2660
	RETURN		QUE 32673
	END		GUE02656
	CU CO DUTTNE	HULTEATION	QUE02690
	SUEROUTINE	4 IS FIFTH LEVEL OF THE BOX SCORING SYSTEM.	QUEC27CO
c		00) +S UREH(40) RULEH(40) + IDEX(40+10)	QUEU2710
			QUE02720
	DIFENSION I NUFAT=IDEX(QUE 32730
	IF (NUMAT.EQ		GUE02743
		.50) GOTO 40	QUE 02750
	AF LIGONAL SOI		

QUEASIE	FORTRAN P ID=WCCWR 12.27.28 WEDNESDAY 12 DECEMBER 1979	PAGE 6
	DC 10 J=1.NUMAT	QUEC2760
	11=1+1	QUE 02775
	L=IDE X (ID, I1)	QUE02780
	IF (SCOREH(L).6E.0) GOTC 10	QUE02790
	CALL MULTES(L)	QUED2800
1 (CCNTINUE	GUE02810
	J1=NUMAT+1	QUE02820
	CC 30 J=1.J1	QUE02430
	INDEX(J)=IDEX(ID,J)	QUEC2840
		QUE02850
	SCCREH(ID)=SCORE	GUEGEBED
	RETURN	QUE 02870
	D.F. TILD I	QUE02380
	F & F	QUEC2990
		QUED2900
		GUE02910
	FUED DUINTLE CENTERS AN ACCESS	QUE 32920
e europa	SUEROUTINE GETR(ILOC, RESP)	QUE02935
C LOCAT	UTINE GETR RETRIEVES THE RESPONSES FROM A SPECIFIED QUESTICHAAIRE	
C LUCPI	The Property and the second seco	QUE02950
	10 11 0 0 - 1	GUED2960
		GUE02570
		QUE02985
	D.C. TILL	QUE02996
		QUEC3000
		QUE03010
	ALL DE ALL BELLE ALL BEELLES	GUEC3020
		QUE0303C
L SUERU		GUE63040
		QUE03050
		GUE0336C
911	FCRMAT(54H YOU HAVE HIT THE LOWEST LEVEL IN THE SCORING ROUTINE) FORMAT(45H PLEASE TRY SCORING LOWER LEVEL BOXES FIRST)	
		QUE03080
		GUE03090
	5. 17 Kr	GUEC3100

QUEAS 38 FORTRAN P IC=WCCWR 12.27.38 WEDNESDAY 12 CECEMBER 1975 PAGE 1 NATIONAL CSS. INC. (SUNNYVALE DATA CENTER) SUNY

QUEDCOID

QUESCC2C SUPROUTINE SCOREG(ID) QUECCO30 GUEDDD40 C SUBROUTINE SCORES SCORES QUESTIONNAIRES. COMMON LOCO(100) + OSCORE (100) + LOCR (100) QUE 30350 QUESSOND DIMENSION IRESP(40) + RESP(40) + RULE(10) + IWRST(40) + WEIGHT(40) DIFENSION SCORE(4,),SCC(40),INDEX(40) QUECCOTO QUEDUDBO INTEGER GDEX(10.4C) 6020090 DATA IEEST/IHA/ QUE00100 BEST=FLOAT(IBEST) CALL GETQ(LOCU(ID) .RULE . IWRST .WEIGHT .NUMQ .NUMGP.QDEX) QUEC 3110 CALL GETR (LOCR (ID) , IRESP) 9912030300 QUECC130 CC 10 I=1 . NUMG GUE00146 IF (IWRST(I).EG.IBEST) GOTO 10 RESP(1)=(BEST-FLOAT(IRESP(1)))/(BEST-FLOAT(IWPST(1))) QUEC0150 RESP(I)=1. - #EIGHT(I) * RESP(I) QUE00160 CONTINUE QUECC17L 10 CC 150 I=1,10 QUEDCIPC 150 SCCRE(1)==1. QUECCISS DC 140 11=1.10 QUE 66202 FLAG=0 . QUESS210 CC 100 I=1.NUMEP QUEL0220 J1=QDEX(1.1?+1 GUE0(230 00 110 J2=2.J1 GUEDC240 QUEDC250 J3=00EX(1, J2) QUEC0260 IF (J3.GE.100) GOTO 115 IF (SCCRE(J3).LT.0) GOTO 120 QUECC270 115 CONTINUE QUEC0280 110 CONTINUE QUECC290 QUECO300 DC 130 J4=2.J1 J5=QDEX(I,J4) QUECO310 1F(J5.LT.100) COTC 160 QUE00320 QUED0330 J5=J5-100 QUE00340 SCC(J4)=RESP(J5) QUEL 3350 GCTC 170 16C SCC(J4)=SCORE(.5) QUEDC360 170 INCEX(J4)=J4 GUECO370 130 CONTINUE QUECO3PC QUE30390 FLAG=1. INCEX(1)=QDEX(1+1) QUEDGADC CALL TXTURE(I.RULE(I).INDEX.SCO.SCORE(I)) QUEDG410 QUF00420 1F(1.E6.1) GOTO 145 120 CONTINUE QUE 30430 GUEUC440 100 CONTINUE IF (FLAG.EQ.0) 60TO 145 GUED0450 QUEDCA62 IF (SCORE (1).GE.0) GOTO 145 QUEGO470 140 CONTINUE QUE00480 escore(ID)=SCORE(1) 145 WRITE(6,9030) OSCORE(IC) QUE00490 QUECOSOD FORMAT(13H THE SCORE =+1F6.3) 9000 QUE01510 RETURN QUED0520 END QUEC0530 SUEROUTINE GETG(LOC, FULE + IWRST, WEIGHT, NUMQU, NUMGP, GDE X) QUEDC540 C SUBROUTINE GETG READS THE STRUCTURE OF A QUESTIONNAIRE OFF DISC FILE 1QUE00550

QUEAS	38 FORTRAN P IC=WCCWR 12.27.38 WECNESDAY 12 CECEMBER 1979	PAGE 2
	INTEGER ODEX(10.40).ISAVE(40).IWRST(40)	QUE00560
	DIMENSION RULE(10) (WEIGHT(40)	
	JELOC	QUEDC570
		QUECOSSO
	READ(1.9000) X.NUMQU.NUMGP	QUEC0590
9010	FORMAT (144+6X+112+6X+112)	QUEDOBCO
	J=J+1	QUEC0612
	READ(1.9002) IWRST	QUECC620
		QUESC630
9001	FORMAT(8F5.3)	GUEDIGAD
	D0 50 12=1.NUMGF	QUEDC650
	J= J+1	QUEDCREC
1.000	READ(1+9003) NAME+NUM+R	QUEC:670
9003	FCRMAT(2(112.8X).142)	QUEDC680
	NAME=NAME=49	GUEDL690
	QUEX(NAME + 1)=NUM	QUECO700
	RULE(NAME)=R	QUEDC710
	1+L=L	GUE00720
	READ(1,9004) (ISAVE(J2),J2=1,NUM)	QUEDO730
	DC 50 I=1,NUM	QUECC740
	11=1+1	GUED:750
		GUECO760
		QUEDC770
50	CCATINUE	QUECO780
	J=J+1	GUEDE790
	READ (1,9001)(WEIGHT(12),12=1,8)	GUE00800
	IF(WEIGHT(1).GT.1) GOTC 70	QUECOELC
	IF (NUMGU.LE.8) RETURN	QUE00820
	J1=J+1	QUECC830
	READ(1,9001)(WEIGHT(12),12=9,NUMQU)	GUECC840
	RETURN	GUEDOB50
	D0 100 J2=1,NUMQU	QUED0860
	WEIGHT (J2) = WEIGHT (2)	QUEDOS70
9004	FORMAT(4012)	QUECOSSO
	RETURN	20203309
	END	QUEDCODO
		QUE. 0910
		60200020
	SUEROUTINE PRINTG(ID, GNAME)	QUE00930
C SUBR	CUTINE PRINTS PRINTS THE STRUCTURE FOR A QUESTIONNAIRE.	GUECOP40
	COPMON LOCG(100) . GSCORE(100) . LOCR (100)	QUESC950
	INTEGER ODEX(10,40)	QUE00960
	CIPENSION RULE (10) RESP (40) . ON AME (100) . IWRST (46) . WEIGHT (40)	QUE00970
	DIMENSION IRESF(4L) . X (40) . Y (40)	GUELLOBRO
	DATA IBEST/1HA/	QUEDO990
	CALL GETG(LOCG(ID) . RULE . INRST . WEIGHT . NUMQU . NUMGP . QDEX)	QUEDICOC
	BEST=FLOAT(IBEST)	QUE01010
	CALL GETR(LOCR(ID), IRESP)	QUEC1020
	WRITE(6,8999)	QUEDICIC
	WRITE(6,9000) GNAME(ID)	QUE01040
8995	FORMAT (11.20X.22H QUESTIONAIRE DATA FOR)	QUE 31050
9000	FCRMAT(23X+13HQUESTIONAIRE +1A4)	QUE01060
	WRITE(6,9002) OSCORE(ID), RULE(1)	QUEC1070
9002	FORMAT(8x, 16HOVERALL SCORE = ,1F10.5, 5x, 7HRULE : ,144,//)	QUE61080
	DC 5 1=1,40	QUE01090
	IF (IWRST(I).NE.IBEST) GOTO 4	QUE 21100

QUE AS 3B	FORTRAN P ID=WCCWR 12.27.38 WEDNESDAY 12 DECEMBER 1975	PAGE 3
	x(I)=0.	QUE01110
	6010 6	GUE01120
	X(1)=1=(BEST=FLOAT(IRESP(1)))/(BEST=FLOAT(IWRST(1)))	QUE01130
	Y(I)=1wE16HT(I)+(1-X(I))	GUED1140
	CONTINUE	GUE01150
	I = 1	GUE011604
	J1=50	QUE 31173/
	WRITE(6,9004) J1.RULE(1)	QUEC1180
9004	FORMAT(7H BCX: +112+8H RULE: +142)	QUE01190
	11=QOEX(1+1)+1	QUE 31200
	IF (11.EG.1) GUTO 10	QUE .1210
	00 20 12=2.11	QUE 3122'
	WR ITE (6.9005)	QUE0123)
9005	FORMAT(7H?)	QUE01240
	13=60Ex(1,12)	GUEC125:0
	1F(13.6E.100) GOTO 30	QUE 31260
	J1=13+49	QUE 012 70
	#RITE(6,9004) J1.RULE(I3)	QUELIZIO
	14=QDEX(13+1)+1	GUEJ1200
	1F(14.EQ.1) GUTO 21	QUE01300
	EC 40 15=2+14	GUE(1310
	UR ITE (6,9005)	QUE 11:20
	WRITE(6,9515)	QUE 01330
	18=QDEx(13,15)	QUEC1340
	IN (18.CE.10C) CCTO 50	GUEC1550
	12=10+49	QUE01360
	WRITE(6,9004) J1,PULE(IE)	GUEC1370
	I6=QDEx(I3+1)+1	GUE01380
	IF(I6.EQ.1) 60T0 45	QUE01390
	DC 60 17=2.16	GUE01400
	D0 11 J2=1.3	GUE 21413
11	WRITE(6,9005)	QUEC1420
	J9=QDEx(IE+I7)	QUEC1430
	IF (J9.GE.100) GOTO 70	QUE01440
	J1=J9+49	WUE01450
	WRITE(6,9004) J1.RULE(J9)	QUE01460 QUE01470
	GOTO 60	GUE01480
70	J1=J9=10.0	QUE01490
	WRITE(6,9066) J1+X(J1)+WEIGHT(J1)+Y(J1)	GUE01500
9006	FORMAT(4H G=+112,7H RESP=+1F6.3,3H W=+1F6.3,3H S=+1F6.3)	GUE01510
6 5	CONTINUE	QUE01520
	GOTD 45	QUE01530
50	J1=I8=100 WRITE(6,9006) c1.x(J1).WEIGHT(J1).Y(J1)	GUE01540
+5	CCNTINUE	GUE01550
40	CONTINUE	QUEL1560
	GOTO 20	QUE 01570
30	J1=13-100	QUE01580
50	WRITE(6,9006) J1.X(J1),WEIGHT(J1),Y(J1)	GUE61590
20	CONTINUE	QUE61600
10	RETURN	QUE01610
	END	QUE01620

APPENDIX A4 DATA FOR EVALUATION EXAMPLE

This appendix contains additional information concerning the input and output for the example shown in Section 4.3. First, a table of component effectiveness test questionnaires is presented. Second, a sample guestionnaire showing the questions and multiple choice responses is illustrated. Third, the disk files for the computer program are listed. FILE1 contains the questionnaire structures. FILE2 contains the questionnaire responses. FILE3 contains the hierarchy structure for the right side of the capability. FILE4 contains the results of the computer run for the right side. FILL3 contains the hierarchy structure for the left side of the capability. FIL4 contains the results of the computer run for the left side. The mnemonics for the eft side correspond directly to the hierarchy boxes shown in Figure 1-1(1). Finally, additional output is presented for the left side of the capability hierarchy. (Note that the computer program simply references units 1,2,3 and 4. The user via computer system commands can make these units correspond to any file of his choosing).

9-20-79

COMPONENT EFFECTIVENESS TEST QUESTIONNAIRES

1. Admittance Authorization Criteria and Schedules 2. Admittance Authorization/Verification Procedures 3. Air and Utility Inlet Barriers 4. Annunciation Systems - Computer Assisted Annunciation Individual Alarm Annunciation Multiplex Alarm Annunciation 5. Area Zoning 6. Balanced Magnetic Switches 7. Breakwire Systems (Foil Strip and Grid Wire) 8. Buried Line Sensors - Seismic Magnetic Geophone String Piezo-electric String 9. Capacitance Alarms 10. CCTV Monitoring/Surveillance 11. CCTV Systems 12. Central and Secondary Alarm Stations 13. Close out Inspection by Third Party 14. Coded Credential System - Active Electronic Badge Reader Capacitance Coded Badge Reader Electric Circuit Badge Reader Magnetic Coded Badge Reader Magnetic Stripe Badge Reader Magnetic Strip Badge Reader Optical Coded Badge Reader Passive Electric Badge Reader 15. Commercial Telephone System Contingency Plans and Procedures 16. 17. Controlled Security Lighting 18. Data Link Via Radio Frequency 19. Direct-Line Telephone/Intercom 20. Direct Monitoring/Surveillance 21. Doors and Associated Hardware 22. Duress Alarms 23. E-Field Fence 24. Electret Cable and Tilt Switch Fence Systems 25. Emergency Access/Egress Procedures 26. Emergency Battery System 27. Emergency Evacuation Procedures 28. Emergency Exits 29. Emergency Generator Systems 30. Equipment Checks/Maintenance

31. Escorts 32. Explosive Detector - Hand Held Package Search 33. Explosive Detector - Hand Held Personnel Search 34. Explosive Detector - Hand Held Vehicle Search 35. Explosive Detector - Volume 36. Explosive Detector - Walk Through 37. Fence Systems 38. Floors 39. Functional Zoning 40. Gates and Associated Hardware 41. Guard Force Personal Equipment 42. Guard Force Qualification 43. Guard Patrols/Intervention 44. Guard Post Assignments 45. Hardware Video Systems 46. Infrared Beam Systems, Exterior Interface Between Alarm Station and Sensors 47. - Individual Hardwire Alarms - Multiple Hardwire Alarms - Hardwire Command Signals 48. Isolation Zones K-9s, Use of - Package Search 49. 50. K-9s, Use of - Vehicle Search 51. Local Audible/Visible Alarms 52. Locks - (Key Locks, Keyless Locks) 53. Manual Alarm Recording 54. Master Fixed Radio 55. Microwave Systems, Exterior 56. Mobile Radio 57. Motion Detectors - Infrared Systems, Interior Microwave Systems, Interior Ultrasonic and Sonic Systems 58. Multi-Man Rule 59. Night Vision Devices 60. Package Search - Visual Inspection 61. Pat-Down Search 62. Personal Identification Numbers/Passwords 63. Photo Identification Badges 64. Physical Controls and Procedures for Keys, Locks, Combinations, and Cipher Systems 65. Portable Radio Positive Personnel Identification 66. - Fingerprint Personnel I.D. Verification - Handwriting Personnel I.D. Verification - Hand Geometry Personnel I.D. Verification - Voice Print Personnel I.E. Verification Response Vehicles 67. 68. Roof

-2-

69. Sally Ports, Pedestrian

```
Sally Ports, Vehicular
70.
     Shielding Detector - Volume
71.
     Shielding Detector - Walk Through
72.
73. SNM Containers
74. SNM Detector - Hand Held Package Search
75. SNM Detector - Hand Held Personnel Search
76. SNM Detector - Volume
77. SNM Detector - Walk Through
78. SNM Holding/Storage Area
79. SNM Identification/Authorization Procedures
80. SNM Liquid and Solid Waste Handling Procedures
81. SNM Scrap Removal Procedures
82. SNM Shipping/Receiving Procedures
83.
    Tamper Indicating Circuitry
84. Tamper Indicating Seals and Tamper Seal Inspection
85. Team Zoning
    Uninterruptible Power System
86.
87.
    Vaults
88. Vehicle Search - Visual Inspection
89. Vibration Sensors
90.
     Walls
     Weapons - Handgun
91.
               Semi-Automatic
               Shotqun
     Weapons Detector - Hand Held Package Search
92.
     Weapons Detector - Hand Held Personnel Search
93.
    Weapons Detector . Volume
94.
   Weapons Detector - Walk Through
95.
96. Windows and Associated Hardware
97. X-Ray Package/Container Search
```

-3-

EFFECTIVENESS TEST

FUNCTION

The function of the annunciation system will be to alert security personnel to alarm activation.

CONDITIONS

Performance Conditions

Installation

 Where will peripheral equipment such as computers and communications electronics be located?

Operation

- How much console space will be occupied by primary controls and displays that require observation or action several times per shift?
- 3. Where will the primary control and display area be situated with respect to the operator?
- 4. Where will all primary controls be located with respect to their accessability to the operator?
- How will the operator's attention be directed to the annunciators?
- 6. Will security annunciators be monitored by the same operator who monitors other annunciators?
- Will the status of sensors (secure/access/alarm/tamper) within a security zone be available to the operator?
- 8. How will the importance or priority of an alarm be determined?
- 9. When an alarm occurs, to what extent will the sensor's location be available to the operator?
- 10. What additional information will be available to the operator if an alarm occurs?
- 11. To what extent will the annunciation system indicate multiple concurrent alarms?
- 12. How will significant events be recorded?

.

Reliatility

- 13. How frequently will the system be checked for proper operation?
- 14. What provisions will be made to maintain operational capabilities when critical elements, i.e., CPU, CRT, audio and visual devices, etc., fail?
- 15. If the system is equipped with self-test capability, what will be the test frequency?

Vulnerabilities

16. What techniques will be used to deter unauthorized modification of programs or data?

ANSWERS

CONDITIONS

Performance Conditions

Installation

- 1. a. In a separate access-controlled room.
 - b. In the same room but away from primary display and control area
 - In the same console area as the primary displays and con-C. trols.

Operation

- a. Less than 250 square inches. 2.

 - b. 250 to 700 square inches.
 c. 700 to 1700 square inches.
 d. More than 1700 square inches.
- 3. a. Approximately perpendicular to a seated operator's line of sight.
 - b. In a vertical plane.
 - c. In a horizontal plane.
- 4. a. Completely within convenient reach of the operator.
 - Partially within the operator's reach. b.
 - c. Not within reach from the operator's normal location and will require the operator to move from his location.
- a. By an audible signal which varies depending on type of alarm 5. plus visual indicators.
 - b. By an unchanging audible signal plus visual indicators.
 - c. By visual indicators only.
- a. No. 6.
 - b. Yes.
- 7. The status of each sensor will be available. 8.
 - The most significant status within a group of sensors will be b. available.
 - The most significant status within the security zone will be c. indicated.
 - Only the occurrence of an alarm will be indicated. đ.
- Automatically, by a hardware or software priority structure. By the operator in a predetermined priority structure. 8. a. b. By the operator using real-time judgment. c.
- 9. a. The location of the specific sensor in alarm will be available.

- b. The location of the sensor group containing the specific sensor in alarm will be available. c. The location of the general area containing the specific sensor in alarm will be available. The time of alarm,
 The priority of alarm,
 telephone numbers 10. a. 3. Emergency telephone numbers, 4. Special precautionary instructions associated with a zone, and 5. Area maps. b. 1., 2., 3., and 4. above.
 c. 1., 2., and 3. above.
 d. 1. and 2. above. e. Only 2. above. 11. a. It will advise the operator of multiple concurrent alarms. b. It will permit only a sequential display of multiple concurrent alarms. c. It will display only one of multiple concurrent alarms. a. They will be automatically printed out. 12. They will be recorded automatically and manually in combinab. tion. They will be manually recorded. c. d. They will not be recorded. Reliability a. Every few seconds.
 b. Every few minutes.
 - c. Every few hours.
 - d. Once per shift.
 - e. Once per day.

 - f. Once per week.
 g. Less than once per week.
- 14. a. A fully redundant system of annunciation is to be provided. b. Full redundancy is to be provided for all critical subsystems and computers.
 - c. Significant increase of patrols will be provided.
- 15. a. At 10- to 30-second intervals.
 - b. At 30- to 60-second intervals.

 - c. At 1- to 5-minute intervals.
 d. The system will not have sel The system will not have self-test capability.
- 16. a. By encryption.
 - b. By multiple passwords.

 - c. By single password.d. By administrative controls.
 - e. None.

FILES DATA P IC=WCCWR 16.20.50 THURSDAY 6 DECEMBER 1979 PAGE 2 04 50 SA 1234 2. .5 21 05 01 ABECC SA 56 64 2345 2. .5 01 11 22 свсвсвсврвс SA 50 11 1 2 3 4 5 6 7 8 91011 2. .5 25 19 01 BDCCEBCCCBBBBBCBCBB EC 19 SA 1 2 3 4 5 6 7 8 910111213141516171819 50 2. .5 28 13 61 BCEEEBABDDEDL 12 SI 50 1 2 3 4 5 6 8 910111213 2. .5 32 14 61 E D E C E B B B B B B C C B 14 SA 50 1 2 3 4 5 6 7 8 91011121314 2. .5 36 18 61 сссвввесвввавссвьо St 50 18 1 2 3 4 5 6 7 8 9101112131415161718 2. .5 36 04 01 ABCB 2 3 4 03 S A 50 2. .5 43 15 01 **B B B B C C B B B C C B B B B** 50 15 SA 1 2 2 4 5 6 7 8 9101112131415 2. .5 51 13 01 сссевссоросос 56 13 SA 1 2 3 4 5 6 7 8 910111213 SA 2. .5 01 **D С С D В В В С С С С В D С** SA 50 14 1 2 3 4 5 6 7 8 910111213141 2. .5 63 18 63 18 01 СССОВССВС ВОВВССС

.

FI	161	1			A		P	1	C =	¥C		R			16	• 2	٥.	50		TH	UR	SD		6	D	EC	EM	BE	R	19	79	1	PAGE	
j,																																		
				18												1																		
		3	2	5 6	7	8	91	01	11	21	31	41	51	61	71	8																		
2.			.5																															
66 B				14					1		÷.	0																						
56		C	B	E C 14	B	C	в	8.	A	C	c	U																						
1		100	4	5 6			91	01		21	* 1	4																						
2.			.5		1	0			**	c .	~.	÷.																						
68		1		03					1																									
	1	F																																
50	Ξ.	<u> </u>		62				S	4																									
2	3																																	
.5			.5																															
69				U6				0	1																									
	B		С	CE																														
50				05				S	4																									
1	2	3	4	5 6																														
2.			.5																															
74				14					1																									
Ł	D	B	B	£ 8	B	В	С	С	E	В	B	B																						
5.				14					4																									
		2	4	5 6	7	8	91	61	11	21	31	4																						
2.			.5																															
75	1			17					1																									
D	В		8	BB	C	C	C			8	6	B	B	c	8																			
50								5	A .																									
1		2	4	5 6	1	8	91	101	11	21	31	41	51	61	1																			
2.			.5																															
83	n		÷.	07 8 8				0	1																									
50	U	U	٢	6 67				s																										
1	2	2	4	5 6				3																										
2.	*																																	
84				69				0	11																									
	Б	C.	С	E B		D	0																											
50		~	~	69	17.	~		S	4																									
	2	3	4	5 6		8	9																											
2.			.5																															
87				05	÷ 1			C	1																									
	A	e	D	E																														
50				04	÷			S	A																									
1	3	4	5	6																														
2.		1	.5																															
90				03	1			L	1																									
A	B	D																																
50				02				S	A																									
2	3																																	
.5		- 3	• 5	1																														
95				29					1																									
		¢	c	CC						8	B	8	8	C	B	c	C	B	B	0	C	C	8	8	В	В	B							
50					-				A						-												-							
1.00				5 6	1	8	91	101	11	21	33	43	. 51	61	/1	181	93	:02	11	222	132	42	5	66	212		19							
2.		•	2																															
12		~	~	33			P		11		P			-		~	~	0	-		P	~	P	P	~	~	P	-	~	P	0			
c	0	5		BC	c	D	D	c	C	Ø	B	c	0	0	E.	C	C	U	U	8	c	6	B	B	С	0	0	0	c	B	0			

FI	LE	1		DA	TA		P	10=	wcc	WR		16.	20.	50	TH	URS	DAY	6	DE	CEM	BEF	19	79	PAG	5
50				3				SA												L.			d.		
1	2	3	4	5	6 7	8	911	0111	213	1415	161	718	192	021	222	324	2521	627	28	293	031	323	3		
2.33			.5																						
33				1 B (3			01																	
D	B	C	В	B	CB	B	Cf	3 C	CB	3															
50				1	3			SA																	
50 1	2	2	4	5	6 7	8	91	1111	213	1															
2.			5																						
AL	45			0				01																	
D	B	D	С	E	C C	E.																			
50				0	7			SA																	
D 50	2	3	4	5	6 7																				
2.			.5																						
PN	55			0				U1																	
B	8	C	Ð	E I																					
50				0	6			SA																	
50 1	2	3	4	5	6																				
2.			5																						
2 · 17 0 FL				1	2			01																	
U	D	2	Đ	DI	0 0	D	DE	BD																	
EL				5	1			SA																	
1	2	3	4	5	6 7	8	91	.11																	
2.16		,	E.																						
16				1	5			v 1																	
B	D	E	¢	8	C F	В	DE	C	DO	ED	1														
50				1				SA																	
	2	3	4	5 1	67	8	91	1111	213	1415															
5 .		,	5																						

									NC		(5)	UNI		YA		0	A 1 1	-	CEN	TER)			SUM	Y						
38																														
16	4	۰.,	6		8		10		12		14		16				20		22	24	26	28	30	32	34	36	38	40	42	
•		1	10	1	40		50		52	-	54		56	1	58	1	60	9	62	64	66	68	70	72	74	76	78	80	82	
A	B		A		A	A				A		8		B	в															
٤.			į.	J.	J.		1			J	J.	÷.																		
10	8	1	^	^	^	1	^	8	^	^	^	^	A	-																
A	B	Đ	A	A	A		B		A	A	A	A	A	A	A															
47					÷.	1	B	1																						
57		1	1	E	^	^	D	^	^																					
A	A		A	*	С	A		A	4	A	A	С	С			A	A	A	A											
۰ <u>،</u>	۵		A	٨																										
2	Ĩ.,	1		1																										
	A		A	A	4	B	A	A	A	4	A	A	A	A	A	A														
3 C	4	E	A	в																										
11																														
A 14	8	Α.	8	•	A	A	A	A	A	4	A	A	A	A	A															
A	A	۵	A	A	A	A	4	A	A																					
16																														
21	A	^	*																											
A	٨	۵	۸	6																										
22						~	B			,																				
25	~	1	6	1			0	2	c	1																				
A	A		٨		A	٨	A	A	A	A	A	₿		A		C		A												
82 A	٨	4		٨	4	۵	B	۵	в	в	4																			
32																														
	A	۵	A	A	A	e	A	A	A	A	A	A	A																	
36	В	4	A					A	A	A	A	A	A	A	A	A	A													
38																														
A 3	A	A	A																											
	A	٨	A	A	A	A	A	A			A	4	A	A																
51																														
A 60	*	^	e	*	^	B	A	^	*	*	A	A																		
	A		٨		A	A		С		A	A	8	A																	
63											1					١.														
66	•	1	e	1	^	^	*	^	^	•	•	^		*	C	A	A													
		A					A																							
68 A																														
69																														
	В		С		D																									
74																														
^		1	^	^	*	^	^	*	*	~	-	-	~																	

FILE2 DATA P ID= UCCUR 15.27.48 WEDNESDAY 12 DECEMBER 1979 PAGE 1 NATIONAL CSS. INC. (SUNNYVALE DATA CENTER) SUNY FILE2 DATA P ID=WCCWR 15.27.48 WEDNESDAY 12 DECEMBER 1979 PAGE 2 75 83 ABEBAAB 84 AAABAACAC 87 AAAB 90 ABA 95 12 33 DABAABA PNSS A A A A A A A A 17 18

01			
(003			
DENACC	2	SA	
DETACC			
RESACC			
DETACC	3	HA	
SENSE			
REPALR			
ASSESS			
RESACC	2	40	
COFRSP	1 T		
RESP			
SENSE	2	SC	
FULTS			
INCM1			
REPALR	2	HA	
TSIC		112	
ANALPP			
ASSESS	3	AV	
	5	A V	
MULTA			
INDA1			
CASSAS			
COFRSE	4	S &	
PETEDS			
GDSSTA			
EETSTN			
ONLFF			
PESP	3	SA	
DELRSP			
EFFRSF			
DRRSP			
MULTS	53	AV	
6			
57			
10			
INCM1	3	SA	
DDS			
6P			
BARR			
TSIG	52	sc	
47	26	50	
18			
ANALPY	53	\$0	
4			
51			
43			
MULTA	51	AV	
10			
INDA1	3	SA	
DDA			
GP			
BARR			
CASSAS	51	AV	
12			
6DSSTN	51	HA	
and the second se			

.

FILLE DATA PIDENCONR 16.21.11 THURSDAY 6 DECEMBER 1979 PAGE 1 NATIONAL CSS, INC. (SUNNYVALE DATA CENTER) SUNY

43		
DELRSP	2	SO
BARR		
GP		
EFFRSP	2	SA
ONSITE		
CFFSIT		
BARR	57	SA
3		
21		
28		
38		
68		
69		
96		
GF	51	AV
43		
ONSITE	2	SA
REGOFF		
CONADY		
OFFSIT	2	S A
RSPRES		
ENGACY		
REGOFF	52	HA
12		
16		
CONACY	52	SA
16		
43		
RSPREG	51	HA
16		1.2.1
ENGADV	51	S A
16		
NOMORE		

FILLE DATA P ID=WCCWR 16.21.11 THURSDAY 6 DECEMBER 1975 PAGE 2

FILE4 CATA PID=WCCWR 16.21.19 THURSDAY 6 DECEMBER 1979 PAGE 1 NATICNAL CSS. INC. (SUNNYVALE DATA CENTER) SUNY

01									
0003									
DENACC C.	ABOETAC	C 0.335RE	ESACC D.	730SENSE		EPALR D.			
ASSESS C.	E49COMRS	SP 1.DUGRE	ESP C.	73 JMULTS		NDM1 C.			
TSIG O.	SADANAL	M 0.523ML	ULTA D.	673INDA1		ASSAS 0.			
BETGES 1.	DUDGESSI	IN 1.630PS	ETSTN 1.	ODDONOFF	1.0000	ELRSF 0.	790		
EFFRSF D.	TC6DRRSP	1.0008	ARR 0.	3736P	1.0000	NSITE 0.	559		
OFFSIT 1.	DOOREGON	F D.338C0	ONADV 1.	DOORSPRE	G 1.COCE	NGADV 1.	000		
DDS D.	833DDA					A State	0.10.100		
0.75493	0.76562	0.66992 1	0.82014	0.57873	1.00000	0.91667	0.60556	0.82031	1.00000
1.00600	0.91111	0.23693 1	0.56250	J.51551	0.74998		1.0:000	1.00000	0.76560
0.51559	0.38672	1.00000 1	1.00000	0.54839	1.00000	1.00000	0.74639		C.73333
6.06667	0.33660	0.33838 1	1.00000	0.59842	1.00000	1.00000	1.00000	0.0	0.0
0.0	0.6	6.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	6.0	0.0 1	0.0	0.0	0.0	0.0	0.3	6.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	C.O	0.0	6.0	0.0
0.0	C.J	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.1	6.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	6.0	0.0	0.0	0.0	0.0	0.0	0.0	C.O	0.0

4

×

01			
0003			
CONACC	2	SA	
NORMAL			
EMERGE			
NORMAL	2	H A	
ADAUTH			
PROCON			
ADAUTH	51	AV	
2			
FRECCA	2	SA	
PERSCA			
MATERI			
PERSON	3	SA	
VERIF			
CONTRA			
RESPVI			
MATERI	3	SA	
VERIF2			
CONTRE			
RESPVI			
VERIF	54	AV	
14			
63			
2			
66			
CONTPA	51	AY	
95			
RESPVI	2	HA	
COPRSP			
RESP			
VERIF2	51	AV	
2			
CONTR2	52	S #	
32			
65			
RESP	3	SA	
DELRSP			
EFFRSF			
DRRSF			
NOMORE			

FIL3E DATA P ID=WCCWR 16.21.26 THURSDAY 6 DECEMBER 1979 PAGE 1 NATIONAL CSS. INC. (SURNYVALE DATA CENTER) SUNY

FILA CATA PIC=WCCWR 16.21.34 THURSDAY 6 DECEMBER 1979 PAGE 1 NATIONAL CSS. INC. (SUNNYVALE DATA CENTER) SUNY

C1									
0003									
CONACC 0.	EATNOR MA	L 0.442E		8324DAUT					
PERSCA C.	496MATER	1 0.6354	ERIF D.	826CCNTR	A 0.3378	ESPVI D.	832		
VERIF2 C.	S17CONTR	2 0.5510	OMRSP 1.	COORESP	C.8320	ELRSP 1.	.000		
FFFRSP D.	706DRRSP	1.000							
0.75493	0.76562	0.66992	0.82014	0.57870	1.00000	0.91667	0.60556	0.82031	1.00000
	6.91111	0.23693	0.56250	0.51551	0.74998	0.87500	1.00000	1.00000	0.76563
	0.38672		1.00000	0.54839	1.00000	1.00000	0.74639	0.63731	0.73333
0.66667	0.33660	0.33830		0.59842			1.00000	C.C	0.0
0.0	C.U	6.6	0.0	C.0	0.0	0.0	0.0	0.0	0.6
0.0	0.0	0.0	C.O	0.0	0.0	0.0	0.0	6.0	0.0
0.0	6.0	0.4	0	C.0	0.0	0.0	0.0	0.0	0.0
0.0	C.0	0.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	C.0	0.0	6.0	0.0
0.0	0.0	0.0	C.C	0.0	0.0	6.0	0.0	0.0	0.0

4

.

WHICH? 42 ENTER BOX NAME -- CONACC HIERAPCHY DATA FOR BOX CONACC BORCONACC RULE:SA SCORE: 0.547 0: BOKINDRMAL RULE: HA SCORE: 0.442 0: BOK: ADAUTH RULE: AV SCORE: 0.917 0: QUESTIONNAIRE: 2 SCORE: 0.917 BOMSPROCON RULE:SA SCORE: 0.482 0: BOX: PERSON RULE: SA SCORE: 0.496 BOX: MATERI RULE: SA SCORE: 0.635 01 0: BD (: EMERGE RULE: SCORE: 0.832 0: SELECT 41-51 --42 ENTER BOX NAME -- PROCON HIERARCHY DATA FOR BOX PROCON BD (: PROCON RULE: SA SCORE: 0.482 Q: BOK: PERSON RULE: SA SCORE: 0.496 0: BOK: VERIF RULE: AV SCORE: 0.826 0: OUESTIONNAIRE: 14 SCORE: 1.000 QUESTIONNAIRE: 63 QUESTIONNAIRE: 2 SCORE: 0.337 SCORE: 0.917 QUESTIONNAIRE: 66 SCOPE: 1.000 BOK: CONTRA RULE: AV SCORE: 0.337 0: OUESTIDHNAIRE: 95 SCORE: 0.337 BOX: RESPVI RULE: HA SCORE: 0.832 0: BOX:COMRSP RULE: SCORE: 1.000 BOX:RESP RULE:SA CCORE: 0.832 0: 01 RULE:SA SCORE: 0.635 0: BOX:MATERI BOX: VERIF2 PULE: AV SCORE: 0.917 0: QUESTIONNAIRE: 2 SCORE: 0.917 BOX:CONTR2 RULE:SA SCORE: 0.551 0: QUESTIDNNAIRE: 32 SCDRE: 0.750 QUESTIDNNAIRE: 60 SCDRE: 0.516 BDX:RESPVI RULE:HA SCORE: 0.832 0: BDX:COMPSP RULE: SCORE: 1.000 BDX:RESP RULE:SA SCORE: 0.832 0: 0: SELECT 41-51 -- 42 ENTER BOX NAME -- RESPVI HIERARCHY DATA FOR BOX RESPVI

BOX: RESPVI	RULE:	HA SC	DRE:	0.833	: Q:		
BOX:	COMRSP	RULE		SCORE:	1.000	0:	
BOX:	RESP	RULE	SA	SCORE:	0.832	0:	
	BOX: D	ELRSP	RUL	E:	SCORE:	1.000	0:
	BDX:E	FFRSP	RUL	E:	SCORE:	0.706	0:
	BOX: D	RRSP	RUL	E:	SCORE:	1.000	0:

LELECT 41-51 -- 50 ENTER BDX NAME -- CONACC HIERARCHY INFORMATION FOR BDX CONACC

.

.

*********	********	*********		
		· ADAUTH ·		
	++++3= 0.442++++			
	RULE: HA			

				· VERIF ·
		· PROCON ·	+ PERCON +	
	•		•••••5= 0.496•••	
	•	+RULE: SA+	+ +RULE: SA+	
	•	********	• •••••	
	•		•	
	•		•	• ••••••
	•		•	· · CONTRA ·
	•		•	••••3= 0.337••••
	•		•	· ·RULE: AV.
	•		•	
				· · RESPVI ·
				····5= 0.832*
	1			+RULE: HA+

	1		2	
	1			
	1		· · MATERI ·	a second as the second s
	1		••••]= 0.635••	
				RULE: AV.
	•			
	•		********	
	• • • • •			
	• • • • • •			
	•			 CONTR2 •
	•			••••)= 0.551•
	•			 •RULE: 39•
	•			
	•			
				 RESPVI ·
				••••S= 0.832•
				•RULE: HA•

	EMERGE .			
	++++3= 0.332+			
	•RULE: •			

SELECT 41-51 -- 3

DISTRIBUTION:

1.00

U.S. Nuclear Pegulatory Commission (50 copies for AN) Division of Document Control Distribution Services Eranch 7920 Norfolk Avenue Pethesda, MD 20014 Attn: P. Larkins

U.S. Nuclear Regulatory Commission Division of Safeguards MSSS881 7915 Eastern Avenue Silver Spring, MD 20852 Attn: P. A. Dwyer (5)

Woodward-Clyde Consultants Three Embarcadero Center Suite 700 San Francisco, CA 94111 Attn: A. Sicherman (5)

400	C. Winter
1000	C. A. Fowler
1230	W. L. Stevens, Attn: R. F. Smith, 1233
1700	W. C. Myre
1710	V. E. Blake
1720	C. H. Mauney
1730	J. D. Kennedy
1750	J. E. Stiegler
1760	J. Jacobs
4400	A. W. Snyder
4410	D. J. McCloskey
4412	J. W. Hickman
4413	N. F. Ortiz
4414	C. E. Varnado
4416	L. D. Chapman (5)
4416	K. G. Adams
4416	J. A. Allensworth
4416	H. A. Bennett (7)
4416	D. Engi
4416	L. M. Grady
4416	C. P. Harlan
4416	R. L. Jones
4416	M. T. Olasceaga (7)
4416	C. J. Pavlakos
4416	D. W. Sasser
4416	D. R. Strip
8266	E. A. Aas
3141	T. L. Werner (5)
3151	W. L. Garner (3)
	For: DOE/TIC (Unlimited Release)
3154-3	R. F. Campbell (25)
	For NRC Distribution to NTIS