UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

(4)

REPORT DOCUMENTATION PAGE

1b RESTRICTIVE MARKINGS

DTIC FILE COP

AD-A212 515

3 DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release; distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
LIDS-IR-1905

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Lab. for Inf. and Dec. Systems

6b. OFFICE SYMBOL (If applicable)

7a. NAME OF MONITORING ORGANIZATION
Office of Naval Research

6c. ADDRESS (City, State, and ZIP Code)
Massachusetts Institute of Technology
Room 35-410/LIDS
Cambridge, MA 02139

7b. ADDRESS (City, State, and ZIP Code)
800 N Quincy Street
Arlington, VA 22217-5000

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL (If applicable)

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
N00014-85-K-0782

8c. ADDRESS (City, State, and ZIP Code)

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.
		NR 564-001	

11. TITLE (Include Security Classification)
COMMAND AND CONTROL THEORY

12. PERSONAL AUTHOR(S)
ALEXANDER H. LEVIS AND MICHAEL ATHANS

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical	FROM 7/1/88 TO 7/31/89	August 1989	30

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Progress on three research problems in Command and Control Theory is described

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION
unclassified

22a NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL
CDR T. WELLS	(202) 696-4713	Code 1224

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

LABORATORY FOR INFORMATION AND DECISION SYSTEMS
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Cambridge, Massachusetts 02139

OSP No. 96833 LIDS-IR-1905

PROGRESS REPORT

for the period

1 July 1988 to 31 July 1989

for

COMMAND AND CONTROL THEORY

Contract No. N00014-85-K-0782

Work Unit NR 564-001

Submitted to:

CDR T. Wells
Code 1224
Office of Naval Research
800 N. Quincy Street
Arlington, VA 22217-5000

Submitted by:

Alexander H. Levis
Michael Athans

August 28, 1989

COMMAND AND CONTROL THEORY

1. PROJECT OBJECTIVES

The main goal of this research is to start bridging the gap between mathematical theories of command and control and empirical studies. More specifically, the goal is to develop theories on the one hand and to model experimental paradigms on the other, so that realistic problems in command and control (C^2) can be studied prior to the design of experiments and the collection of relevant data.

The research program undertaken for this project has three main objectives:

(a) The extension of a mathematical theory of C^2 organizations so that it can be used to design an experimental program;

(b) The further development of an analytical methodology for measures of effectiveness, and

(c) The investigation of organizational architectures for distributed battle management (many weapons on many targets resource allocation problems).

The unifying theme of this research is the concept of distributed information processing and decision making. The emphasis is on the development of models and basic analytical tools that would lead to the design of an experimental program as contrasted to ad hoc experimentation.

The project drew upon and contributed to the theoretical developments on Naval Distributed Tactical Decision Making (DTDM) that was being carried out in parallel under ONR Contract No. N00014-84-K-0519. The coexistence of the two programs has made it possible to undertake long-range, basic research on fundamental issues and problems in command and control. The DTDM program was concluded on July 30, 1989.

2. STATEMENT OF WORK

The research program has been organized into five tasks, four that address the research objectives and a fifth that addresses the question of disseminating the results of this project both directly to the members of the Basic Research Group of the Technical Panel on C^3 of the Joint Directors of Laboratories and to the C^3 community at large through publications and presentations.

2.1 Research Tasks

Task 1: Development of Computer-Aided Design System

1.1 Develop the specifications for the Computer-Aided Design System. Specifically, design the data base, the architecture generator, the performance-workload locus module, and the

analysis and evaluation module. The system should be able to handle a generic five member, three echelon organization.

1.2 Implement the design developed in Task 1.1. Design the graphics module to be used in presenting the performance-workload locus and its projections as well as the loci obtained from the analysis and evaluation module.

1.3 Design and implement the user interface. Use the Petri Net formalism for the specification of the interactions between organization members and the design of protocols.

Task 2: Command and Control Organization Design and Evaluation

2.1 Develop and implement a set of tasks, as well as sets of information processing (situation assessment) and decision making (response selection) algorithms for use with the decision maker models. These tasks and algorithms should be appropriate to future experimental efforts.

2.2 Use organizations with up to five members to exercise and test the CAD system developed in Task 1.

2.3 Analyze and evaluate command and control organizational architectures using the CAD system. Begin developing hypotheses that can be tested through experimental efforts.

2.4 Incorporate in the design system and in the analysis module the theoretical results obtained from parallel research projects.

Task 3: C^3 Organizations and Architectures for Distributed Battle Management

3.1 Develop a unified theory for complex engagements of several weapons against several targets. Assume imperfect defensive weapons systems so that the elemental "one-on-one" kill probability is non-unity. Also assume imperfect defensive surveillance so that the target-decoy discrimination probability is non-unity.

3.2 Develop several "many-on-many" engagement strategies and evaluate their impact upon decentralized C^3 systems requirements and architectures. Develop the necessary tools so as to design distributed C^3 architectures compatible with the engagement strategies.

3.3 Illustrate the tactical doctrine and C^2 interface requirements via computer simulations.

Develop hypotheses that could be tested in the field.

Task 4: Measures of Effectiveness

4.1 *Conceptual Development.* Develop and refine the concepts and definitions of measures of effectiveness (MOEs), measures of performance (MOPs), and system/mission parameters. Interpret the concept of measure of force effectiveness (MOFE) as a global effectiveness measure in the context of C^3 systems.

4.2 *Implementation of the Methodology.* Develop a quantitative framework where models of various types can be used to estimate measures of performance (MOPs). Develop analytical, computational and graphical tools for measuring effectiveness (MOEs). Begin the implementation of these techniques on the same workstation used for Task 1 with the objective of developing a system based on MOE evaluation that can be used as an aid in system development and selection. Note that many of the software utilities to be developed are common to Tasks 1 and 4.

4.3 *Implication of the Methodology.* Illustrate the various conceptual and technical developments with examples drawn from actual or planned C^3 systems. Apply the methodology to an evolving C^3 system. While motivated by real systems, the applications will be described in generic terms.

Task 5: Information Dissemination

5.1 Participate in technical session of the Basic Research Group to be held approximately once per calendar quarter.

5.2 Present the research results at technical conferences and meetings and publish articles in archival journals.

3. PROGRESS REPORT

During this period (July 1988 to July 1989) the research effort focused on tasks 2, 3, and 5. A number of subtasks were completed and the results documented in technical papers and thesis reports as stated in Section 5, Documentation; they constitute the complete record of research done.

Highlights of the work done in this period are presented below.

3.1 Development of Computer-Aided Design System (Task 1)

The computer-aided design system, which was named CAESAR for Computer-Aided Evaluation of System Architectures, consists of four major components:

(1) The *Architecture Generator*, which constructs feasible organizational forms using the Petri Net formalism and includes two novel algorithms - the Lattice Algorithm and the DFS Allgorithm - described in earlier reports. (2) The *Analysis and Evaluation Module* which contains algorithms for the analysis of organizational architectures and the computation of Measures of Performance (MOPs). (3) The *Data Base* which is used to store the results of the analysis (the MOPs) or organizational architectures. (4) The *Locus* module which contains routines that construct the Performance-Workload locus of an organizational form or the Performance Locus of a C^3 system that is carrying out a given task, as well as routines that compute and present graphically selected measures of effectiveness (MOEs).

CAESAR incorporates theoretical and computational developments obtained over a period of seven years through sixteen completed theses. Some modules were developed explicitly under this contract; others were developed with support by the Distributed Tactical Decision Making initiative of the Office of Naval Research.

During the late part of 1988, all MS-DOS modules were ported to the Macintosh environment. Since that time, all researchers have been directed to integrate any new programs and algorithms into CAESAR. A presentable version of CAESAR on the Macintosh was completed in January 1989 and since then it has been demonstrated to many visitors at MIT, from government agencies and industry, and to interested persons in other locations such as the Naval Ocean Systems Center. Further development of the MIT prototypes for Petri Net analysis and simulation, MIT/PN and MIT/Sim, was discontinued pending the arrival of the Design/CPN software. This Colored Petri Net system arrived in July for beta testing. The current plan is to maintain the MIT prototypes for our research needs until we ascertain that Design/CPN works properly and reliably. Then we expect to enhance Design/CPN by attaching to it the various modules developed at MIT.

3.2 Organizational Design and Evaluation (Task 2)

Background: The common approach to the design of distributed systems is to interconnect known subsystems or components. However, there is no guarantee that the proposed design will

satisfy the requirements or constraints. A trial and error procedure is used in which the system is tested and modifications are made when requirements are not met. Despite its increasing importance, the assessment of the functionality of a C^3 system is hard to accomplish. Thus, a methodology is needed for modeling distributed systems, developing a compatible representation of the requirements for a system and of its proposed or implemented counterpart, and for comparing the two representations. The problem of developing a compatible representation of the requirements was addressed by the project undertaken by Perdu and Levis and reported in the next section; the following section focuses on the comparison of the two representations and the determination of shortfalls and overlaps using a technique developed by Valraud and Levis.

3.2.1 Requirements Specification using the Cube Tool Methodology

The determination of the functional requirements of a system is usually done by representing the relationships among the different processes which have to take place for the execution of a mission. When the systems are distributed, the requirements must include not only the processes, but also the communications among the different parts of the system. The Cube Tool has been developed at Thomson-CSF in France as a methodology for deriving the processing and communication needs for each system function. In this project, carried out by Mr. Didier Perdu (a visiting scientist from Thomson-CSF and former graduate research assistant) under the supervision of Dr. Alexander H. Levis, the methodology is extended to address the determination of system requirements and their representation in terms of Petri Nets.

The application of Cube Tool to the design and the analysis of a system is done in four steps, as shown in Figure 1.

- Identification of the system Functions and of the different resources (personnel and hardware/software) involved,
- Functional Analysis for the determination of the processing and information exchanges for each function,
- Quantitative Evaluation of Automated Data Processing (ADP) and communication loads in workstations,
- Consideration of different possible architectures through the allocation of the functions to different sites.

The first step consists of defining the system functions from the missions expected to be accomplished. Each function is divided in subfunctions. Simultaneously, the resources needed for the execution of these functions are defined. They consist of personnel and hardware/software entities such as databases or decision aids and are referred to as Actors. In a second stage, a functional analysis is performed for each function in a three dimensional space with axes corresponding to functions, actors and time. In this framework, subfunctions are defined as a collection of activities

with their interrelated information exchanges.

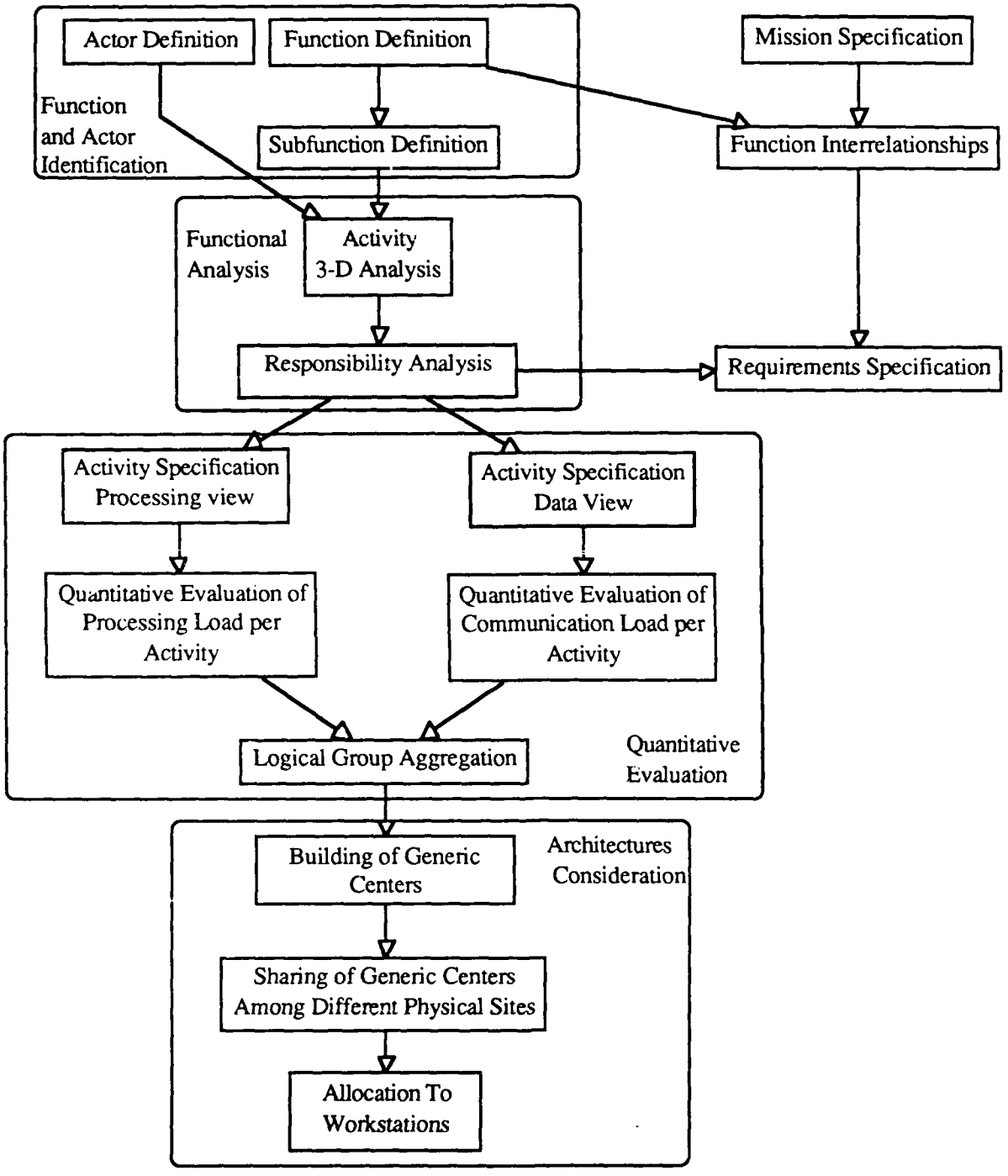

Figure 1 Methodology Flow Chart

The third step is the quantitative evaluation of Automated Data Processing and communica-

tion loads. To evaluate the processing load, each activity of the functional structure is defined using a pseudo-code formalism close to PASCAL or ADA. The number of queries to databases, the kinds of display and the required computations are included by using a set of primitives gathered in a dictionary. To evaluate the communication load, the ways information is displayed and sent are analyzed for the incoming and outgoing data. A processing and communication load is assigned to each of these primitives. The processing and communication load quantification for each activity is made by summing the loads of the primitives used to describe the ecution of this activity. Simultaneously, a quantification is made for the maximum response time to determine the minimum processing power threshold. By summing these estimates of each logical group (which is the set of the activities related to a given system function and performed by a single actor,) the number and type of workstations, the processing requirements, the number of database updates and retrievals and the load associated with processing and related communication flows can be determined.

The last stage is the investigation of different architectures through the allocation of logical groups to different sites. Generic sites are first defined by gathering logical groups meant to operate together and sufficient to constitute an independent node. This is done to check data coherency. Then, the logical groups with their associated loads are assigned onto different sites according to the areas of responsibilities and interests specific to each logical group and to the different modes of operation (normal and backups). Within these new system sites, the load is reallocated to the different workstations according to the type of processing (scientific vs. expert system) and the security requirements. Different architectures can be obtained and the selection is made according to criteria such as cost or ease of implementation.

The focus of the research effort at LIDS was on the first two stages, which are essential for the specification of the detailed requirements of a system.

The first stage of Cube Tool consists of identifying the Functions of the system. At this stage, the designer must find out the user needs, the type of missions the system will have to accomplish, and the personnel and types of hardware and software which will be used. The missions are used as the basis for the identification of the system Functions. For example, a system for planning an air interdiction mission will have as functions the determination of the status of allied forces, weather projection, threat assessment, strike assessment, intelligence report processing, target prioritization and development, weapon system availability, etc.

Then, each system function can be decomposed into tasks or subfunctions. Processing tasks are differentiated from transmission tasks. A function can be considered to be an interleaved sequence of processing and communication tasks, while a subfunction can be defined as a single pair of a processing task and a communication task. The execution of a function will require the sequential execution of its subfunctions.

The initial specification of system elements, activities, and information exchanges is done through functional analysis in the three dimensions of the Cube Tool, as shown on Figure 2. The three axes of interest are :

- Functions: These are the processes which have to be executed for the fulfillment of the mission.
- Actors or Hierarchical Levels: These are the personnel and the hardware and software nodes responsible for executing the different tasks. Personnel are layered in hierarchical levels and are most of the time specialized per functional domain
- Time: This axis allows to define on the same time scale the execution time of the functions, their frequency and their sequence.

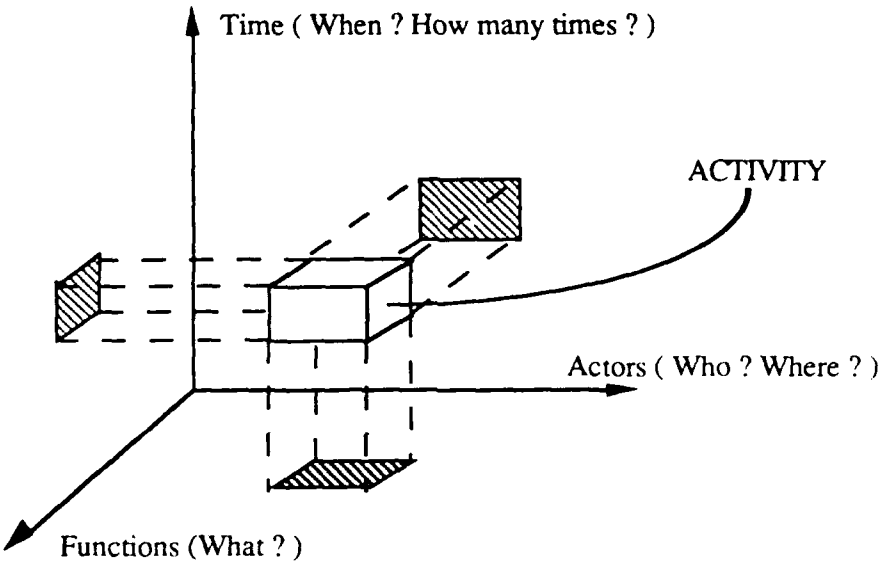

Figure 2 Three Dimensional Functional Analysis

In this analytical framework, a subfunction is composed of activities. An activity is defined as a process which supports a given system function and which is performed by a single actor or hierarchical level without major interruption. Therefore, activities can be part of a processing task, a communication task, or contain elements of both. Activities are differentiated according to the type of processing they represent and which are called roles. The roles considered by the method are:

- *Elaborate* (**E**): transform or generate information.
- *Acknowledge* (**A**): receive an order(from an E role) important enough to warrant the generation of an acknowledgement.
- *Check* (**C**): receive a report in response to an order previously generated.
- *Warn* (**W**): receive an information which does not require taking any measures in the current mode of operation.
- *Monitor* (**M**): receive an information on system operation allowing to accomplish com-

mand control and communication resources management.

- *Monitor Locally* (**L**): same as M but on a local basis
- *Secure* (**H**): exchange of secured data such as encryption keys, access keys and certification mechanisms of users trustworthiness.

These activities can be looked at from three different perspectives represented by the analysis planes defined by the three axes, as shown in Figure 3. These are :

- Responsibilities Plane (Functions / Actors): This plane shows which actor is in charge of a set of specific activities.
- Sequences Plane (Functions / Time): This plane shows when (and how many times) an activity will be executed.
- Actions Plane (Time / Actors): The plane of actions shows when actors are busy performing some activity.

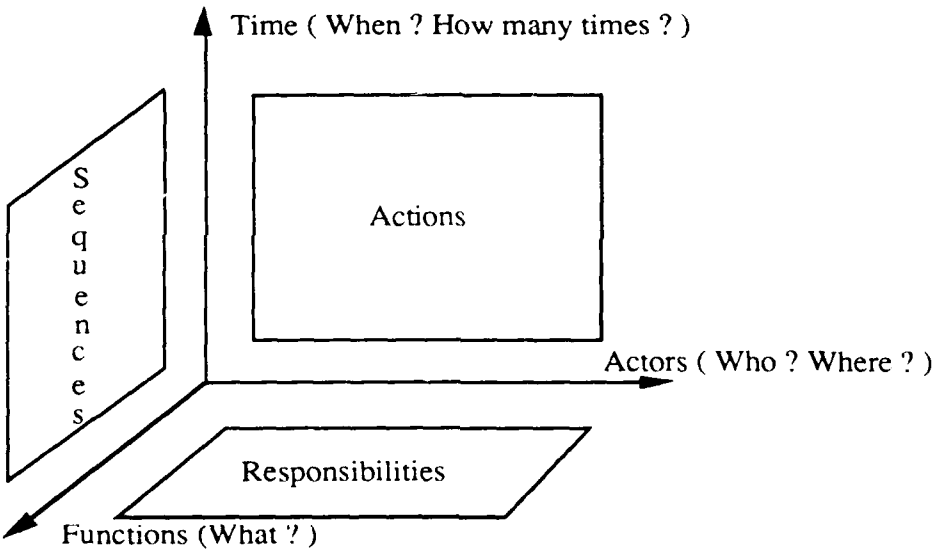

Figure 3 The three Analysis planes

The main analysis is performed in the responsibility plane. The roles which are used most and are the only ones considered for the requirements specification are E, A, C and W. The responsibility plane is constructed by allocating the roles for each subfunction to the different actors. This allocation must verify the following rules:

- There is one and only one role E per subfunction.
- Except for the first subfunction which starts the execution of a function, a role E can only be triggered by a role A or C.
- The presence of a role A requires the presence of a role C in the column of the actor

which has generated the order. More generally the exchange which take place from a higher hierarchical level to a lower one is done by the presence of roles A, W. Exchanges which take place from a lower hierarchical level to a higher one are done by the presence of role C. The couples E-A, E-W and E-C correspond to exchange of information from the actor performing the role E to the actor performing the other role (A, W or C).

This is illustrated in the example shown on Table 1.

Table 1: Responsibilities for a Function with six subfunctions performed by four actors

	actor 1	actor 2	actor 3	actor 4
subfunction 1	E	A	W	
subfunction 2	C	E	A	W
subfunction 3		C	E	A
subfunction 4			C	E
subfunction 5		C	E	
subfunction 6	C	E	W	

Explicit exchanges take place across columns, between activities contributing to the execution of the same subfunction (i.e., on same row). Implicit exchanges occur from row to row between activities performed by a single actor. The interesting aspect of this methodology is that several configurations, differing as to the resources used or reflecting variations in operational needs, can be represented in a consistent manner. This allows to define different thresholds of responsibilities in different modes (normal mode or emergency modes) and to point out how the reallocation of the tasks has to be made among the available actors when the system switches from one mode to another.

Next, the allocation of roles is converted into Petri Nets and the detailed requirements of a system for a particular mission are generated.

The requirements of a system are the set of processes which have to take place for the correct execution of a mission. These requirements are scenario-dependent and are most often defined by the set of functions with their sequences and interrelationships. Operational Sequence Diagrams or Structured Analysis (SADT) diagrams can be used as the front end for the Responsibility Plane Analysis. The Cube Tool can be used to define, for each function, the processes and the communication exchanges among the different actors involved in the execution of that function. The Petri Nets depict graphically these processes and communication exchanges for each function. When these representations are linked together to construct the requirements, a global and consistent graphical representation can be defined that lets the designer or the analyst take advantage of the mathematical framework which underlies Petri Nets.

The first two steps of Cube Tool result in the definition of the different system functions,

their subfunctions, and how the activities constituting these subfunctions are allocated to the different actors of the system. For each system function, the responsibility analysis plane defines the activities performed by the different actors. From this representation, the generation of the equivalent Petri Nets representation of the responsibilities for each function is done in three steps.

In the first step, each activity is depicted by a transition. The transitions representing the activities performed by the same actor are aligned horizontally, while the ones representing the activities belonging to the same subfunction are aligned vertically. In other words, the transpose of the array of responsibilities is obtained and the non-null elements of this array are transformed into transitions, as shown in the Figure 4.

A label is attached to each transition identifying (1) the function, (2) the subfunction to which the represented activity belongs, (3) the type of activity (E, A, C or W) and (4) the actor performing this activity. For example, in Figure 4, the label 1.3E3 means that the activity represented by the transition belongs to subfunction 3 of function 1, is of type E, and is performed by actor 3. In the application described in this paper, the subfunctions are not identified by their order of appearance in a function, but by the identification number of the processing they represent throughout the system.

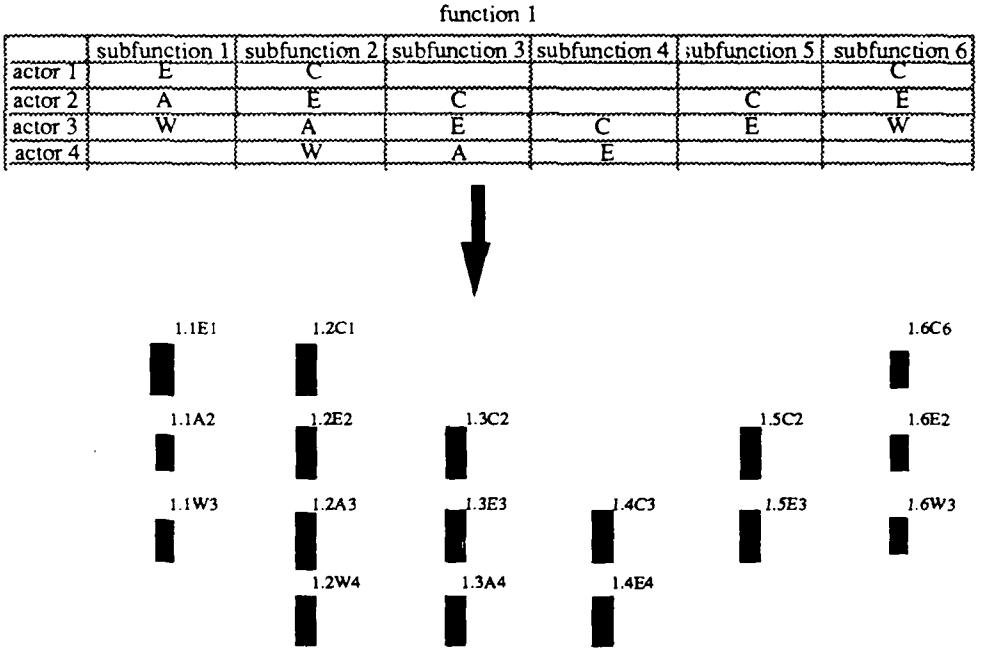

Figure 4 Drawing the transitions grid

The second step is to add places between the transitions representing the activities performed by a single actor and to connect them. In this way, the implicit information exchanges which take place between the successive activities performed by each actor are modeled. Figure 5 shows the net ob-

tained for the example.

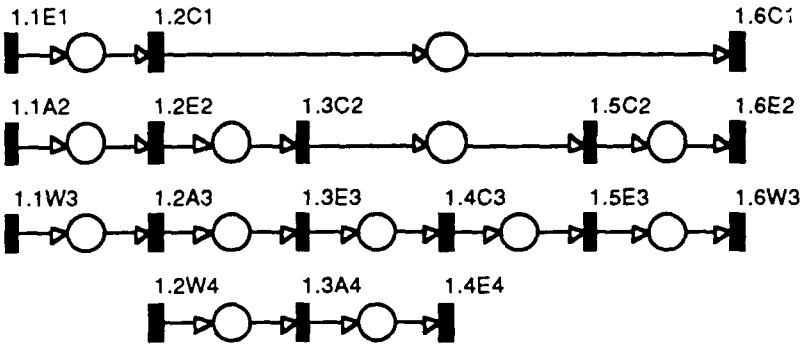

Figure 5 Adding Implicit Information Exchanges

The third step consists of adding the information exchanges which take place among the actors for each subfunction. Let us recall that in the Cube Tool methodology, an exchange originates from a role E and ends at a role A, W or C and that there is one and only one role E for each subfunction. Therefore, for each column of the Petri Net representation obtained after the two first steps, the transition representing the role E is identified and is connected to the other transitions of the columns with a connector-place-connector set. Figure 6 shows the net obtained by adding these explicit information exchanges.

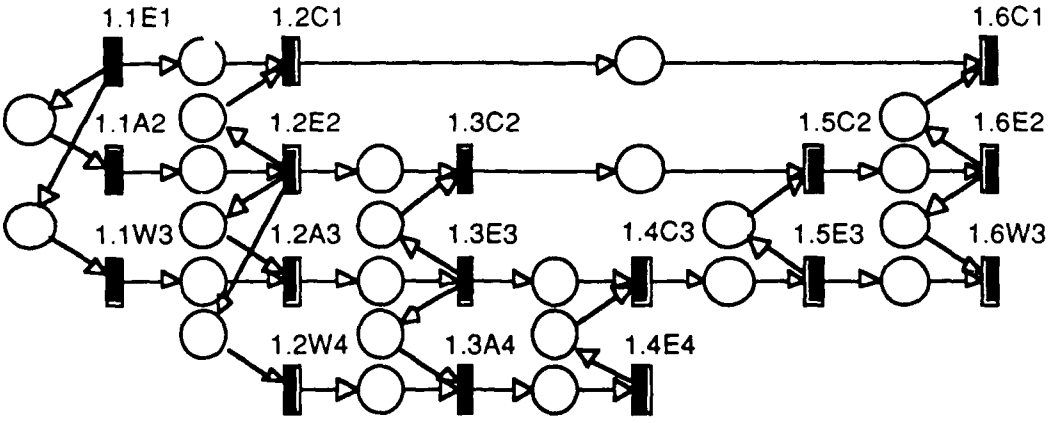

Figure 6 Adding Explicit Information Exchanges

The procedure for modeling the detailed requirements for a given scenario is shown on Fig. 7. The definition of a scenario, that is a mission to be carried out, leads to the specification of the relationships and sequences of system functions. For the fulfillment of a mission, one can identify the system functions which can be executed concurrently as well as the functions which will have to be executed first to trigger the execution of a sequence of functions. These interrelationships among functions vary from one scenario to another. Petri Nets are used to represent the sequencing and

concurrency of functions so that the global requirements of a system can be derived. The procedure for determining the detailed requirements starts with the definition of the responsibilities for the chosen scenario. To list the functions on the Functions axis, the slices (Hillion,1986) of the Petri Nets representing the global requirements are computed. These slices represent the functions which can be executed concurrently. The functions are listed on this axis in the order of appearance in the slices list. Then, for each function, the actor which triggers the execution and gets the final report is identified. This actor is designated as the main one responsible for the execution of this functions. Once the main actors are listed on the Actors axis, the responsibility plane for the scenario can be constructed. For each function:

- A role E is placed on the cell defined by the function and by the main actor.
- Roles W are placed on the cells defined by the functions and by the main actors who are responsible for the execution of the subsequent functions as determined by the Petri Nets of the global requirements.

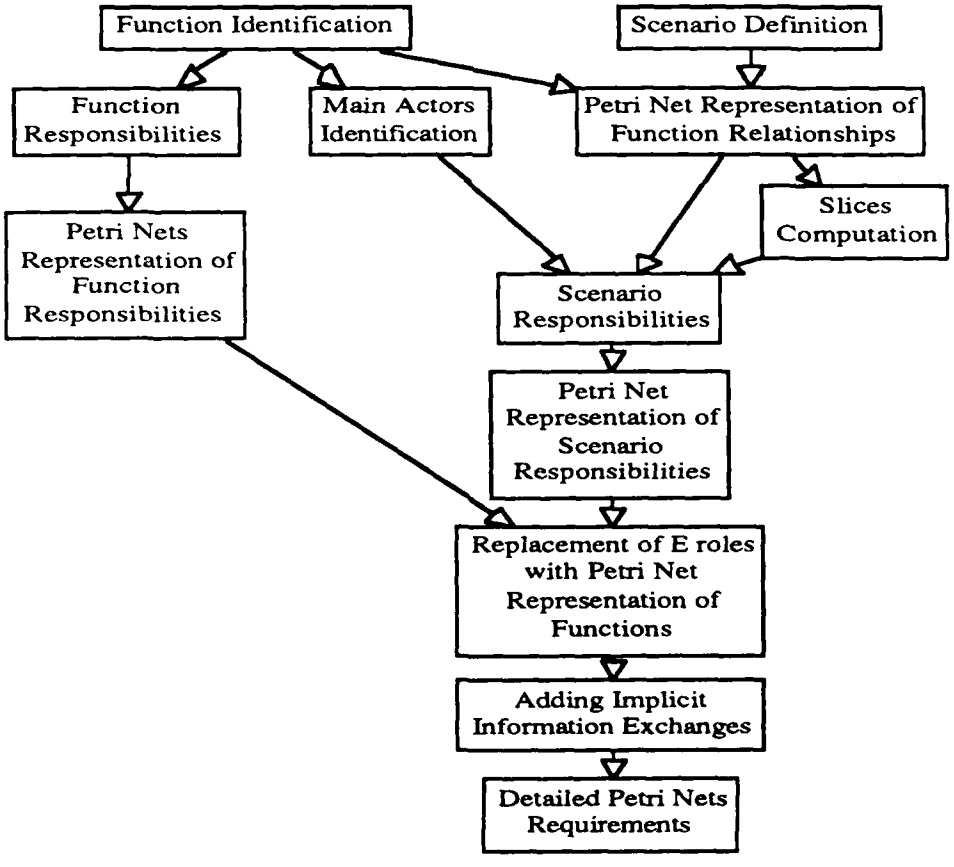

Figure 7 Procedures to Model the Detailed Requirements of a System

Let us consider an example where there are three functions: f1, f2 and f3, and three actors (A1,

A2 and A3). The scenario specification has determined that f1 and f2 have to be executed before f3. The Petri Net is shown on Figure 8 and the slices are:

Slice 1: f1, f2

Slice 2: f3

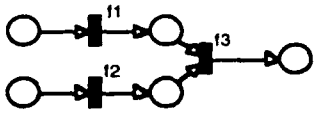

Figure 8 An example of Global Requirements represented with Petri Nets

The responsibilities specification of each function show that A1 is the main actor for f1, A2 for f2, and A3 for f3. The scenario responsibility plane is constructed by placing a role E in the cells (f1, A1), (f2, A2) and (f3, A3) and a role W in the cells (f1, A3) and (f2, A3) (Table 2).

Table 2 Example of Scenario Requirements

	A1	A2	A3
f1	E		W
f2		E	W
f3			E

From the information in the scenario responsibilities plane, the equivalent Petri Net can be constructed following the same procedure that was used for the functions. The next step is to replace each transition representing a function with the equivalent representation of the responsibilities of this functions. By adding the implicit exchanges among actions for each actor, the Petri Net of the detailed requirements is constructed.

In this recently completed project, Cube Tool has been extended from functions to systems. and a methodology for deriving structural requirements has been proposed. It is used to represent with the Petri Net formalism the processes and communications which take place for the correct execution of a mission. This methodology fills a gap between the description of requirements and the quantitative models needed for the analysis and evaluation of C^3I systems designs.

Documentation:

[1] D. M. Perdu, "Requirements Specification using the Cube Tool Methodology", Report LIDS-R-1899, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, August 1989.

[2] D. M. Perdu and A. H. Levis, "Requirements Specification using the Cube Tool Methodology", *Proc. 1989 Symposium on C2 Research,* National Defense University, Ft. McNair, Washington, DC, June 1989.

[3] D. M. Perdu and A. H. Levis, "Requirements Specification in Distributed Intelligence Systems using the Cube Tool Methodology", *Proc. 4th IEEE Int'l Symposium on Intelligent Control*, Albany, NY, September 1989.

3.2.2 On the Quantitative Evaluation of Functionality in C^3 Systems

Progress: This problem was addressed by Mr. François Valraud under the supervision of Dr. Alexander H. Levis. The mathematical framework used to model command and control was based on Petri Net theory. To embed different strategies (options and choices) switches were used. A switch is a transition that, when it fires, generates a token at only one of its output places in contrast to an ordinary transition that creates tokens at all its output places. The choice is made according to a rule associated with the switch. In Figure 9, a first switch, s1, permits to choose between two distinct courses of actions (COAs). When v equals 0, then a second set of alternative courses of action is offered. According to the value of w, outputs will be produced either at p8 or p9.

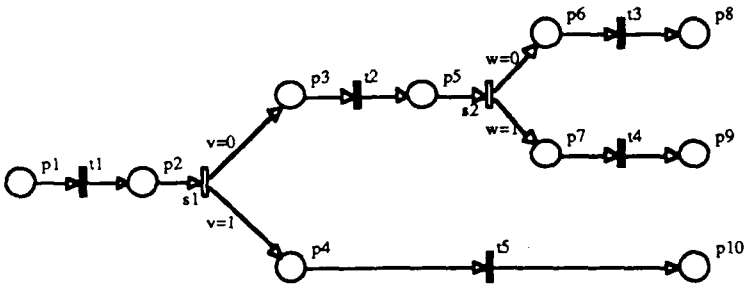

Figure 9 Petri Net PN1 with two Switches

Note that switches are unnecessary, if High Level Petri Nets such as Colored Petri Nets or Predicate Transition Nets are used. However, at the beginning of the project, the Design/CPN software was not yet available; it became available only after completion of thius task. Thus, switches were used as one minor extension of Ordinary Petri Nets.

Given these modeling considerations, it became possible to express the functional requirements in the form of a Petri Net. It should be stressed that the Petri Net representing the functional requirements does not embody any specifications about system performance nor does it specify the degree to which functions are distributed nor does it specify the degree of redundancy in the system, if any. Therefore, these requirements may be viewed as the minimal functional requirements, in the sense that they specify the desirable interrelationship between functions so that the mission can be accomplished.

The relation between functions can be effected directly through the transmission of a signal from one function to another. It can also be done through the use of a database: a function sends the result of its processing to a database which stores the information while another function ac-

cesses that information through a query to the same database.

Resources are modeled with loops in a Petri Net. Each physical resource is represented in a Petri Net by a place that describes the availability of that resource. A resource loop from one resource place to a particular agent performing a certain role indicates the need for that resource. To avoid self-loops, these loops contain more than just one place and one transition. The marking in the resource loop indicates whether the resource is available or not.

Using such constructs, an ordinary Petri Net augmented with switches is developed that represents the proposed command center design or, in some cases, the prototype system. This general representation is then refined by considering a specific scenario. Then, the unnecessary functional paths for that scenario are removed. To do that, we consider all input messages that are not relevant for the chosen scenario and eliminate them. This alleviates part of the complexity of the representation. The elimination of the unused nodes can be cumbersome because some of the nodes that are part of an unused simple path may still be crucial for the completeness of the representation of the net.

In a Petri Net model of a C^3 system, a *simple information flow path* is any directed path from a source to a sink. A simple information flow path represents a token movement from a source to a sink and corresponds to a line in Petri Net theory. A *simple functionality* is an ordered sequence of processes that operates on an input message to produce an output. Note that other operations on other inputs may be necessary to produce the output. Thus, a simple functionality tracks the processing of a single output, but does not provide all the necessary processes that are needed to produce an output. Nor does it trace the only possible sequence of processes starting from a given input.

A *complete information flow path* consists of all the simple information flow paths that end at the same sink. A more precise definition is as follows: A complete information flow path contains all the simple information flow paths that are involved in producing a specific output at some sink; it may also contain circuits. A complete information flow path is a sub-net with a single sink. Therefore, in a Petri Net, there are as many complete information flow paths as the number of sinks. A *complete functionality* consists of the complete set of coordinated processes that operate on all the necessary inputs to produce an output.

It follows directly from these definitions that a simple information flow path belongs to one and only one complete information flow path. Furthermore, there is a direct correspondence between the structural properties of a system, as described by a Petri Net, and the functions that the system performs. A simple information flow path, a structural element of a Petri Net, corresponds to a simple functionality, while a complete information flow path corresponds to a complete functionality.

In order to identify the complete information flow paths and the elementary information flow paths that constitute each complete information flow path, the minimal support S–Invariants are used. The methodology for obtaining the complete information flow paths consists of three steps:

1) compute the minimal support S-Invariants. This step can be realized by applying the algo-

rithm developed by Alaiwan and Toudic.

2) for each minimal support S-Invariant, construct its corresponding sub-net.
3) By applying a set of rules, coalesce some of these sub-nets to obtain a complete information flow paths.

A detailed presentation of this algorithm can be found in Valraud's MS Thesis that has been published as a LIDS technical report.

Once the complete information flow paths have been determined, the next problem is to compute the simple information flow paths that they contain. To do that, an enhanced version of the algorithm developed by Jin et al. at MIT is used.

The technique was applied to a hypothetical command center for planning air interdiction missions. To evaluate the functionality of the command center, a pair of ordinary Petri Nets was selected for comparison, one from the requirements net, the other from the system net.

Shortfalls: The most extreme case is that of the absence of a complete functionality from the proposed or implemented system net which is present in the requirements net. That corresponds to the absence of a sink in the system net. Formally, given the correspondence between complete functionality and complete information flow path, this type of shortfall is defined as follows:

A *complete shortfall* is observed if a complete functionality in the requirements net has no counterpart in the system net, i.e., if a complete information flow path in the requirements net has no corresponding sub-net in the system net. The most obvious test is to compare the sink nodes of the two nets.

A *partial shortfall* is observed when the complete information flow path in the system net that corresponds to a required complete functionality does not contain all the simple information flow paths corresponding to all the embedded simple functionality in the required complete functionality.

Partial shortfall addresses the issue of diminished functionality existing in the system, e.g., an output is produced either with reduced processing (some processing steps are missing), or with reduced inputs (some data are either not available or not used). The latter case is easier to check because it relates directly to the sources and the simple information flow paths.

To detect partial shortfalls, the simple information flow paths of corresponding complete functionalities in the requirements net and in the command center net have to be compared. Such a comparison requires some knowledge about the sources and functions in both nets and, more specifically, the establishment of correspondence between sources and functions embedded in both Petri Nets.

Rule 1: If a simple functionality, represented by a simple information flow path in the requirement net, has no equivalent in the system net, then there is a partial shortfall.

Rule 2: If the sources of a simple functionality specified in the requirements net are not contained in the sources of the simple information flow paths of the system net that have an equivalent sequence of processes, then there is no functional equivalence.

Overlaps. In the context of Command and Control, an overlap means that a certain sequence of operations on a given input can occur in more than one way in the net. For example, signals from a source can go to different workstations and be processed in parallel to produce the same output signal. If this capability is used in the sense of options or alternatives, i.e., only one of the paths is used for each individual message that is processed, then this type of redundancy is beneficial because it increases the reliability of the functions. However, explicit protocols must be in place that determine without any ambiguity the selection of the particular simple path to process the signal. If the protocols are not well designed, a conflict may occur which results in confusion. Consider now the case where a given input signal is processed in parallel by several paths. If only one of the outputs is used and the results of the other paths are ignored, then this parallel processing represents a waste of processing resources. The question is then to evaluate if this waste is prejudicial to the proper functioning of the system. If the workstations carrying the parallel processes could have been carrying other functions instead, thus reducing the time of the whole task, then clearly the answer is yes. If, on the other hand, the workstation would have been idle, then the answer is no.

Redundancy with Conflict: An example is shown in Figure 10. The source is p1, the sink is p3, but there are two alternative simple information flow paths leading from p1 to p3. Let us assume that the two paths are equivalent, i.e., the two processes represented by t3 and t4 produce the same result as the processes t1 and t2, even though the intermediate signals in p2 and p4 may be different. This Petri Net represents an overlap as the two simple functionalities are equivalent. However, this is not a desirable redundancy because it creates conflict; there is no rule associated with p1 to determine which of the two transitions, t1 or t3, will fire or execute when a token appears in p1. This kind of redundancy is detrimental to the efficient operation of the system and reflects a weakness in the concept of operations.

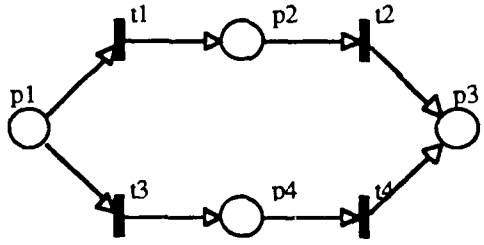

Figure 10 Redundancy with Conflict

The solution to this problem is to implement a switch, s1. This is shown in Figure 11. With the switch, only one of the places p5 and p6 will receive the signal. The choice will be made according to the rule that is embedded in the switch. Confusion is avoided by forcing the clarification of the concept of operations. This type of structural change in the description of the system net is to be made prior to the final identification of pairs of Petri Nets to be compared, since the introduction of a switch creates new strategies.

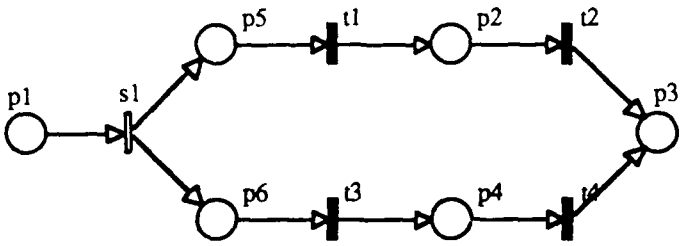

Figure 11 Redundancy with Resolved Conflict

Redundancy with Concurrency: As depicted in Figure 12, the incoming signal goes to both simple information flow paths and the two outputs arrive at the sink independently. While more resources are used, both speed of response and higher reliability may be achieved because the sink can receive either one of the processed signals and the rest of the system can proceed as soon as the first one arrives.

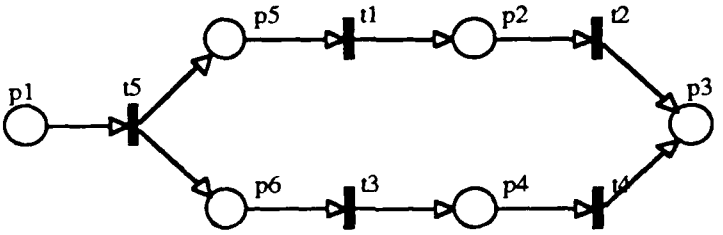

Figure 12 Redundancy with Concurrency

Redundancy with Synchronization: In contrast to the redundancy with concurrency, both simple information flow paths process the incoming signal, but the sink can accept only a fused result. This need for fusion is modeled by transition t6 in Figure 13. Both simple information flow paths must complete their execution in order for t6 to be enabled, so that it can fire and produce the output. Clearly, in this case, the response time will be the maximum of the response times of the two processing paths, while in the case of redundancy with concurrency, the response time is the minimum of the two processing times. On the other hand, the quality of the output may be enhanced, provided the fusion algorithm is adequate. A trade-off is identified between increased accuracy and shorter processing time.

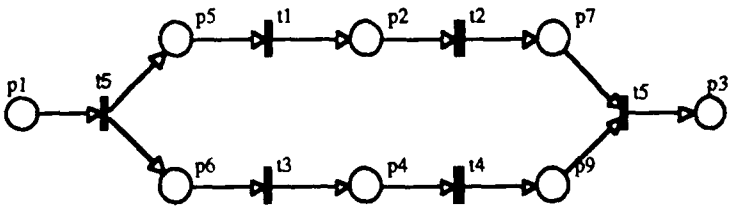

Figure 13 Redundancy with Synchronization

When the comparison of pairs of ordinary nets is completed, it is then possible to evaluate the set of alternative structures in order to assess how well the system meets its requirements. The goal of this evaluation is to identify functional problems in the design of the system.

It is clear that a particular ordinary Petri Net in the command center net which exhibits either a complete or a partial shortfall does not operate according to the requirements. Therefore, this mode of operation of the command center should be eliminated; it should be designed out. One case of interest is when all the ordinary Petri Nets corresponding to a particular strategy contain the same partial shortfall. In this case, it is possible that this strategy cannot be implemented satisfactorily. A way has to be found to implement the missing simple functionality within the existing structure of the net. This is sometimes possible, if all the functions contained in the missing simple functionality have corresponding functions in the ordinary Petri Net of the system.

The lack of some inputs may not be critical. The operation of that simple functionality and its contribution to the mission of the system has to be evaluated in greater detail so as to decide whether or not there is an operational problem. It may happen that some data considered critical for a simple functionality in the requirements are in fact not essential for the fulfillment of the mission.

The issue of redundancy is more sensitive. Clearly, a redundancy with conflict is detrimental to the functional effectiveness of the system. Further, the uncertainty that results is also a factor increasing the chance of deadlock. A redundancy with concurrency is a good thing as it increases survivability and reliability. A redundancy with synchronization is somewhat dangerous in the sense that reliability decreases compared to an equivalent processing structure with no redundancy. However, the fusion algorithm associated with such a redundancy may lead to improvement in the quality of the output. Therefore, there is a trade-off to be evaluated on a case by case basis. However, the need to be able to detect such a redundancy, so that an evaluation of the trade-off can be performed, is not in question.

Documentation:

[1] Valraud, F., "Evaluation of Functionality in Distributed Systems," MS Thesis, Technology and Policy Program, also Report LIDS-R-1868, Laboratory for Information and Decision Systems, Cambridge MA, 1989.

[2] F. Valraud and A. H. Levis, "On the Quantitative Evaluation of Functionality in C3 Systems", *Proc. 1989 Symposium on C2 Research,* National Defense University, Ft. McNair, Washington, DC, June 1989.

[3] F. Valraud and A. H. Levis, "On the Quantitative Evaluation of Functionality in Distributed Intelligence Systems", *Proc. 4th IEEE Int'l Symposium on Intelligent Control,* Albany, NY, September 1989.

3.3 C3 Organizations and Architectures for Distributed Battle Management.

Project Objective: The long-range goal of this research is to understand basic issues associated with Battle Management/C3 (BM/C3) architectures associated with many weapons engaging several targets. Thus, we are concerned with studies of static and dynamic versions of strategies most often referred to as *Weighted Subtractive Defense* and *Adaptive Preferential Defense*. The defensive weapons are assumed imperfect, and the offensive targets may have a finite probability of being decoys. The offensive weapons are aimed against different type of assets of different value to the defense; if an asset is not defended, then there is a non-unity probability that the offensive weapon will destroy it. Thus, the problem is one of wise Weapon-to-Target (WTA) assignment strategies, and their interface with other BM/C3 functions. We also want to understand stochastic dynamic strategies, such as shoot-look-shoot-... in a many-on-many context. Finally, we seek the evaluation of centralized, decentralized, and distributed BM/C3 architectures that support such (static or dynamic) "many-on-many" engagements, including issues related to the replication/redundancy of command-and-control centers and the impact that the redundancy of the BM/C3 function has upon both offensive and defensive strategies. Very few basic research papers have addressed such BM/C3 and WTA problems in the unclassified/unrestricted literature. Out goal has been to fill this gap, in view of the generic importance of this problem to all different types of military planning and tactics.

Progress in Problem Definition: Several mathematical formulations of the problem are possible. First, let us consider a static version of what we call a "Target-Based WTA problem" which corresponds to weighted subtractive defense. Suppose that we have a total of M defensive weapons which we are willing to commit against a total of N offensive targets. At the most general level, the effectiveness of each weapon can be different against each target; this can be quantified by having a different kill probability p_{ij} for weapon j assigned against target i (j = 1, 2, ..., M; i = 1, 2, ..., N). The WTA function should allocate the right weapons against the correct targets so as to minimize some cost function.

The simplest cost function is leakage, i.e. the expected number of surviving targets (corresponding to pure subtractive defense). Thus, if we adopt an optimization framework we wish to minimize the leakage L which is given by

$$L = \sum_{i=1}^{N} \prod_{j=1}^{M} (1-x_{ij}p_{ij}) \qquad (1)$$

by selecting optimally the M.N allocation decision variables x_{ij} , each of which is either 0 or 1. Thus, $x_{ij} = 1$ if the j-th weapon is assigned to the i-th target and 0 otherwise and

$$\sum_{i=1}^{N} x_{ij} = 1 \quad , \quad j = 1, 2, ..., M \tag{2}$$

which simply states that each weapon can only engage a single target.

The solution of such optimization problems for the WTA function is very difficult, because it has a strong combinatorial flavor; in fact, it has been proven to be NP-complete by Lloyd and Witsenhausen in 1986. Part of the complexity relates to the fact that the kill probabilities p_{ij} are different; also, the problem is nonlinear because it may be optimal to engage a particular target with two or more defensive weapons (salvo attack in a shoot-shoot-... context). If the kill probabilities are the same, i.e. $p_{ij} = p$ for all i and j, then the optimal solution (to minimize the leakage) is easy and it requires the maximally uniform assignment of the weapons among the targets. The problem is inherently hard even in the special case that the kill probabilities depend only on the weapons but not the targets, i.e. p_{ij} is independent of i.

More realistic versions of this problem can be formulated in a similar manner. For example, each target indexed by i = 1, 2, ..., N can be assigned a value V_i reflecting the importance of that specific target to the defense. In this case, the strategy is referred to as *weighted subtractive defense*. Under this assumption, the defense may wish to minimize the expected total surviving value associated with all targets, i.e. minimize the cost function

$$C = \sum_{i=1}^{M} V_i \prod_{j=1}^{N} (1 - p_{ij} x_{ij}) \tag{3}$$

again by selecting optimally the M.N allocation decision variables x_{ij} , subject to the constraints of eq. (2). Once more, salvo tactics may be optimal. Also, it may be optimal not to engage low value targets.

Another, still more complicated, setting couples the WTA problem to that of *preferential defense*. In this framework we explicitly take into account the value of the defense assets. So let us suppose that the defense wishes to protect a total of Q assets, indexed by q = 1, 2, ..., Q, and that each asset has a value denoted by D_q. Each one of the defense assets can be attacked by one or more enemy targets. Let π_{qi} denote the probability that the i-th target can kill the q-th asset. Note that the π_{qi} captures such important attributes as target yield, asset hardness, targeting accuracy (including the accuracy of impact point prediction in a tactical setting) etc. In this case, we can form a utility function which the defense wishes to maximize. This utility function takes the form:

$$U = \sum_{q=1}^{Q} D_q \prod_{i=1}^{N} (1 - \pi_{qi} \prod_{j=1}^{M} (1-p_{ij}x_{ij})) \qquad (4)$$

The above formulation allows for optimal selective defense of the defensive assets. It may be worthwhile to leave a low-value over-targeted defense asset undefended in order to direct the defensive weapons against other targets. If many offensive weapons attack a particular asset then if it is optimal to defend the asset, then the defense must engage (possibly with salvos) every target aimed toward that asset. On the other hand, if it is optimal to leave a particular asset undefended then all attacking targets should not be engaged. These characteristics make the solution of the optimal preferential defense problem far more difficult as compared to the corresponding one of subtractive defense.

The above formulations correspond to the static version of these problems, i.e. when the defensive weapons allocation is executed over a single time stage. Dynamic versions of these problems, leading to shoot-look-shoot ... (SLS) versions of these problems are possible. Because of the kill assessment information, the defense effectiveness increases for the same stockpiles and kill effectiveness. However, the algorithms to carry out these optimal dynamic WTA strategies are extraordinarily complicated (NP-hard), and computation of truly optimal dynamic solutions is restricted to a small class of problems. The challenge is to develop near-optimal algorithms, with reasonable computational requirements, which can provide us with both a qualitative and a quantitative feel for the engagement strategies and the performance of these most complex stochastic dynamic engagements.

Progress in Solution Methodologies. This research is being carried out by two doctoral students, Mr. J. Walton and Mr. P. Hosein, under the supervision of Prof. M. Athans. Both students are working on research which will constitute their Ph. D. thesis topics in this area.

Mr. P. Hosein and Professor M. Athans have studied both the static and dynamic versions of the subtractive and preferential defense problems defined above, under the assumption that the C2 function is perfect (i.e. not vulnerable to enemy attack). We have been successful in deriving several analytical results, algorithms for near-optimal solutions, upper and lower bounds for performance, and have carried out several numerical studies for understanding the subtleties and properties of the solutions to these problems. We have proven certain somewhat counter-intuitive results. For example, suppose that all targets have the same value and that the kill probabilities are the same. We have examined multistage versions of the shoot-look-shoot (SLS) type of strategy. In the two stage version of the problem we have shown that the optimal allocation of M weapons against 2 targets is to attack at each stage in a uniform manner with (the integer nearest to) M/4 interceptors. Thus each target is attacked initially by M/4 weapons; if both targets survive then at the second stage each is attacked by M/4 weapons; if only one survives then it obviously gets attacked

with M/2 interceptors (our optimization problem formulation does not try to "save" interceptors). What appears -- at least to us -- to be counter-intuitive is that more than one interceptor is committed during the first stage. However, it turns out that this is the optimal thing to do. A similar uniform attack strategy is true for three or more stages. In fact we have derived a recursive formula that can be used to evaluate the benefits of multi-stage SLS strategies as compared to "blind" salvo attacks. We plan to examine these problems in a different setting, e.g. when the kill probability changes as a function of the stage that the target is shot at.

In general, multistage SLS strategies can be very beneficial from the viewpoint of improving the overall defensive effectiveness. Obviously, multistage strategies require expanded battlespace in addition to the sensors that will provide kill assessment information. We have proven (for small engagements) that having enough battlespace so that we can implement multistage SLS strategies can improve by 100% the defensive effectiveness in reducing target leakage, as compared to single stage (static) salvo attacks. Similar conclusions appear to be true (on the basis of specific numerical studies) for more complex scenarios. The price that one has to pay for such a performance improvement is more complex WTA algorithms. Thus, the sophistication of the near-optimal dynamic WTA algorithms is equivalent to improving the kill probability of a defensive weapon!

We also have been successful in deriving new near-optimal algorithms for the dynamic version of the asset defense problem (i.e. preferential defense). Unfortunately, the problem is so complex so that it is impossible to derive an optimal algorithm even for the simplest academic cases. However, we have demonstrated via numerical studies that significant performance payoffs are possible when one uses the asset-based WTA algorithms as compared to the (simpler) target-based ones. We have obtained significant insight into the nature of both static and dynamic preferential defense strategies and their sensitivities to parameter variations. Once more the use of dynamic SLS strategies results in significant performance improvements.

Mr. J. Walton and Professor M. Athans have been studying the impact of vulnerable C2 nodes in the context of preferential asset defense; we study the tradeoffs between the distribution and level of redundancy of the C2 function and the change in strategies associated with the WTA algorithms. In particular, we were unable to find in the literature a systematic study which points out the quantitative changes in both offensive and defensive strategies when redundant, vulnerable C2 nodes are used. To be sure, one can treat C2 nodes as a special type of asset and have an arbitrary (albeit reasonable) value assigned to it; and then use the preferential defense strategies studied above. We felt, however, that this approach was arbitrary and not natural; also, we wanted to study a two-sided version of the problem, i.e. optimizing both the offensive and the defensive strategies and studying potential shifts in these strategies as a function of key parameters (weapon stockpiles, kill probabilities, etc).

First, we assumed that the defense replicates R command-and-control (C2) nodes. An initial assumption is that any surviving C2 node can launch and guide all defensive weapons to their correct offensive targets. The offense has a choice on whether it is going to attack any of the C2 nodes or not; any offensive weapons used against the C2 nodes cannot be obviously used to attack the

other assets. If all C2 nodes are destroyed, then the defensive weapons are useless. If all replicated C2 nodes are the same (i.e. equally vulnerable to the offensive weapons) then it is optimal for the offense to attack all C2 nodes uniformly if at all, while the defense need defend well one and only one C2 node. Thus, as a function of the kill probabilities and stockpiles involved one can expect (and we discovered) fascinating shifts in both offensive and defensive strategies.

In the special case of perfect kill probabilities (i.e. an unintercepted target will surely destroy a C2 node and a launched interceptor will certainly kill its target) the defense should use preferential defense and defend perfectly only one of the C2 nodes. Thus, it is suboptimal for the offense to target any C2 nodes under these assumptions. This strategy changes when the targets and weapons have non-unity kill probabilities; in that case, it may be optimal for the offense to attack (uniformly) all the C2 nodes, while the defense still has to defend only one. The level of the allocated offensive and defensive forces changes as one changes the size of the stockpiles and the kill probabilities, including parameters that define the relative hardness of the C2 nodes vs the remainder of the assets.

A much harder problem arises when we remove the restriction that the C2 nodes are perfectly replicated. We wish to ensure some level of redundancy in the C2 function, but avoid complete duplication. We have formulated the problem under the assumption that a particular defensive interceptor can be commanded and guided by only a subset of the R C2 nodes. To put it another way, a particular C2 node can only launch and control a subset of the defensive interceptors. Under these assumptions it is possible to formulate a two-sided optimization problem whose solution would specify the optimal attack and defensive strategies, and would provide insight on how the level of redundancy deters the offense from attacking the C2 nodes. Unfortunately, this problem is so complicated so that we have been unable to obtain any analytical insights to its solution properties as yet.

Documentation: Partial documentation of the technical results can be found in the following:

[1]. P.A. Hosein, J.T. Walton, and M. Athans, "Dynamic Weapon-Target Assignment Problems with Vulnerable C2 Nodes," *Proc. 1988 Symposium on Command and Control Research,* Monterey, CA, June 1988, pp. 77-85 (published by SAIC, McClean, VA).

[2]. P.A. Hosein and M. Athans, "The Dynamic Weapon-Target Assignment Problem," *Proc. 1989 Symposium on Command and Control Research,* Washington, DC, June 1989 (LIDS-P-1887).

[3]. J.T. Walton and M. Athans, "Strategies for the Asset Defense with Precursor Attacks on the Command and Control System," *1989 Symposium on Command and Control Research,* Washington, DC, June 1989 (LIDS-P-1886).

[4]. P.A. Hosein, "A Class of Dynamic Nonlinear Resource Allocation Problems," Ph.D. Thesis, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, October 1989 (expected).

The doctoral thesis by J. Walton is targeted for completion in the summer of 1990.

4. RESEARCH PERSONNEL

Dr. Alexander H. Levis, Principal Investigator
Professor Michael Athans
Dr. Stamatios K. Andreadakis (to December 1988)
Mr. Didier Perdu - visiting researcher

Mr. Patrick Hosein - Graduate research assistant (Ph.D. Candidate)
Ms. Cindy Mok - Undergraduate research assistant
Mr. François Valraud - Graduate research assistant (MS)
Mr. James Walton - Graduate research assistant (Ph.D. Candidate)

5.0 INFORMATION DISSEMINATION

The following documents were issued as Laboratory Technical Reports or as Technical Papers. These were submitted to ONR, to the Basic Research Group of the JDL Panel on C^3, and to the distribution list specified in the contract. Some aspects of the work contained in these reports were supported in part by other related projects, such as the one from the Office of Naval Research on Distributed Tactical Decision Making (N00014-84-K-0519).

Theses

[T1]. C. Bohner, "Computer Graphics for System Effectiveness Analysis," S.M. Thesis, LIDS-TH-1573, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, May 1986.

[T2]. P. J. F. Martin, "Large Scale C3 Systems· Experiment Design and System Improvement," S.M. Thesis, LIDS-TH-1580, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, August 1986.

[T3]. H. P. Hillion, "Performance Evaluation of Decision Making Organizations Using Timed Petri Nets," S.M. Thesis, LIDS-TH-1590, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, August 1986.

[T4]. S. T. Weingaertner, "A Model of Submarine Emergency Decision Making and Decision Aiding," S.M. Thesis, LIDS-TH-1612, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, October 1986.

[T5]. P. A. Remy, "On the Generation of Organizational Architectures Using Petri Nets," S.M.

Thesis, LIDS-TH-1630, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, December 1986.

[T6]. J. Kyratzoglou, "Computer Aided Design for Petri Nets," S.M. Thesis, LIDS-TH-1694, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, August 1987.

[T7]. D. M. Perdu, "Modeling and Evaluation of Expert Systems in Decision Making Organizations," S.M. Thesis, LIDS-TH-1726, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, December 1987.

[T8]. A.-C. A. Louvet, "The Bounded Rationality Constraint: Experimental and Analytical Results," S. M. Thesis, LIDS-TH-1771, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, June 1988.

[T8] F. Valraud, "Evaluation of Functionality in Distributed Systems," MS Thesis, Technology and Policy Program, also Report LIDS-R-1868, Laboratory for Information and Decision Systems, Cambridge MA, 1989.

[T9]. P.A. Hosein, "A Class of Dynamic Nonlinear Resource Allocation Problems," Ph.D. Thesis, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, October 1989 (expected).

5.2 Technical Papers

[P1]. A. H. Levis, "Modeling and Measuring Effectiveness of C3 Systems," *Proc. 7th AFCEA European Symposium*, Brussels, Belgium, October 1986. (LIDS-P-1608)

[P2]. V. Y. Jin, A. H. Levis, and P. A Remy, "Delays in Acyclical Distributed Decision Making Organizations," *Proc. 4th IFAC/IFORS Symposium on Large Scale Systems,* Pergamon Press, Oxford 1086. (LIDS-P-1528)

[P3]. M. Athans, "Command and Control Theory: A Challenge to Control Science," *IEEE Transactions on Automatic Control,* Vol. AC-32, No. 4, April 1987. (LIDS-P-1584)

[P4]. A. H. Levis and M. Athans, "The Quest for a C^3 Theory: Dreams and Realities," *Proc. 1987 Symposium on C^2 Research*, National Defense University, Fort McNair, Washington DC, June 1987; also in *Science of Command and Control,* S.E. Johnson and A. H. Levis, Eds., AFCEA International Press, Washington DC, 1988. (LIDS-P-1691)

[P5]. H. P. Hillion and A. H. Levis, "Performance Evaluation of Decision Making Organizations," *Proc. 1987 Symposium on C^2 Research*, National Defense University, Fort McNair, Washington DC, June 1987. (LIDS-P-1683)

[P6]. A.C. Louvet, J. T., Casey, and A. H. Levis, "Experimental Investigation of the Bounded Rationality Constraint," *Proc. 1987 Symposium on C^2 Research*, National Defense University, Fort McNair, Washington DC, June 1987; also in *Science of Command and Control,* S.E. Johnson and A. H. Levis, Eds., AFCEA International Press, Washington DC, 1988. (LIDS-P-1680)

[P7]. S. T. Weingaertner and A. H. Levis, "Evaluation of Decision Aiding in Submarine Emergency Decision Making," *Proc. 1987 Symposium on C^2 Research*, National Defense

University, Fort McNair, Washington DC, June 1987. (LIDS-P-1688)

[P8]. P. J. F. Martin A. H. Levis, "Measures of Effectiveness and C^3 Testbed Experiments," *Proc. 1987 Symposium on C^2 Research*, National Defense University, Fort McNair, Washington DC, June 1987. (LIDS-P-1678)

[P9]. P. Remy, and A. H. Levis, "On the Generation of Organizational Architectures Using Petri Nets," *Proc. 8th European Workshop on Applications and Theory of Petri Nets*, Zaragoza, Spain, 24-27 June, 1987. To appear in *Advances in Petri Nets 1988*, G. Rozenberg, Ed., Springer-Verlag, Berlin, 1988. (LIDS-P-1634)

[P10]. H. P. Hillion, and A. H. Levis, "Timed Event-Graph and Performance Evaluation of Systems," *Proc. 8th European Workshop on Applications and Theory of Petri Nets*, Zaragoza, Spain, 24-27 June, 1987. (LIDS-P-1639)

[P11]. P. Remy, A. H. Levis, and V. Y.Y. Jin, "On the Design of Distributed Organizational Structures," *Proc. 10th IFAC World Congress*, Pergamon Press, New York, July, 1987; also in *Automatica*, Vol. 24, No. 1, 1988.

[P12]. S. T. Weingaertner, and A. H. Levis, "Evaluation of Decision Aiding in Submarine Emergency Decision Making," *Proc. 3rd IFAC/IFIP/IEA/IFORS Conference on Man Machine Systems*, Oulu, Finland, June 1988; also in *Automatics*, vol. 25, No.3, May 1989.

[P13]. D. M. Perdu, and A. H. Levis, "An Expert System Model Using Predicate Transition Nets," Laboratory for Information and Decision Systems, MIT, Cambridge, MA, January 1988. (LIDS-P-1736)

[P14]. A. H. Levis, "Human Organizations as Distributed Intelligence Systems," *Proc. IFAC Symposium on Distributed Intelligence Systems*, Pergamon Press, Oxford, 1988.

[P15]. D. M. Perdu and A. H. Levis, "Evaluation of Expert Systems in Decision Making Organizations," *Proc. 1988 Symposium on C^2 Research*, Monterey, CA, June 1988. (LIDS-P-1788)

[P16]. P.A. Hosein, J.T. Walton, and M. Athans, "Dynamic Weapon-Target Assignment Problems with Vulnerable C2 Nodes," *Proc. 1988 Symposium on Command and Control Research*, Monterey, CA, June 1988, pp. 77-85 (published by SAIC, McClean, VA). (LIDS-P-1786).

[P17]. P. A. Remy, and A. H. Levis, "Algorithmic Generation of Feasible System Architectures," *Proc. 12th IMACS World Congress on Scientific Computation*, Paris, France, July 1988.

[P18]. J. L. Grevet, L. Jandura, J. Brode, and A. H. Levis, "Execution Strategies for Petri Net Simulations," *Proc. 12th IMACS World Congress on Scientific Computation*, Paris, France, July 1988.

[P19]. I. M. Kyratzoglou and A . H. Levis, "Computer Aided Petri Net Design for Decision-Making Organizations," Laboratory for Information and Decision Systems, MIT, Cambridge MA, August 1988. (LIDS-P-1806)

[P20]. P.A. Hosein and M. Athans, "The Dynamic Weapon-Target Assignment Problem," *Proc. 1989 Symposium on Command and Control Research*, Washington, DC, June 1989

(LIDS-P-1887).

[P21] J.T. Walton and M. Athans, "Strategies for the Asset Defense with Precursor Attacks on the Command and Control System," *1989 Symposium on Command and Control Research,* Washington, DC, June 1989 (LIDS-P-1886).

[P22] F. Valraud and A. H. Levis, "On the Quantitative Evaluation of Functionality in C3 Systems", *Proc. 1989 Symposium on C2 Research,* National Defense University, Ft. McNair, Washington, DC, June 1989.

[P23] D. M. Perdu and A. H. Levis "Requirements Specification Using the Cube Tool Methodology", *Proc. 1989 Symposium on C2 Research,* National Defense University, Ft. McNair, Washington, DC, June 1989.

[P24] D. M. Perdu and A. H. Levis, "Analysis and Evaluation of decision Aids in Organizations" *Proc. 4th IFAC Symposium on Man Machine Systems,* Xian, P. R. China, September, 1989.

[P25] F. Valraud and A. H. Levis, "On the Quantitative Evaluation of Functionality in Distributed Intelligence Systems", *Proc. 4th IEEE Int'l Symposium on Intelligent Control,* Albany, NY, September 1989.

[P26] D. M. Perdu and A. H. Levis, "Requirements Specification for Distributed Intelligence Systems using the Cube Tool Methodology," *Proc. 4th IEEE Int'l Symposium on Intelligent Control,* Albany, NY, September 1989.