NPS-OR-94-009

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THE PROWLER IADS PERFORMANCE EVALUATION TOOL (PIPE)

Michael P. Bailey

Allan D. Crane

J. Peter Miluski

May 1994

Approved for public release; distribution is unlimited.

Prepared for:
Naval Postgraduate School
Monterey, CA 93943

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943-5000

Rear Admiral T. A. Mercer
Superintendent

Harrison Shull
Provost

This report was prepared for the Naval Postgraduate School.

Reproduction of all or part of this report is authorized.

This report was prepared by:

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
May 26, 1994

3. REPORT TYPE AND DATES COVERED
Technical

4. TITLE AND SUBTITLE
The Prowler IADS Performance Evaluation Tool (PIPE)

5. FUNDING NUMBERS

6. AUTHOR(S)
Michael P. Bailey, Allan D. Crane, and J. Peter Miluski

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER
NPS-OR-94-009

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The Prowler IADS Performance Evaluator is a computer simulation model of an airstrike protected by electronic countermeasures platforms. It is designed for integration into mission planning systems and analysis tools used to determine the effectiveness of electronic countermeasures or allocate scarce countermeasures equipment. PIPE's features include

- flexible hierarchical IADS specification,
- the capability to construct and calculate appropriate measures of performance,
- graphical presentation analysis results,
- mission visualization.

14. SUBJECT TERMS

15. NUMBER OF PAGES
19

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL

The Prowler IADS
Performance Evaluation Tool
(PIPE)

MICHAEL P. BAILEY
DEPARTMENT OF OPERATIONS RESEARCH
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943, U.S.A.

ALLAN D. CRANE
J. PETER MILUSKI
PRB ASSOCIATES, INC.
HOLLYWOOD, MD 20636, U. S. A.

May 26, 1994

ABSTRACT. The Prowler IADS Performance Evaluator is a computer simulation model of an airstrike protected by electronic countermeasures platforms. It is designed for integration into mission planning systems and analysis tools used to determine the effectiveness of electronic countermeasures or allocate scarce countermeasures equipment. PIPE's features include

- flexible hierarchical IADS specification,
- the capability to construct and calculate appropriate measures of performance,
- graphical presentation analysis results,
- mission visualization.

1. INTRODUCTION

Current mission planning systems for strike aircraft feature high quality graphical presentation of geographic information in the mission area. While effective in some planning processes, these tools are inadequate for planning or analyzing the effects of electronic countermeasures (ECM). What is missing in these systems is the capability to predict events in time.

PIPE evaluates user-specified condition and event measures of effectiveness of ECM taken against an integrated air defense system (IADS). The system models the behavior of the IADS network using a modification of stochastic timed Petri networks, and produces samples of the IADS time-domain behavior. An object-oriented simulation model is used to generate these samples by simulating the interactions of the protected strike group, jamming assignments, and IADS objects. IADS sensors, communications assets, and command structure are modeled using a template structure populated with data from the Air Force Electronic Warfare Center (AFEWC) Constant Web database [6]. The system then allows ECM planners to allocate scarce jamming resources while cognizant of the impact of jamming assignments on the IADS performance.

2. BACKGROUND

The primary mission of the U. S. Navy's EA-6B aircraft is to provide ECM to disguise or protect other platforms. Most often, these protected platforms are tactical bombers formed into a strike group. Heretofore, we refer to the protected platforms collectively as the *protected entity* (PE). The Tactical EA-6B Mission Support system (TEAMS) system has been providing support for EA-6B mission planning for nearly a decade.

Soon after the TEAMS system was developed and fielded, mission planners realized a tremendous leap forward in Electronic Countermeasures Officer (ECMO) proficiency, however experienced ECMOs still did a great deal of off-line planning. Although the system provided reasonable presentation of the geographic relationships of locations, weapon systems, and emitters, it failed to reflect the organizational relationship between these entities.

The Advanced Capability TEAMS (ATEAMS) was designed, in part, to correct this shortcoming. The design of ATEAMS uses the notion of parent-child organizational relationships between IADS elements, as well as emitter neighborhoods (groupings of IADS elements which share data, [7]). These relationships can be displayed to the user, and can be used as input to the mission planning process. Using these relationships requires that the ECMO assimilate graphical information about emitter and weapon system relationships, and produce thumbrules useful in allocating scarce jamming resources to threat emitters.

It is difficult to use the graphical display of weapon system and emitter relationships as a jammer planning aid. The impact of the relationship of a set of emitters on the performance of the IADS is not explicit when presented in the context of a geographical display. Furthermore, the relationship itself seems to have little to do with geography – the relationship affects **reaction time**!

In appreciating the impact of IADS node interconnections, the mission planner must change his frame-of-mind from geographic to temporal. The purpose of ECM used by the EA-6B and other platforms has less to do with providing space in which the assigned protected entity can fly unmolested, and more to do with delaying or controlling detection times of the protected entity. *The true goal of ECM on the battlefield must be stated as the control of the battle timeline.*

Strike mission planning must evolve into a process aimed at controlling *when* events occur on the battlefield. Often, ECM is used to delay detection events to such a degree that the enemy is powerless to prevent the protected entity from safely accomplishing its mission. Deception ECM is designed to induce diversion of enemy resources from the protected entity. This is accomplished by controlling what the enemy sees *before* he detects the protected entity, so that his reaction is inappropriate. Analyzing which IADS nodes perceive which threats, and when these perceptions are made and revised, is impossible using geographically-based planning tools.

A method for examining and evaluating the time-domain performance of the IADS as the engagement evolves is required. Without such a method, the value of data about the relationships of IADS nodes is minimal.

3. MODELING THE SYSTEM

This section describes the backbone which we use to model the evolution of the engagement. This model, first presented in [1] is called the stochastic relaxed Petri network (SRPN). This model is a generalization of the stochastic timed Petri network model given in Haas and Shedler [9], where the relaxations are:

1. allowance of transition durations which are either instantaneous, of deterministic length, or a positive random length;
2. each transition is allowed to happen in a specified deterministic time window, and is prohibited from happening outside the window;
3. a generalized enabling condition structure.

3.1. PERT Networks. Let $n_1, n_2, \ldots, n_N$ be a set of nodes in a network. Let two nodes be joined by a directed arc X which has a natural tail and a natural head, denoted tail(X) and head(X). In Petri nets, activity diagrams, or PERT charts, the arcs represent tasks and the nodes represent milestones.

The rules for starting tasks are: *A task X emanating from a node n may not commence until all of the tasks pointing into n are complete.* Thus, the task `put shoes on` cannot be started until the task `put socks on` is completed. This simple rule dictates the time required to complete the entire task. When all of the tasks pointing into a node are complete, the node is said to be *enabled*, and all of the tasks pointing out of that node begin in parallel.

Some interesting observations can be made about a process like a PERT network. Most interesting is the fact that most of the tasks in the diagram have little effect on the time of completion. There exists a path running through the network which contains tasks whose completion times dominate the time of completion of the entire project. This path, called the *critical path*, can be identified by looking at the last arc which completes in the network, then working backward.

We can construct some very interesting structures within this framework. However, we need to slightly extend this system to include structures which model IADS activity. This extension is called the Relaxed Petri Net.

3.2. Relaxation. The leap to IADS structures from basic PERT networks is intuitive. Instead of the tasks relating to the donning of a particular piece of clothing, they will be detection, communications, and decision tasks. These also take time to complete, and have a precedence structure similar to the one just described. However, in the IADS structures we will construct, there will usually be redundant sources of information, redundant communications, several decision makers, and several data fusion nodes. Hence, we need to be a little less strict about the conditions under which a task can begin.

We wish to relax the requirement that *all* of the tasks pointing into the tail of a task must be completed before the task can start. Suppose instead that each node in the structure has two sets of tasks pointing into it:

- the essential ones - these MUST be completed before the outpointing tasks can commence;
- the nonessential ones - those which are redundant and are all of equal value.

We shall denote the essential tasks for node n as E_n and the nonessential tasks for node n as E'_n. We will also allow the node to specify how many of the nonessential arcs need to be completed before the outpointing tasks can commence. This number, denoted k_n, will be specified for each node.

As we can see from the Figure 1, only two things are really going to be required before the TA detection task can begin

- the command to allow the TA search to begin must be given;
- there must be some sort of early warning message passed to the emitter.

Obviously, this structure is not appropriate for every IADS, but it serves as a good example.

The tasks associated with the EW data are nonessential – we only need one EW data que to begin the TA detection task. However, we do require that there be a command given to allow us to start the detection process. Hence, the tasks $a, b, c = E_n$, $d = E'_n$, and $k_n = 1$.

3.3. Hierarchical Construction. To make PIPE viable in a mission planning system, there needs to be a hierarchical system for constructing IADS SRPNs. The network we have described so far is constructed of arcs and nodes. If we were to construct the entire IADS SRPN with these simple objects, we would produce a structure which was not recognizable or usable by mission planners.

What is required to improve this situation is a way to describe and design the IADS SRPN using templates, well-understood low level models of specific network pieces, and a language for linking simple pieces together to make more complex IADS structure which we can analyze. We will do this by describing the system in terms of Networks, Subnets, and Simple Arcs. We will describe these objects from the simplest to the most complex.

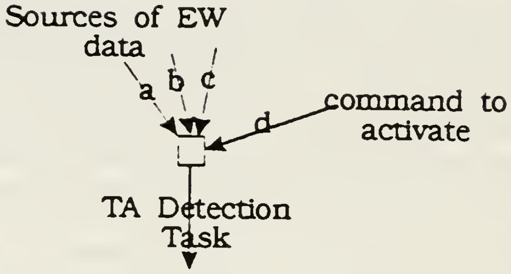

Figure 1: Fragment of an IADS SRPN. EW data sources are redundant, only one data source is required to begin the task `TA Detection`.

Simple Arcs. Simple arcs correspond to detection, data fusion, decision, and communications activities – these are the tasks described in the previous sections. A simple arc is always tied to a specific geographic location, and may be an ECM target. Simple arcs can be knitted together to make more sophisticated structures called subnets. The relationship between a simple are and the evolution of the engagement comes from its location (geographical and structural) and its stand-alone performance characteristics, like its range and speed of execution.

Subnets. Subnet objects are template-like specifications of network fragments. Subnets are constructed of simple arcs, or of other subnets, plus the required nodes. In fact simple arcs are objects derived from subnet objects. Each subnet has

1. a set of nodes which are sources for the subnet;

2. a set of nodes which form the ends of the subnet;

3. a set of interior nodes;

4. a set of subnets which connect one interior node to another.

Simple arcs are considered subnets with no interior nodes.

Networks. For ease of construction, we will insist that every subnet data set have one subnet called IADS. When the IADS is retrieved from the subnet data, the system will then instantiate the IADS and look for all the interior subnets called for in the IADS specification. This iterative expansion of subnets continues until the IADS is complete.

Example. We have constructed a set of subnets which allow us to expose the richness of the structure we use. Lets establish some terminology, and to show the data set required to instantiate this simple system. The example network is constructed to be a torture test for our subnet paradigm. It has more complexity and depth than any real system. It has the required IADS subnet, containing one internal subnet called network 1. The subnet network 1 takes some simple Arc 1's and Arc 2's to connect to parallel subnets both of type system 1. System 1 is, in turn, comprised of a simple arc, Arc 5, and two types of subsystems, subsys1 and subsys2. Each of these subsystems are comprised of simple arcs, called Arc i. Figure 2 shows the hierarchy of subnets used for this example. The specification of E_n, E'_n and k_n for each node n is difficult to show in a diagram, but resides in the appropriate data for the network.

Our example network has over 210 simple arcs connected together. When PIPE reads the appropriate files and constructs the IADS, it uses each of these templates in an iterative manner.

The true power of the subnet structure is seen when we attempt to modify the structure of the example. Suppose that we had a different subsys 1 structure which we just learned about. Let's call this new subsys1 subsys 1a. If we change the specification of system 1 in the three instances of subsys 1, replacing each with a subsys 1a, and then enter subsys 1a's structure in the subnet data file, we have accomplished the switch.

Of keen interest might be the use of this capability to *inherit and modify*, as we do in object-oriented programming. In this vane, suppose subsys 1 actually had an arc which emanated from node ss4 to some new node ss5. We could accomplish this alteration by making subsys 1a a two-subnet system with one interior node. One of the arcs in subsys 1a would be of type subsys1, and the other would represent the new arc. The single interior node of subsys 1a would take the place of ss4.

3.4. Timing the Engagement. PIPE models the PE as a single point mass which travels through the engagement area. Based on an emitter's maximum range, the detection task associated with an emitter cannot commence until:

- the PE is within the maximum range of the emitter;
- all of the structural SRPN requirements are met.

A task is terminated when either of these requirements is no longer met, either the PE goes out of range, or some other IADS element which was supporting the detection process goes away. Communications assets are restricted only be the structural constraints. Detection tasks which are complete (the radar is locked onto the PE) remain unaltered until the PE exits the maximum range. Thus, each task in the IADS is limited to a time window in which it can be completed, where the window is determined by movement of the PE.

The jamming aircraft are similarly modeled as moving point masses, but they have the ability to employ ECM against emitters in the IADS. Jammer effects are modeled by two means:

- range ring reduction for detection tasks;
- speed degradation for any vulnerable task.

Each jamming modulation/emitter pair has a set of degradation data used in the model.

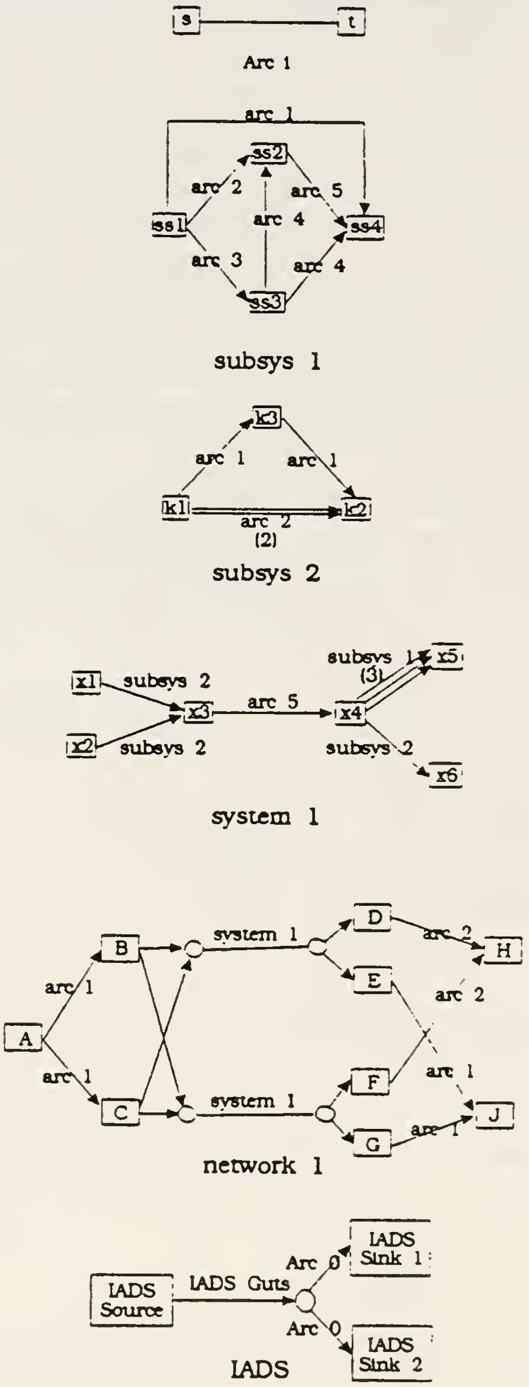

Figure 2: Example Network Structure which Challenges the Ability of the Subnet Paradigm to the Fullest.

Mission Visualization. The navigation of the PE and jammers is modeled using the object-oriented toolbox called the *Platform Foundation* [2]. This tool also provides an animation of the mission using a map of the mission area, moving icons representing platforms, and multicolored range rings to show the status of an emitter. This technology provides a crude but effective mission rehearsal and analysis capability, but the users of PIPE are cognizant of the shortcomings of watching the animation of a small set of replications using an animation.

4. MEASURING EFFECTIVENESS OF THE IADS

In this section, we discuss the construction of measures of effectiveness (MOEs) for the IADS during an analyzed mission. Before proceeding, we must explicitly acknowledge the importance of MOEs in the strike mission planning process. Ultimately, the IADS is designed to destroy aircraft as they travel toward the target, so as to prevent the aircraft from destroying the target. The strike group seeks to overcome this protection. Here we describe events and conditions which indicate whether the IADS succeeds, hence the degree to which the IADS is defeated.

In our model of the IADS performance, we stop short of calculating the distribution of the number of missiles fired at the strike group because we profess to know little about the weapons employment of the enemy in any particular scenario. Instead, we will calculate the distribution of several events and conditions which are strong indicators of the lethal threat posed by the IADS. In what follows, we describe a flexible way for the user to design their own MOE, and give suggested standard measures.

4.1. Condition and Event MOEs. There are two basic categories for measures of IADS effectiveness, these being

- the first time a specified event occurs, and
- the mission time during which a condition exists.

Examples of the former class are the time until a specified neighborhood can track the PE with target acquisition radar, the time that any target tracking radar tracks the target, or the time at which an early warning broadcast is issued. The latter class contains MOEs such as the time in the mission when the strike group is tracked by a specified set of target tracking radars, or the length of time that the target acquisition solution remains current. We will heretofore refer to the former class as *event MOEs* and the latter as *condition MOEs*.

For an event of interest, e.g. the time the first target tracking radar in the IADS locks onto the PE, let E be a random variable representing the time the event occurs. We will estimate the cumulative probability distribution function F_E, which is defined as

$$\begin{aligned} F_E(t) &= P[\textit{the event occurs before } t] \qquad (1) \\ &= P[E < t]. \qquad (2) \end{aligned}$$

The associated probability density function, $f_E(t) = \delta F_E(t)/\delta t$ gives us the rate of occurrence of E over any time interval. Thus, if E is the time that the first target tracking radar locks on, then $F_E(t)$ tells us the likelihood that lock on occurs at time t or before, and $f_E(t)$ gives us relative measures of the times that E is likely to occur.

Let $e_1, e_2, \ldots, e_m$ be a set of observations of E generated from independent simulation replications. We wish to construct our best-guess estimate of $F_E(t)$. This guess is denoted $\hat{F}_E$, and can be constructed via a histogram of the outcomes e_i as follows. Suppose that we have a histogram with

k evenly spaced cells between some lower and upper bounds L_E and U_E, each cell being exactly $w = \frac{U_E - L_E}{k}$ wide. We tabulate the $e_i, i = 1, 2, \ldots, n$ samples in the cells so that

$$\hat{f}_i = [number\ of\ e_i\ in (L_E + (i-1)w, L_E + iw)]. \tag{3}$$

Thus, $\hat{f}_i, i = 1, 2, \ldots, k$ is a rough estimate of the probability density of E in $(L_E + (i-1)w, L_E + iw)$. We can then construct the empirical probability distribution function $\hat{F}_E$ using

$$\hat{F}_E(x) = \sum_{l=1}^{m:x<L_E+(m+1)w} \hat{f}_l + \frac{(x - (L_E + mw))}{w} \hat{f}_m. \tag{4}$$

The first term is obviously the relative mass of observations of e_i which fall in cells completely below x, while the second term is the linear interpolation of $\hat{f}$ in the cell to which x belongs. This method is extremely common and is explained fully in [5].

For condition MOE C, we define the function

$$p_C(t) = P[C\ persists\ at\ time\ t]. \tag{5}$$

This function is *not* a probability density function in t, but shows the likelihood that condition C persists at a point in time. By integrating $p_C(t)$ over $[0, T]$,

$$P_C(T) = \int_0^T p(t)dt \tag{6}$$

we can derive a function which shows accumulated exposure to a condition over the length of the mission timeline.

For any condition MOE, the stretch of time that the condition persists is recorded on a histogram-like object, one count of frequency is added to each time interval that the node was enabled. When the histogram is presented, the profile presented is the probability that the condition was true in each time interval:

$$\begin{aligned} \hat{p}(t) &= \frac{number\ of\ times\ condition\ true\ at\ t}{number\ of\ replications} && (7) \\ &\approx P[condition\ true\ at\ t]. && (8) \end{aligned}$$

4.2. MOE Nodes. We can use some extra nodes in the network to facilitate the collection of MOEs. These special nodes are called `MOENodeObjs`, and have extra capabilities beyond those of run-of-the-mill `NodeObjs`. Whether an MOE node is an event node or a condition node, it needs to be integrated into the IADS SRPN. All MOEs can be calculated once we know the time that all of the nodes (milestones) in the IADS SRPN are enabled. In the examples above, the MOE is attached to IADS SRPN nodes as follows:

- EVENT: IADS-wide TT LOCK-ON: enabled when the first TT lock-on node is enabled anywhere in the IADS;
- CONDITION: Neighborhood 1 TA SOLUTION: enabled as long as the neighborhood's TA data fusion element is actively supporting the neighborhood.

We can get this information *for free* by adding nodes (MOE nodes), connected with zero-duration `Facilitator` arcs to the nodes which comprise the MOE. This is possible by using objects which make it easy to snatch-up groups of nodes and connect them to the MOE nodes after the rest of the IADS is constructed. Using zero-length tasks is a standard PERT network technique, see [13].

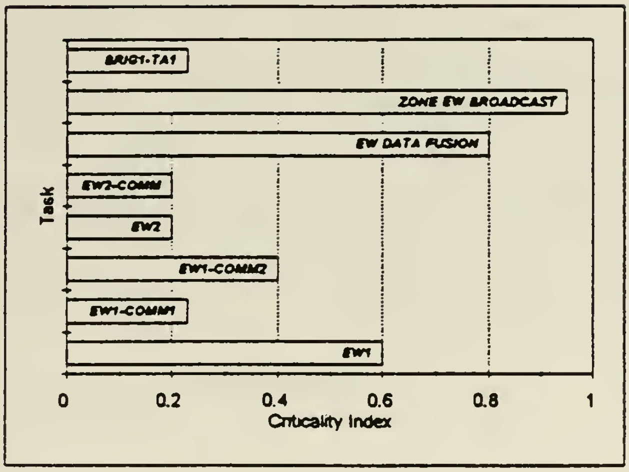

Figure 3: Criticality Indices for some Tasks in an Example Network Relative to the MOE `EVENT: IADS-wide Engagement Phase Begins`

4.3. Off-The-Shelf Measures. The flexibility afforded by the MOE nodes and bundles enables the user of PIPE to monitor any identifiable event or condition. For ease of use, there are a selection of standard event and condition MOEs available for evaluation in the analysis or planning process, as well as the capability to examine the performance of user-specified IADS nodes.

The standard MOEs are constructed by considering the phases of the engagement, early warning, target acquisition, and engagement. We wish to get the information about events indicating the transition from one phase to the next, as well as the condition that the engagement is in a given phase. Finally, we want this information for the entire IADS, for each data-sharing emitter neighborhood, and for each individual detection or communications emitter. After an IADS is constructed, MOE nodes and bundles are added for all of these measures automatically. When the model is run for m replications, $\hat{f}$ and $\hat{p}$ are automatically constructed.

5. NETWORK ANALYSIS

5.1. Criticality. Let n be a MOE node. Every time that we sample a replication of the SRPN performance, we can perform the backward chaining, starting at n, to determine the tasks which contributed directly to the time n was enabled. Since each of these replications are statistically independent, it is likely that the tasks which are on the *critical path* to n will be different in each replication.

Let

$$c_X = P[task\ X\ is\ critical]. \tag{9}$$

c_X is called the *criticality* of task X with respect to milestone n. We can measure the importance of any task X to the accomplishment of milestone n using c_X – if we interfere with the speed at which X is performed, we can expect to interfere with the accomplishment of milestone n with probability c_X. Calculating the criticality of a task is a straightforward affair, it is efficient, and can be done without any specific ideas about how we wish to alter the jamming plan or other factors.

We should point out that the criticality indices of each task depend on the choice of n. Thus, we need to choose a central MOE for criticality analysis before replications are done. Furthermore, if n is an MOE like `EVENT: IADS-wide Engagement Phase Begins`, all of the tasks which are important

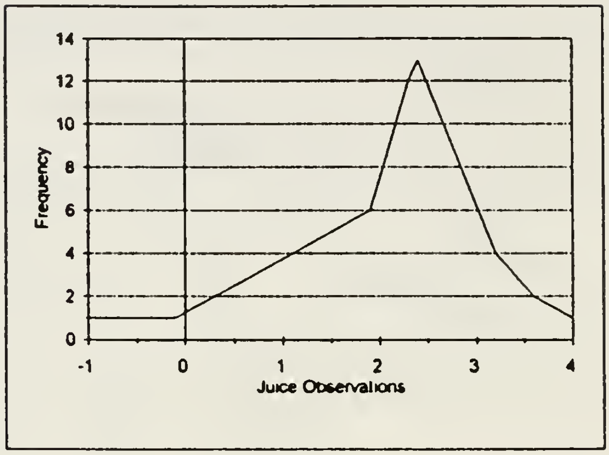

Figure 4: Frequency Plot for Juice for the MOE `EVENT: IADS-wide Engagement Phase Begins` for two alternative Jammer Gameplans. The calculations show $\bar{d} = 2.20$, with a 95% confidence interval of $(1.98, 2.41)$ based on 53 replications. Clearly, $S1$ is inferior to $S2$, because the difference is significantly positive. Time units are in minutes.

to n will have high criticality indices. This includes tasks which are EW and TA detections and communications. Hence, we can use criticality indices for engagement phase transition to target emitters used in all of the phases of the engagement.

Let δ_X be an indicator variable

$$\delta_i(X) = \begin{cases} 1 & X \textit{ is on the critical path} \\ 0 & \textit{otherwise.} \end{cases} \tag{10}$$

Then we can estimate the criticality of X as

$$\hat{c}_X = \frac{\sum_{i=1}^{n} \delta_i(X)}{n}. \tag{11}$$

$\hat{c}_X$ is simply the proportion of the replications which observed X as critical to n.

The shortcoming of using criticality measures to guide decisions concerning jammer management is that the *yield* in terms of delaying n is not stated, only that there will be *some* yield. c_X is an *instantaneous* yield estimate, in that the expected time n occurs will increase $c_X \delta t$ if we increase the time X takes to complete by δt, given that δt is infinitesimally small.

5.2. Juice. Let n be an MOE we are interesting in controlling using jamming assignments and other actions. Let S_1 and S_2 be two collections of jamming assignments (these may also include other actions like route alterations, HARM missile employments, or strike package composition). The question we wish to answer is, "What is the effect of changing from S_1 to S_2 on the time n occurs?" We replace this formal question with the slang, "How much *juice* is in the altered jammer management plan S_2?" This is particularly appropriate if S_1 and S_2 differ in only one assignment, say against task X associated with a target emitter. In this case, we are seeking the *juiciness* of adding the assignment against X.

To estimate this quantity, we collect m replications of the SRPN performance under each of the sets of conditions S_1 and S_2, and pair the samples as $(e_i^1, e_i^2), i = 1, 2, \ldots, m$. Let d_i be the difference within pair i,

$$d_i = e_i^1 - e_i^2. \tag{12}$$

The collection of sample differences $d_i, i = 1, 2, \ldots, m$ is comprised of independent samples, and is amenable to constructing empirical distribution functions and sample moments.

6. THE MISSION PLANNING PROCESS USING PIPE

We have described a set of capabilities useful to the ECM mission planner, but need to describe their use in the planning process. The mission planning process in this domain is limited to selecting targets for ECM, choosing ECM transmitters for the mission, and choosing a preprogrammed reaction to detection of threat signals by the ECM aircraft or the PE. Collectively, we call these decisions the *jammer gameplan*. Our goal is to use the information provided by PIPE to improve the gameplan.

We start the PIPE planning support with the following in hand:

- a current gameplan;
- PE route;
- all of the automatically generated MOE and criticality data for the current gameplan.

Our highest priority in mission planning is to counter lethal threats to the PE. Thus, standard measure `EVENT: IADS-wide Engagement Phase Begins` is the initial key for planning purposes. We wish to delay this event as much as possible, and raise the probability that it doesn't occur at all.

We will use an established sequence to improve our gameplan:

1. ASSESS: Determine how well we do with the current gameplan – use the mission visualization tools and IADS engagement phase event and condition MOE displays.
2. SEARCH: Determine which elements in the IADS are good candidates for jammer targets – use criticality indices for each task, neighborhood and platform MOEs, and visualization.
3. EXPERIMENT: Determine the effect of attacking some subset of the candidate targets – calculate and tabulate assignment juice for each choice
4. REASSIGN: Adopt a superior subset of the candidate targets into the current gameplan.

By looking at a graph like Figure 5, we get an initial feel for the efficacy of the current gameplan. Our goal is to move the *hump* in the event graph to the right, delaying the onset of the lethal phase of the engagement. We wish to press the condition graph down and to the right. Recall `EVENT: IADS-wide Engagement Phase Begins` can be used to target tasks which are EW and TA detections and communications.

After iterating through the sequence a number of times, the planner should reach a point of diminishing returns in juice for any assignment or gameplan alteration. At that point, the planner can repeat the process for `EVENT: IADS-wide Acquisition Phase Begins`. This should be done only under the constraint that no assignment used to counter engagement MOEs be dropped.

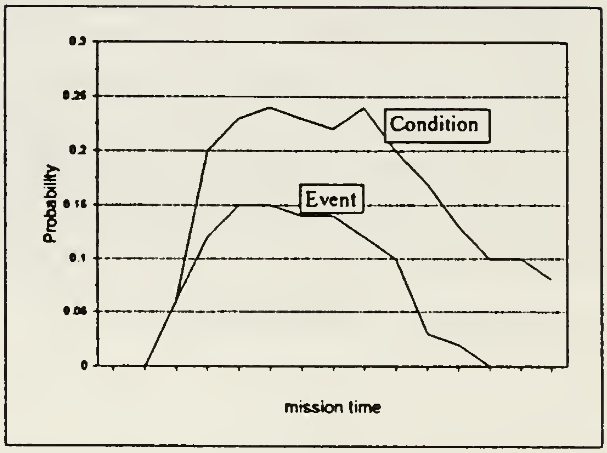

Figure 5: Probability Plot for Event and Condition MOE Estimates for Neighborhood 2: Target Acquisition

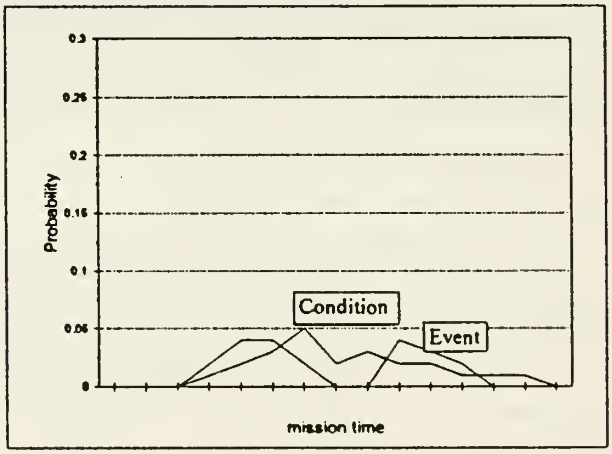

Figure 6: Probability Plot for Event and Condition MOE Estimates for Neighborhood 2: Target Acquisition after Some Action is Taken to Counter Target Acquisition.

7. CONCLUSION

PIPE is a powerful tool to analyze the performance of an ECM mission gameplan. This analysis is conducted by

- simulating the engagement of the EA-6B's, the IADS, and the PE;
- measuring the performance of the IADS in a way useful to the ECM mission planner;
- identifying likely jammer targets for enhancing the gameplan;
- facilitating direct comparisons between gameplans so that superior gameplans can be identified.

PIPE does not isolate jammer targets without the experienced hand of the ECMO, but it allows the ECMO to experiment with the gameplan before encountering the IADS in person. PIPE allows a limited mission rehearsal, and provides graphical demonstration of the value of ECM in the strike mission.

GUIDE TO NOTATION

AFEWC	Air Force Electronic Warfare Center
E	time an event of interest occurs
$E_n(E'_n)$	Essential (nonessential) tasks pointing into a node n
ECM	Electronic Countermeasures, jamming radar or communication transmissions
EW	Early Warning
k_n	number of nonessential tasks pointing into node n before it is enabled
IADS	Integrated Air Defense System, a network of sensors, command, control, and communications assets designed to protest an area from bombing by strike aircraft
L_E, U_E	bounds on the tabulation of empirical density for event E
MOE	Measure of Effectiveness
MG	Missle Guidence
$n_1, n_2, ..., n_N$	node set for a network model
PE	Protected Entity, the strike group
PERT	Project Evaluation and Review Technique, methodology for project milestone planning
PIPE	Prowler IADS Performance Evaluator analysis software
SRPN	Stochastic Relaxed Timed Petri Net
TA	Target Acquisition
TEAMS	The Tactical EA-6B Mission Support System, mission planning computer system for ECM aircraft
TT	Target Tracking
X	task in a PERT network or SRPN

References

[1] Bailey, M. 1992. Measuring Performance of Integrated Air Defense Networks Using Stochastic Networks. *Operations Research* 40(4) 647-659.

[2] Bailey, M. 1994. The Naval Postgraduate School Platform Foundation. Technical Report NPS-OR-94-005, Department of Operations Research, Naval Postgraduate School, Monterey, California.

[3] Banks, and Carson. 1984. *Discrete Event System Simulation*, Prentice-Hall, New York.

[4] Bratley, Paul, Bennett Fox, and Linus Schrage. 1983. *A Guide to Simulation*, Springer-Verlag, New York.

[5] Chambers, J. M, W. S. Cleveland, B. Kleiner, and P. A. Tukey. 1983. *Graphical Methods for Data Analysis.* Pacific Grove, California: Wadsworth & Brooks-Cole.

[6] *Constant Web Database Specification.* 1992. Special Project, U. S. Air Force, Air Force Intelligence Command, San Antonio, Texas. (SECRET-NOFORN)

[7] Crane, A. 1989. *Counter Network Mission Planning,* Document #89-S931-0, PRB Associates, Inc., Hollywood, MD 20636.

[8] Fishman, G. S. 1978. *Principles of Discrete Event Simulation.* New York: John Wiley and Sons.

[9] Haas, P. and S. Shedler. 1988. *Stochastic Timed Petri Nets.* Report available from IBM Almaden Research Center, San Jose, CA 95120.

[10] Law, A. and D. W. Kelton. 1982. *Simulation Modeling and Analysis,* McGraw-Hill Book Company, New York.

[11] Lewis, Peter A. W. and E. J. Orav. 1989. *Simulation Methodology for Statisticians, Operations Analysts, and Engineers, Volume 1,* Wadsworth & Brooks/Cole, Pacific Grove, CA.

[12] *MODSIM II Programming Language Reference Manual,* Version 1.6, CACI, LaJolla, CA.

[13] Wagner, H. 1972. *Principles of Operations Research.* New York: John Wiley and Sons.

INITIAL DISTRIBUTION LIST

1. Research Office (Code 08) 1
   Naval Postgraduate School
   Monterey, CA 93943-5000

2. Dudley Knox Library (Code 52) 2
   Naval Postgraduate School
   Monterey, CA 93943-5002

3. Defense Technical Information Center 2
   Cameron Station
   Alexandria, VA 22314

4. Department of Operations Research (Code OR) 1
   Naval Postgraduate School
   Monterey, CA 93943-5000

5. Prof. Michael P. Bailey (Code OR/Ba) 50
   Naval Postgraduate School
   Monterey, CA 93943-5000