



# AMSAA



6

AMSAA TECHNICAL REPORT NO. 559

USE OF PETRI NETS IN THE SIMULATION OF  
COMMAND AND CONTROL SYSTEMS

JOHN J. DiLEO

OCTOBER 1994



19950118 066

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

**U. S. ARMY MATERIEL SYSTEMS ANALYSIS ACTIVITY  
ABERDEEN PROVING GROUND, MARYLAND**

#### DISPOSITION

When this report is no longer needed, Department of the Army organizations will destroy it in accordance with the procedures given in AR 380-5. Navy and Air Force elements will destroy it in accordance with applicable directives. Department of Defense contractors will destroy the report according to the requirements of Section 14 of the Industrial Security Manual for Safe-guarding Classified Information. All others will return the report to the U.S. Army Materiel Systems Analysis Activity, Aberdeen Proving Ground, Maryland 21005-5071.

#### DISCLAIMER

The findings in this report are not to be construed as an official Department of the Army position unless so specified by other official documentation.

#### WARNING

Information and data contained in this document are based on the input available at the time of preparation. The results may be subject to change and should not be construed as representing the AMC position unless so specified.

#### TRADE NAMES

The use of trade names in this report does not constitute an official endorsement or approval of the use of such commercial hardware or software. The report may not be cited for purposes of advertisement.

# 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 October 1994	3. REPORT TYPE AND DATES COVERED FINAL	
4. TITLE AND SUBTITLE Use of Petri Nets in the Simulation of Command and Control Systems		5. FUNDING NUMBERS	
6. AUTHOR(S) John J. DiLeo		8. PERFORMING ORGANIZATION REPORT NUMBER TR-559	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Director, U.S. Army Materiel Systems Analysis Activity Attn: AMXSU-CA Aberdeen Proving Ground, MD 21005-5071		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Director, U.S. Army Model Improvement and Study Management Agency Attn: L. McGlynn 1725 Jefferson Davis Highway (Suite 808) Arlington, VA 22202		11. SUPPLEMENTARY NOTES	
12a. DISTRIBUTION / AVAILABILITY STATEMENT <u>DISTRIBUTION STATEMENT A</u> : Approved for public release; distribution is unlimited.		12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) <p>Petri Nets were first proposed by C.A. Petri in 1962, in his doctoral dissertation, entitled <u>Kommunikation mit Automaten</u>. Modeler, a simulation package based upon STAPN's--an extended version of Petri Nets--was developed under contract to the Air Force Systems Command Foreign Technology Division. Further enhancements to Modeler were made under contract to the Army Training and Doctrine Command (TRADOC) Analysis Center (TRAC)-Operations Analysis Center (OAC). Several studies and Cost and Operational Effectiveness Analyses (COEA's) at TRAC-OAC have utilized models developed with Modeler.</p> <p>This study effort was comprised of three major components: verification of Modeler's simulation engine; comparison of results generated by Modeler-based simulations to those of simulations written in procedural languages; and exploration of the overall suitability of Modeler-based simulations for production studies.</p> <p style="text-align: center;"><i>DISCONTINUED</i></p>			
14. SUBJECT TERMS Petri Nets; Simulation; Command and Control; STAPN's		15. NUMBER OF PAGES	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED		16. PRICE CODE	
18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED		19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	
20. LIMITATION OF ABSTRACT UL			

ACKNOWLEDGEMENTS

The U.S. Army Materiel Systems Analysis Activity wishes to acknowledge the following individuals for contribution to this report:

Author:

John J. DiLeo

Peer Reviewer:

Eric Snyder, Logistics Analysis Division

The author wishes to recognize the support of the following individuals in providing background information for relevant sections of this report:

Mr. Kelley Stephens, U.S. Army Training and Doctrine Command  
Analysis Center-Studies and Analysis Center

Mr. Paul Zimmerman, National Ground Intelligence Center

Editorial Support:

Cinnie Henry  
Marian Brooks

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification .....	
By .....	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

The next page is blank.

CONTENTS

	Page
ACKNOWLEDGEMENTS . . . . .	iii
LIST OF TABLES . . . . .	vii
LIST OF FIGURES . . . . .	vii
1. INTRODUCTION . . . . .	1
2. BACKGROUND . . . . .	1
2.1 Stochastic Timed Attributed Petri Nets (STAPNs) . . . . .	1
2.2 The AMSAA Simulation Technology (SIMTECH) Project . . . . .	4
2.3 Development of Modeler . . . . .	4
3. RESULTS OF AMSAA EFFORTS . . . . .	6
3.1 Validation of Modeler--Basic Queueing Models . . . . .	6
3.2 Petri Net Fire Support Communications Model (PN-FSCom) . . . . .	8
3.3 All-Source Analysis System (ASAS) Network Model (ASASNET) . . . . .	13
3.4 The Fire Support Command and Control Analysis Tool (FISCCAT) . . . . .	15
4. LIMITATIONS OF MODELER. . . . .	17
5. OTHER ARMY PETRI NET EFFORTS . . . . .	20
5.1 TRAC-OAC/SAC . . . . .	20
5.2 FSTC/NGIC . . . . .	20
6. THE FUTURE OF MODELER AT AMSAA . . . . .	21
7. CONCLUSIONS . . . . .	21
REFERENCES . . . . .	23
LIST OF ACRONYMS . . . . .	25
DISTRIBUTION LIST . . . . .	27

The next page is blank.

## LIST OF TABLES

Table No.	Page
1. Input Parameters for Queueing Models. . . . .	8
2. Queueing Model Performance Parameters (Theoretical Values vs. Modeler Results). . . . .	9
3. Detailed Modeler Results-- <i>M/M/4</i> Queueing Model. . . . .	10
4. PN-FSCom Messages Generated: Expected vs. Observed (No Time Limit). . . . .	11
5. Network Performance Statistics: Comsim vs. PN-FSCom (Modeler, 20 replications). . . . .	13

## LIST OF FIGURES

Figure No.	Page
1. A Petri Net Structure Represented by: (a) A 4-tuple of Set Definitions; and (b) A Graph . . . . .	1
2. Petri Net Transition Firing. . . . .	2
3. Graphical Elements Used in Modeler: (a) Place; (b) Transition; (c) Clustering Box; (d) Standard Arc; (e) Enable Arc; (f) Inhibit Arc. . . . .	5
4. Modeler Structure Used in Queueing Model Verification: (a) Top-level Structure; (b) Arrival Generator; (c) Servers. . . . .	6
5. Modeler Structure for <i>M/M/∞</i> → <i>M/M/4</i> Series Queueing Model. . . . .	7
6. Comsim "Battalion Independent User Center (BN IUC) Input" Thread Diagram. . . . .	11
7. PN-FSCom Top-Level Structure. . . . .	12
8. A Segment of FISCCAT Code in (a) SIMSCRIPT; and (b) Modeler. . . . .	16
9. PN-FISCCAT Top-Level Structure. . . . .	17
10. PN-FISCCAT: Forward Observer (FO) Structure. . . . .	18
11. PN-FISCCAT: FO_Oper_Rds_Landed Function. . . . .	18

The next page is blank.

# Use of Petri Nets in the Simulation of Command and Control Systems

## 1. INTRODUCTION

Modelers throughout the U.S. defense community have, over the past several years, developed a great interest in the potential modeling and simulation uses of Petri Nets. Petri Nets were first proposed by C. A. Petri in 1962 in his doctoral dissertation, entitled *Kommunikation mit Automaten* (Reference 1). Petri's original purpose for the graphs which he developed was in the analysis of asynchronous concurrent computing systems.

Researchers soon realized that Petri Nets could be put to more general use. Their general utility is described by Peterson as "the modeling of systems of events in which it is possible for some events to occur concurrently but there are constraints on the concurrence, precedence, or frequency of these occurrences" (Reference 2). As this definition applies, in at least some way, to nearly every "system" in existence, the potential applications are many. Of particular interest to the U.S. Army Materiel Systems Analysis Activity (AMSAA) is their potential use in the simulation of command and control (C2) systems, which will be further addressed herein.

## 2. BACKGROUND

### 2.1 Stochastic, Timed, Attributed Petri Nets (STAPNs).

Petri Nets, in their basic form, are relatively simple networks. A Petri Net,  $C$ , may be represented either by a 4-tuple of set definitions (Figure 1a) or graphically (Figure 1b). Petri Nets are composed of a set of places, represented graphically by circles; a set of transitions, represented by bars; and the

$$C = (P, T, I, O)$$

$$P = \{p_1, p_2, p_3, p_4, p_5\}$$

$$T = \{t_1, t_2, t_3, t_4\}$$

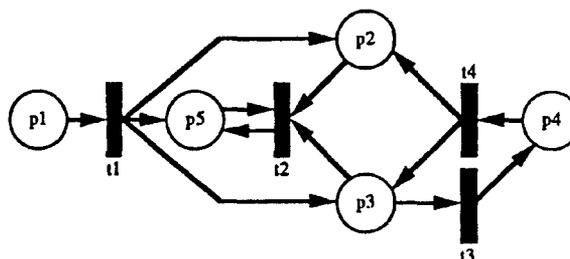
$$I(t_1) = \{p_1\} \quad O(t_1) = \{p_2, p_3, p_5\}$$

$$I(t_2) = \{p_2, p_3, p_5\} \quad O(t_2) = \{p_5\}$$

$$I(t_3) = \{p_3\} \quad O(t_3) = \{p_4\}$$

$$I(t_4) = \{p_4\} \quad O(t_4) = \{p_2, p_3\}$$

(a)



(b)

**Figure 1.** A Petri Net Structure Represented by: (a) A 4-tuple of Set Definitions; and (b) A Graph (from Reference 2)

input/output functions defining connections among them, shown by directed arcs. The graphs are bipartite, in that arcs are constrained to connect dissimilar nodes only (i.e., Place→Transition or Transition→Place).

The execution of a Petri Net is controlled by the existence of tokens (generic objects) in places. A Petri Net is executed by "firing" a sequence of transitions: a token is removed from every place which is an input to a transition, and a token is added to every place which is an output of that transition. It is required that all input places contain tokens, in order that a token may be removed from each input place (see Figure 2). This restriction on the firing of a transition is referred to as an enablement condition. No restrictions are placed upon the firing sequence among concurrently enabled transitions.

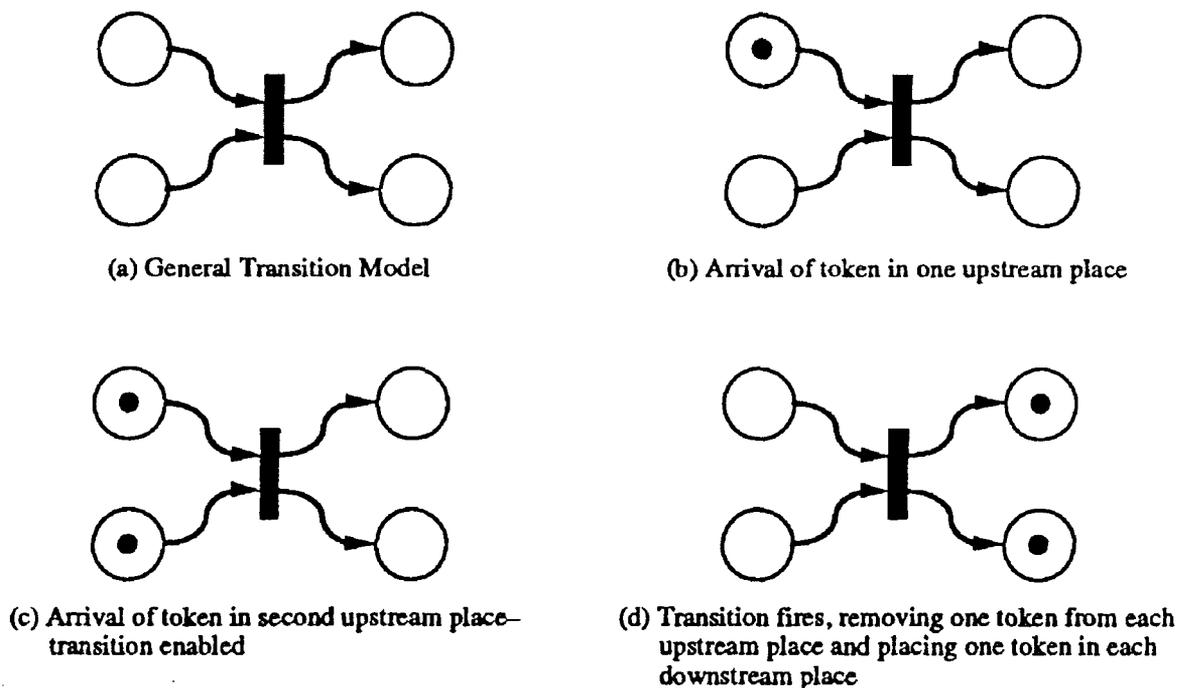


Figure 2. Petri Net Transition Firing (from Reference 3).

Petri's original nets (Ordinary Petri Nets) required that every place appear in the input and output functions of each transition at most once (i.e.,  $I(t_i)$  and  $O(t_i)$  must be proper sets). A more general form of Petri Net (General Petri Nets) was later introduced which did not have this restriction (i.e., places may appear in  $I(t_i)$  and  $O(t_i)$  more than once), and was shown to be equivalent (Reference 4). In this more general case, a transition is enabled only when each of its input places contains at least one token per Place→Transition arc.

Further extensions, many of which are incorporated into current Petri Net-based modeling software packages, further increase both convenience and modeling power. Most significant among these extensions is the addition of an "inhibit" arc, which permits the predication of transition enablement upon the emptiness of a place, or zero-testing. The addition of zero-testing increases the modeling power of a Petri Net to that of a Turing machine and, by extension, allows Petri Nets to model any system. The addition of inhibit arcs, however, reduces the decidability of Petri Nets (the ability to determine analytically the characteristics of a net) to zero, so it is not without cost (Reference 2).

The addition of the converse "enable" arc allows the enablement of a transition only when the population of a place is non-zero. Note that "enable" arcs do not increase the modeling power of Petri Nets; they are added for simplicity and convenience only. Neither "enable" nor "inhibit" arcs are considered inputs to transitions, in that no tokens are removed from the places to which they are connected.

Petri Nets become complete simulation tools with the addition of four other constructs: token attribution, random values and events, hierarchical clustering, and timing. General Petri Nets supplemented with the combination of random events, timing, and token attribution are referred to as Stochastic, Timed, Attributed Petri Nets (STAPNs).

By permitting the definition of attributes for tokens, a Petri Net tool allows the user to define distinct tokens (and token types) and to condition model execution on the attribute values of individual tokens (e.g., queue ranking and token equivalence testing). Attributed tokens are similar in nature to temporary entities in many simulation languages, and can be used to represent distinct resources (operators, devices, etc.) and customers (e.g., incoming messages to be processed).

Random values and events add the ability to define attributes of tokens and enablement conditions as subject to random--rather than strictly deterministic--events. This ability is extremely important in the modeling of command and control elements, particularly in the representation of human task processing.

Hierarchical clustering is an enhancement to the appearance of a Petri Net model, permitting sub-models to be clustered within "box" structures, much like the definition of a subroutine in a procedural language. Potentially unlimited levels of clustering--providing ever-increasing levels of detail--are possible. Although clustering does not enhance the modeling power of a Petri Net tool, it significantly improves the manage-

ability and clarity of models, facilitating the creation of much larger models than would be practical without clustering.

The concept of time is extremely important to simulation, but has no real meaning in General Petri Nets. In a non-timed Petri Net, the relative order in which events are allowed to occur is explicitly represented by the net structure, but the length of time required by an event cannot be represented. The addition of timing on transitions permits varying durations for events.

The most common means for implementing timing in Petri Net tools is the use of a discrete-event calendar, similar to those found in discrete-event simulation languages such as SIMSCRIPT II.5 and ModSim, both of which were developed by CACI Products Company. When a transition in a Petri Net tool fires, all input tokens are immediately removed, and the appearance of tokens in the output places is scheduled at some later point of simulated time. When all transitions enabled at a particular simulated time have fired, the simulation "clock" is advanced to the next point of simulated time appearing on the event calendar.

## 2.2 The AMSAA Simulation Technology (SIMTECH) Project.

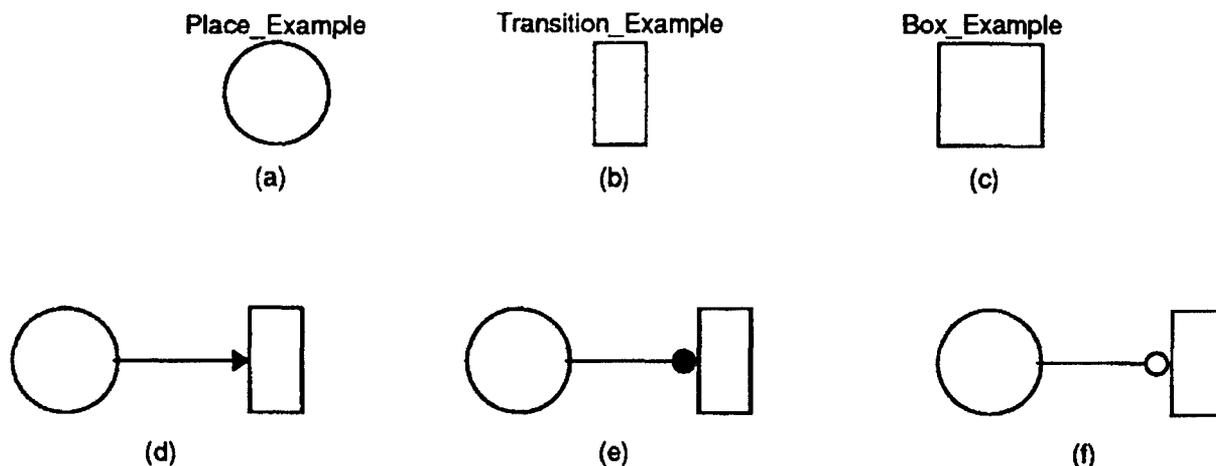
In March 1990, funding was granted to AMSAA in the amount of \$70,000 by the Army Model Improvement Program (AMIP) Management Office (AMMO) SIMTECH program, supporting the investigation of Petri Nets and their use--through existing Petri Net-based tools--in command and control modeling efforts. Efforts were focused in three major areas: verification of Modeler's simulation engine through comparison of Modeler results to the closed-form solutions of several queueing models; comparison of results generated by Modeler-based simulations to those of simulations written in procedural languages, such as SIMSCRIPT; and exploration of the overall usability of Modeler-based simulations for production studies. This project was continued by AMSAA on a self-funded, time-available basis from October 1990 through March 1994.

## 2.3 Development of Modeler.

The modeling tool used in this effort is Modeler, developed by Alphatech, Inc., under contract to the Air Force Systems Command Foreign Technology Division (FTD--now the National Air Intelligence Center (NAIC)). Prior to the development of Modeler, a "proof-of-principle" package, MicroModeler, was developed for use on Macintosh II series computers. MicroModeler supports the execution of General Petri Nets with the extensions of inhibit and enable arcs.

FTD supported several years of development effort by Alphatech to produce Modeler, which is currently hosted on Sun Micro-

system platforms (Sun4/SPARCstation). Modeler supports all features of STAPNs, as well as enable/inhibit arcs and hierarchical clustering. The graphical elements used to represent these modeled structures are shown in Figure 3. Along with these features, Modeler provides the user with built-in statistics capture facilities for basic performance parameters of individual transitions and places (e.g., mean population of a place, mean delay in a place, etc.).



**Figure 3.** Graphical Elements Used in Modeler:  
 (a) Place; (b) Transition; (c) Clustering Box; (d) Standard Arc;  
 (e) Enable Arc; (f) Inhibit Arc.

In 1992, the U.S. Army Training and Doctrine Command (TRADOC) Analysis Center (TRAC)-Operations Analysis Center (OAC--now the Studies and Analysis Center [SAC]) contracted with Potomac Systems Engineering (PSE) to develop several enhancements to the Alphatech version of Modeler. Many of the enhancements made proved quite useful, but a number of known bugs in the base version were not addressed.

Most recently, further improvements to Modeler have been funded by the U.S. Army Foreign Science and Technology Center (FSTC--now the National Ground Intelligence Center [NGIC]). Under the FSTC-funded effort, a version of Modeler which corrected most known bugs (Version 1.6) was released in December 1993. All results contained herein were generated using Version 1.6. A further enhanced version of Modeler (Version 2.1.1) was released for Department of Defense use in May 1994. Alphatech has concurrently developed a commercial version of Modeler, which is available for distribution in the United States and Canada, with further export licensing pending.

### 3. RESULTS OF AMSAA EFFORTS

#### 3.1 Validation of Modeler--Basic Queueing Models.

In order to validate Modeler, four basic queueing models for which properties can be determined in closed form were implemented, and their outputs checked against the known solutions. The four models implemented are:

- Poisson Arrivals/Exponential Service/Single-server ( $M/M/1$ )
- Poisson Arrivals/Exponential Service/Four-server ( $M/M/4$ )
- Poisson Arrivals/Erlang-2 Service/Single-server ( $M/E_2/1$ )
- Erlang-2 Arrivals/Exponential Service/Single-server ( $E_2/M/1$ )

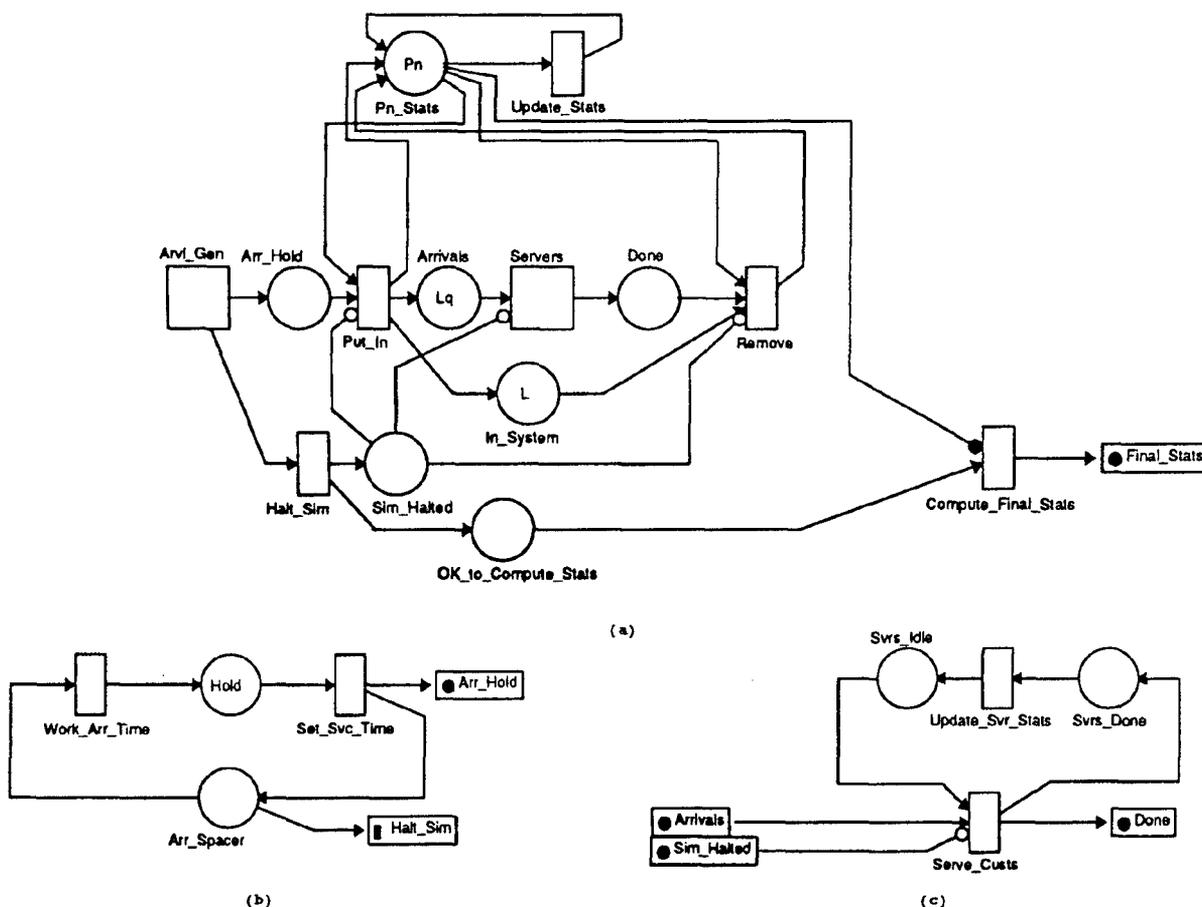


Figure 4. Modeler Structure Used in Queueing Model Verification: (a) Top-level Structure; (b) Arrival Generator; (c) Servers.

Figure 4 depicts the Modeler structure for these models, including the Generator and Server submodels. The  $M/E_2/1$  and

$E_2/M/1$  cases used slightly different structures, since the Erlang-2 distribution was generated using a sequence of two Exponential draws. In addition to these four basic models, a simple series queueing model was developed, incorporating a  $M/M/\infty$  (self-service) subsystem, the output of which was the input to a  $M/M/4$  subsystem. The Modeller structure of this series model is depicted in Figure 5.

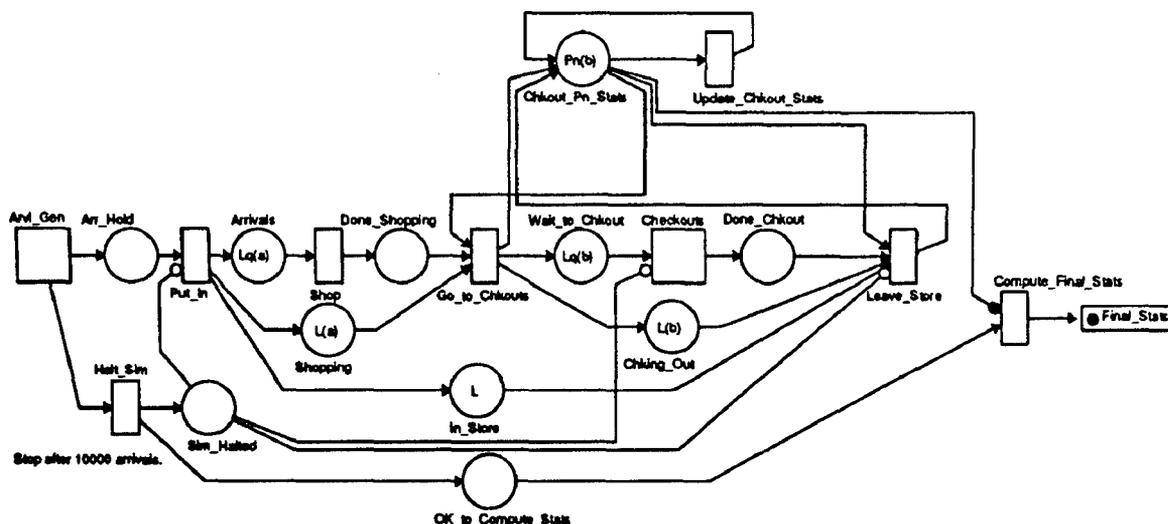


Figure 5. Modeller Structure for  $M/M/\infty \rightarrow M/M/4$  Series Queueing Model.

The model parameters of interest were: the mean number of customers in the system ( $L$ ), the mean number of customers waiting in the queue ( $L_q$ ), the mean time spent by customers in the system ( $W$ ), the mean time spent by customers in the queue ( $W_q$ ), and the expected proportion of time during which the system is empty ( $P_0$ ). For each of the four basic models described above, 20 replications of 3,000 arrivals each were executed using Modeller, and a 98 percent confidence interval for each of the five parameters was determined. Due to Bonferroni's Inequality (Reference 5), the resulting simultaneous confidence for the five parameters has a lower bound of 90 percent.

For the  $M/M/\infty \rightarrow M/M/4$  model, there were seven parameters of interest:  $L$  and  $W$  for the entire system, as well as  $L$ ,  $W$ ,  $L_q$ ,  $W_q$ , and  $P_0$  for the  $M/M/4$  subsystem. For this model, 20 replications of 10,000 arrivals each were executed using Modeller and a 98 percent confidence interval was determined for each of the seven parameters. Since seven parameters were being estimated, the Bonferroni Inequality yields a lower bound of 86 percent on the simultaneous confidence.

Table 1 provides a summary of the input parameters used for the five models studied. Table 2 presents a summary of the results obtained for the five models. As can be seen from the results shown, Modeler demonstrates a reasonable level of agreement with the expected results for these models. A minor bias seems to exist, since nearly all resultant means lie in the same direction relative to the theoretical values for the respective parameters. While the magnitude of this apparent bias is not of significant concern, it should be checked carefully in any analysis performed using Modeler. A thorough discussion and derivations of the expected results are presented in Reference 6. Further detail of the results obtained for the M/M/4 model is presented in Table 3.

**Table 1.** Input Parameters for Queueing Models.

Model	Mean Inter-Arrival Time ( $1/\lambda$ )	Mean Service Time ( $1/\mu$ )	Number of Servers ( $c$ )	Traffic Intensity ( $\rho=\lambda/c\mu$ )
M/M/1	4.5	3.5	1	0.778
M/M/4	4.5	14.0	4	0.778
M/E <sub>2</sub> /1	4.5	3.5	1	0.778
E <sub>2</sub> /M/1	4.5	3.5	1	0.778
<i>M/M/∞→M/M/4:</i>				
M/M/∞	1.5	40.0	∞	0.000
M/M/4	1.5	4.0	4	0.667

### 3.2 Petri Net Fire Support Communications Model (PN-FSCom).

The PN-FSCom was developed by the author during October 1991 as a rapid development exercise. PN-FSCom is based upon a model developed by Magnavox--C30MSIM, generally known as Comsim. Comsim was developed for the purpose of examining communications loading on networks supporting the Advanced Field Artillery Tactical Data System (AFATDS) in a division during a representative one-hour period. Provided as appendices to Magnavox's preliminary modeling report (Reference 7) were descriptions of all possible message threads which individual events could trigger, the laydown of units in one of the topologies modeled, and several of the time-ordered events lists (TOELs) which were used to drive the model. The message thread tree for one of the 21 possible events is shown in Figure 6.

**Table 2.** Queueing Model Performance Parameters (Theoretical Values vs. Modeler Results).

Model	Parameter	Theoretical Value	Modeler Results	
			Mean	98% C.I.
M/M/1	$P_0$	0.222	0.228	(0.217, 0.240)
	$L$	3.505	3.333	(3.119, 3.546)
	$L_q$	2.727	2.560	(2.355, 2.765)
	$W$	15.75	14.98	(14.11, 15.85)
	$W_q$	12.25	11.51	(10.65, 12.36)
M/M/4	$P_0$	0.032	0.033	(0.029, 0.037)
	$L$	5.060	4.883	(4.655, 5.110)
	$L_q$	1.949	1.805	(1.607, 2.003)
	$W$	22.77	21.99	(21.11, 22.88)
	$W_q$	8.77	8.12	(7.27, 8.96)
M/E <sub>2</sub> /1	$P_0$	0.222	0.226	(0.215, 0.236)
	$L$	2.819	2.783	(2.561, 3.005)
	$L_q$	2.042	2.010	(1.796, 2.223)
	$W$	12.69	12.48	(11.60, 13.36)
	$W_q$	9.19	9.00	(8.12, 9.87)
E <sub>2</sub> /M/1	$P_0$	0.222	0.222	(0.214, 0.231)
	$L$	2.699	2.680	(2.492, 2.867)
	$L_q$	1.922	1.901	(1.719, 2.083)
	$W$	12.15	12.01	(11.22, 12.79)
	$W_q$	8.65	8.52	(7.75, 9.29)
Series	$L$	33.44	32.89	(32.68, 33.11)
	$W$	50.14	49.90	(49.65, 50.15)
M/M/4 Only	$P_0$	0.060	0.061	(0.059, 0.063)
	$L$	3.44	3.73	(3.65, 3.81)
	$L_q$	0.76	1.07	(0.99, 1.15)
	$W$	5.14	5.17	(5.11, 5.24)
	$W_q$	1.14	1.17	(1.11, 1.22)

PN-FSCom was developed to fulfill two unrelated objectives. Since Comsim was not available to AMSAA, it was desired that a "clone" of Comsim be produced to permit verification of the Magnavox results and performance of excursion analyses. Additionally, the development of PN-FSCom was used to evaluate the

**Table 3.** Detailed Modeler Results--M/M/4 Queueing Model.

Replication	$L$	$L_q$	$W$	$W_q$	$P_0$
1	4.91	1.82	21.75	8.05	0.039
2	5.54	2.39	24.32	10.48	0.028
3	5.18	2.11	23.59	9.60	0.037
4	4.51	1.60	20.56	7.29	0.048
5	4.75	1.67	21.78	7.65	0.024
6	5.43	2.23	23.68	9.71	0.028
7	5.00	1.92	22.14	8.51	0.030
8	4.74	1.73	21.35	7.82	0.037
9	4.52	1.48	20.89	6.85	0.035
10	4.25	1.22	19.41	5.54	0.035
11	5.23	2.00	23.20	8.87	0.022
12	4.10	1.07	18.96	4.95	0.032
13	5.14	2.09	23.81	9.67	0.041
14	4.71	1.59	21.13	7.13	0.031
15	4.75	1.67	21.82	7.66	0.024
16	4.60	1.64	21.07	7.52	0.041
17	5.22	2.13	23.41	9.53	0.036
18	4.51	1.50	20.14	6.68	0.032
19	5.32	2.19	23.51	9.67	0.033
20	5.24	2.05	23.36	9.14	0.020
Mean	4.883	1.805	21.994	8.116	0.033
Variance	0.160	0.121	2.447	2.199	0.000
Half-Length	0.227	0.198	0.888	0.842	0.004
98% Lower Bound	4.655	1.607	21.106	7.274	0.029
98% Upper Bound	5.110	2.003	22.882	8.958	0.037

utility of Modeler as a rapid development tool. The initial version of PN-FSCom was built and executed, using only the descriptions given in Reference 7, in approximately seven working days. The top level structure of PN-FSCom is depicted in Figure 7.

When the model was first executed, significant differences were noted between the throughput characteristics of PN-FSCom and Comsim. To determine if a modeling error had been made, the structure of PN-FSCom was thoroughly traced, and the expected numbers of messages generated were calculated, for the baseline (stochastic, twenty replications) case and three deterministic

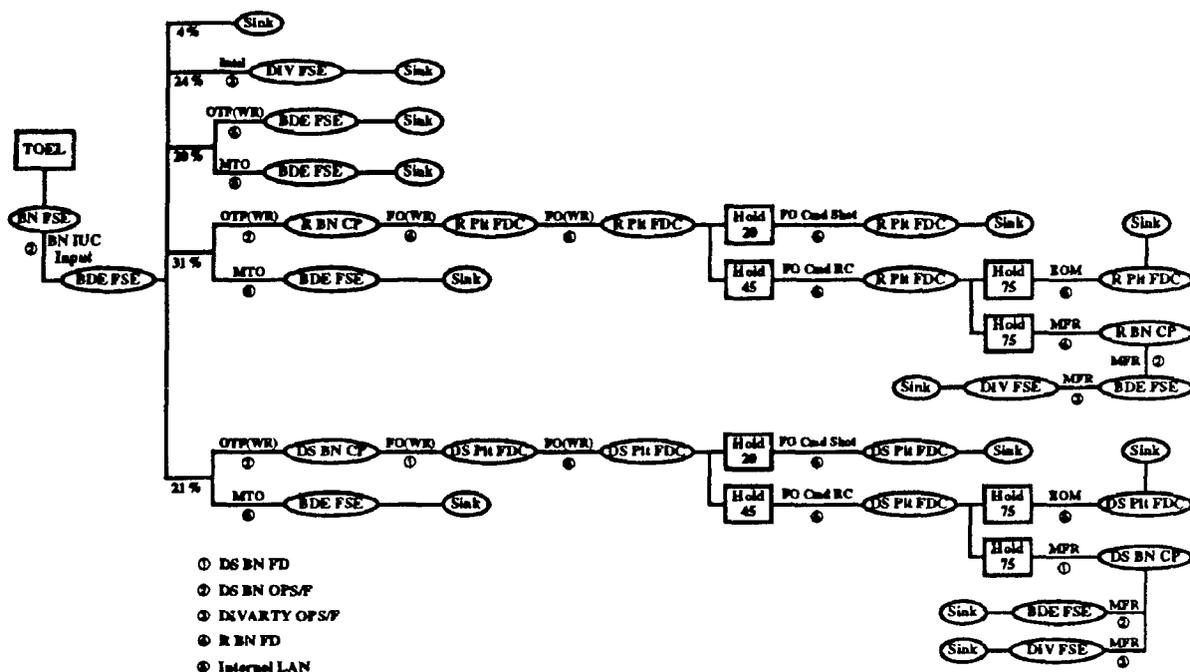


Figure 6. Cossim "Battalion Independent User Center (BN IUC) Input" Thread Diagram (from Reference 7).

Table 4. PN-FSCom Messages Generated: Expected vs. Observed (No Time Limit).

Net Name	Messages Generated							
	Stochastic		Det. Case 1		Det. Case 2		Det. Case 3	
	Exp.	Obs.	Exp.	Obs.	Exp.	Obs.	Exp.	Obs.
DIVARTY OPS/F	639	604	140	140	750	750	870	870
DS BN OPS/F	1489	1491	624	624	1844	1844	1964	1964
DS BN FD	1355	1388	623	623	638	638	4558	4558
R BN FD	466	466	0	0	1400	1400	0	0
HVY MORT FD	188	176	15	15	2175	2175	15	15
Total	4137	4125	1402	1402	6807	6807	7407	7407

Stochastic: All decision probabilities as specified in Reference 7.  
 Det. Case 1: Minimize message traffic on all nets.  
 Det. Case 2: Maximize traffic on R BN FD and HVY MORT FD  
 Det. Case 3: Maximize traffic on DS BN FD

cases. Some minor errors in the model structure were found and corrected. Table 4 presents the results of the four test cases. For the purpose of generating the results given in Table 4, the

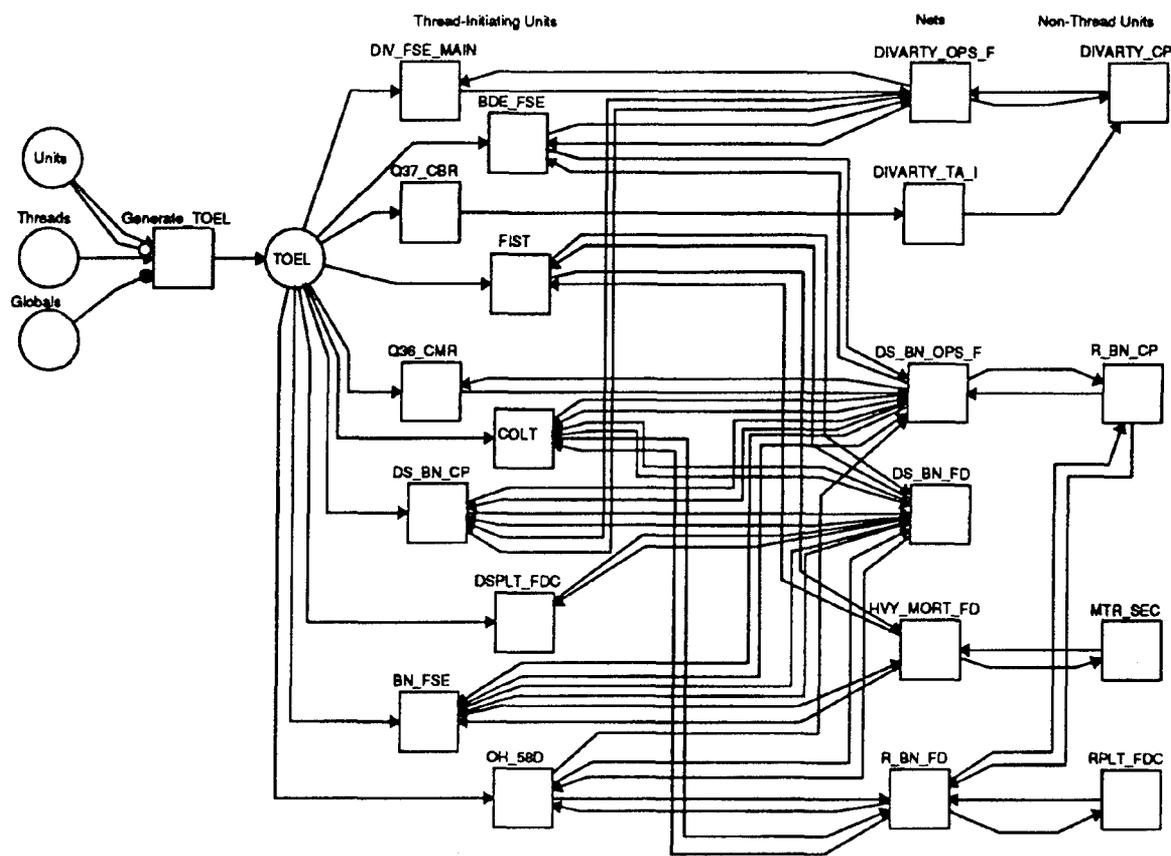


Figure 7. PN-FSCom Top-Level Structure.

model was permitted to execute until all threads had been completed, rather than halting execution at precisely 1.00 hours, as it is in the "production" case. It can be seen from the data in Table 4 that the number of messages generated exactly matched the number expected for all three deterministic cases. For the stochastic case, the numbers generated were within seven percent of the expected values for individual nets, and within one percent of the expected total. These results strongly indicate that, after correcting the few errors found, the model performs as intended.

After the model had been corrected and the test cases had been checked, the "production" case was re-run. The results of this run are compared with the results given by Magnavox (Reference 7) in Table 5. The data given indicate that PN-FSCom's results still differed somewhat from those generated by Comsim, but their agreement was significantly improved (e.g., the total number of messages processed differed by less than four percent). It is likely that the remaining differences in observed performance arise from either of two sources. First, it is possible that other assumptions were made in the original development which were not defined in Reference 7. Further, due to a problem

**Table 5. Network Performance Statistics: Comsim vs. PN-FSCom (Modeler, 20 replications).**

Net Name	Model	Msgs. Passed		Mean Waiting Time (Seconds)	
		Pri. 1	Pri. 2	Pri. 1	Pri. 2
DIVARTY OPS/F	Comsim	158	338		
	PN-FSCom	147.0	298.6	0.89	2.12
DS BN OPS/F	Comsim	448	676		
	PN-FSCom	427.1	711.3	10.76	149.44
DS BN FD	Comsim	293	137		
	PN-FSCom	281.0	132.7	1.03	1.24
R BN FD	Comsim	83	52		
	PN-FSCOM	78.5	49.6	0.39	0.44
HVY MORT FD	Comsim	66	0		
	PN-FSCom	53.0	2.3	0.72	0.37
Overall	Comsim	1048	1203	62.4	212.9
	PN-FSCom	986.6	1194.5	3.21	68.62

with the original code design, it was not possible to vary the random number seed used by Comsim. Therefore, all Comsim data represent the results of only one replication.

After completion of the baseline work described above, a number of excursion runs were performed to evaluate the impact on throughput of alternative network laydowns. The results of these excursions indicated that a number of alternatives existed which could significantly reduce network congestion without equipment replacement or doctrinal changes. These results were shared with the AFATDS Program Office for their consideration.

### 3.3 All-Source Analysis System (ASAS) Network Model (ASASNET).

In March 1993, AMSAA became involved in the "certification" of ASASNET at the request of the Deputy Undersecretary of the Army for Operations Research (DUSA-OR). ASASNET simulates the time required at division and corps intelligence elements to process incoming tasks using the ASAS. In support of the Cost and Operational Effectiveness Assessment (COEA) for ASAS and other performance analyses, ASASNET was developed and executed in-house by TRAC-OAC personnel using the version of Modeler produced by PSE.

The evaluation of ASASNET focused on two areas: verifying ASASNET against its existing design documentation; and assessing the validity of the measures of performance generated by ASASNET, given the assumptions under which it was developed.

Several of the assumptions made in the initial design of ASASNET were found to be inappropriate and to diminish the validity of the results produced. In particular, ASASNET operated from a scripted list of task handling requirements (e.g., generate 25 reports of one type per hour, process 32 incoming reports of another type per hour, etc.) and these requirements were assumed to have no relation to one another. It was further assumed that all tasks could be represented by an average number of bytes processed per task, which was then sub-divided into a set of tokens each representing 500 bytes of processing. The time required to process a "token" at a workstation varied in accordance with the equipment at that workstation, but all times were assumed to have triangular distributions with fixed parameters (minimum, mode, and maximum).

The use of a sequence of tokens to represent a lengthy task particularly reduced the validity of ASASNET's results since, in many cases, groups of related workstations shared an input place in the model and no provision was made in the model to prevent multiple workstations from processing tokens representing a single task. This also presented a problem when task statistics were desired, since Modeler provides useful information only for tokens. Given the problem described above, no valid procedure could be developed to re-aggregate token statistics into useful task statistics.

During the initial study performed by TRAC-OAC, a number of excursion runs were performed, wherein individual tokens were arbitrarily discarded at varying rates until the timeliness of processing for those which remained was considered acceptable. This further reduced the validity of ASASNET's results since the discarded tokens represented only fractions of tasks.

In addition to the validation concerns described above, a small number of data entry errors were found in the model and brought to the developer's attention.

As a result of the evaluation performed, TRAC-OAC was provided with written recommendations for improving ASASNET's validity (Reference 8). In particular, it was recommended that the representation of task requirements be modified from a sequence of  $n$  tokens each of which is processed in time  $t_i \sim \text{Triang}(a, b, c)$  to a single token which is processed in time  $t \sim \text{Triang}(na, nb, nc)$ . Although this would not produce exactly the same distribution for the total task time, there was no empirical basis for the original use of a sum of Triangular distributions,

so it is reasonable to consider the resulting distribution to be no less valid. Further, the use of single tokens to represent tasks permitted the built-in statistics gathering facilities of Modeler to generate task statistics directly. An additional side benefit was a significant reduction in execution time, since a fraction of the original number of tokens required processing by the simulation engine.

### 3.4 The Fire Support Command and Control Analysis Tool (FISCCAT).

Efforts to represent an existing production simulation using Petri Nets focused on AMSAA's FISCCAT. FISCCAT is a simulation, written in SIMSCRIPT II.5, of a brigade slice of the fire support functional area. This simulation was originally developed in support of the COEA effort for AFATDS. FISCCAT is driven by an externally-generated TOEL providing a sequence of targets of opportunity. These target events cause the initiation of fire requests, which are processed by the simulation, resulting in either a target engagement or the rejection of the request.

Originally, versions of FISCCAT were separately developed to represent the functional processing of both AFATDS and the Tactical Fire Direction System (TACFIRE), which AFATDS is intended to replace in the field. AMSAA performed a study comparing the abilities of TACFIRE and AFATDS to support a variety of mean target arrival rates, as well as a representative one-day scenario. The results of this study are contained in Reference 9.

In 1991, a verification and validation (V&V) effort was completed for Version 1.0 of FISCCAT. It is this version upon which the Modeler efforts were based. Development of Version 2 of FISCCAT is currently underway, with completion expected by the end of Fiscal Year 1994. The new version of FISCCAT will be used in AMSAA's analyses in support of the next AFATDS COEA and other AFATDS analysis efforts. Among the anticipated features of Version 2 will be the ability to represent a force structure containing a mixture of AFATDS and TACFIRE nodes utilizing a variety of communications media.

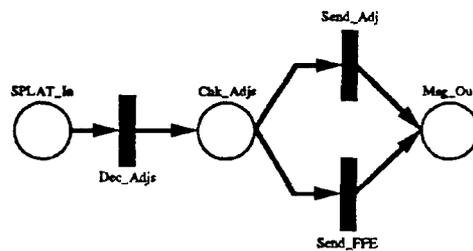
Prior to initiation of the Petri Net implementation, the source code for FISCCAT was completely reviewed, formatted and commented to facilitate translation and future maintenance efforts. This review included corrections as required to bring the source into compliance with recommended programming practices (e.g., elimination of GOTO statements).

From this formatted code, a Modeler representation can be developed almost directly. Differences in representation occur where global data structures are utilized in the SIMSCRIPT version. The problem with global data structures in Modeler will be further addressed below. A sample segment of the FISCCAT

```

ROUTINE PROCESS.FO.ROUNDS.LANDED given FO
...
REMOVE THE FIRST MESSAGE FROM THE FO.ROUNDS.LANDED.QUEUE(FO)
...
SELECT CASE MSG.TYPE(REC.MSG)
CASE "SPLAT"
  SUBTRACT 1 FROM FM.NUM.REMAIN.ADJ(FM)
  CREATE A MESSAGE CALLED MSG
  LET MSG.FM.PNTR(MSG) = FM
  LET MSG.ORIG.TYPE(MSG) = "FO"
  LET MSG.ORIG.PNTR(MSG) = FO
  IF FM.NUM.REMAIN.ADJ(FM) NE 0
    WORK PREP.TIME .SECONDS
    LET MSG.TYPE(MSG) = "ADJ"
    LET MSG.CLASS(MSG) = .FM
    LET MSG.TT(MSG) = FOC.ADJ.TT(TYPE)
    LET MSG.PRIORITY(MSG) = 1
  ELSE
    WORK PREP.TIME .SECONDS
    LET MSG.TYPE(MSG) = "FFE"
    LET MSG.CLASS(MSG) = .FM
    LET MSG.TT(MSG) = FOC.FFE.TT(TYPE)
    LET MSG.PRIORITY(MSG) = 1
  ALWAYS
CASE "SPLAT(RC)"
...
ENDSELECT
IF MESS.FLAG EQ .ON
  IF FM.CM.PNTR(FM) EQ 0      " SEND MESSAGE TO FIST
    LET MSG.RECIP.TYPE(MSG) = "FIST"
    LET MSG.RECIP.PNTR(MSG) = FO.FIST.PNTR(FO)
    LET MSG.NET.PNTR(MSG) = SYS.SUB.TBL(FO.NUM(FO),
      FIST.NUM(FO.FIST.PNTR(FO)), MSG.CLASS(MSG))
    LET MSG.PROB.SUCCESS(MSG) = SYS.PROB.SUCCESS.TBL(.FO,.FIST)
  ELSE      " MISSION FIRED BY MORTARS SO SEND DIRECT
  ...
  ALWAYS
  FOR EACH NET.UNIT IN NET.UNIT.LIST(MSG.NET.PNTR(MSG))
    WITH NET.UNIT.NAME = FO.NAME(FO)
    FIND THE FIRST CASE
  IF FOUND
    LET MSG.NADT.L(MSG) = NET.UNIT.NADT.HL
    LET MSG.NADT.NL(MSG) = NET.UNIT.NADT.HNL
    LET MSG.HOLD.TIME(MSG) = NET.UNIT.HOLD.TIME
  ALWAYS
  CALL PRINT.MESSAGE GIVEN MSG
  FILE MSG IN THE FOD.SEND.MSG.QUEUE(DEV)
  IF FOD.BUSY.FLAG(DEV) = .IDLE
    LET FOD.BUSY.FLAG(DEV) = .BUSY
    REACTIVATE THE FO.DEVICE CALLED DEV NOW
  ALWAYS
  ALWAYS
  DESTROY THE MESSAGE CALLED REC.MSG
  RETURN
END      " PROCESS.FO.ROUNDS.LANDED

```



(a)

(b)

**Figure 8.** A Segment of FISSCAT Code in (a) SIMSCRIPT; and (b) Modeler.

source code is shown in Figure 8a, and the corresponding Modeler representation is shown in Figure 8b. Initially, the Modeler representation, referred to as PN-FISCCAT, includes only a subset of the units represented by FISCCAT (maneuver battalions, controlling several sensors and one mortar company each). Examples of the structural components of PN-FISCCAT are given in Figures 9 through 11. Figure 9 depicts the top-level structure of PN-FISCCAT, Figure 10 depicts the structure of the Forward Observer (FO) node and Figure 11 one function within the FO.

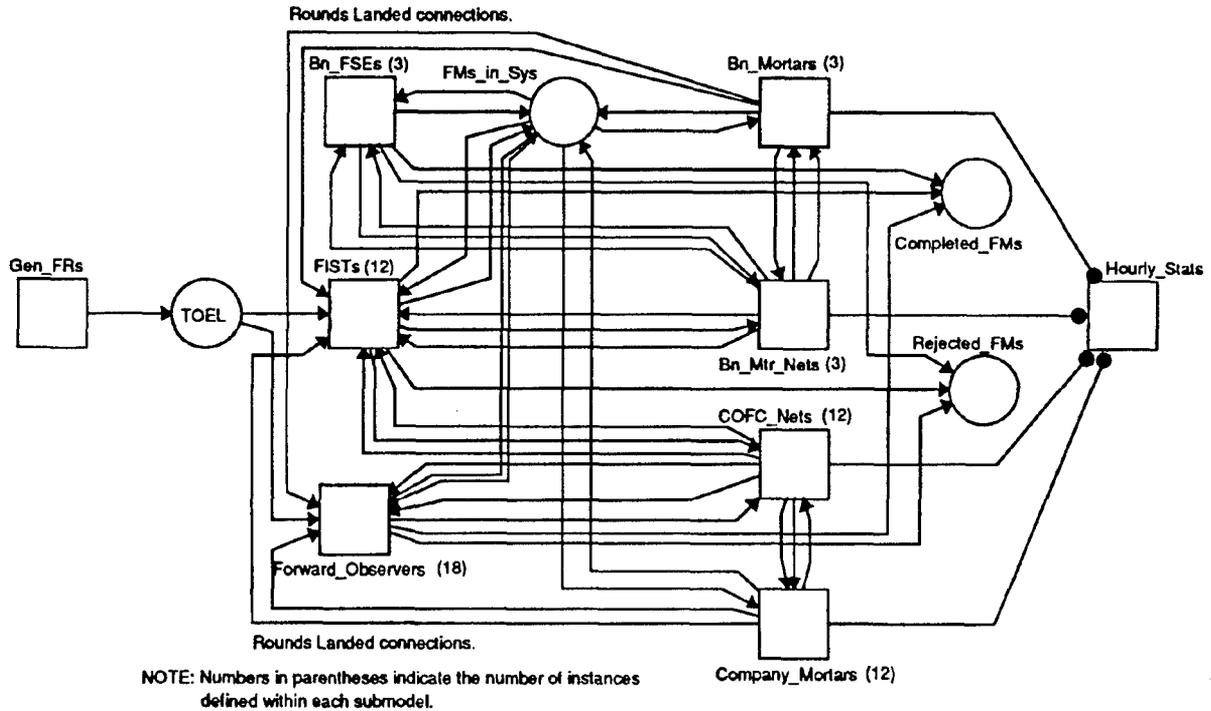


Figure 9. PN-FISCCAT Top-Level Structure.

It should be noted that no effort has been made to comply with the model's functional requirements document (Reference 10) in the course of this translation (version 2 of FISCCAT is expected to comply with more than 90 percent of these requirements, while version 1.0 complies with less than 40 percent). Rather, emphasis was placed upon the ability of Modeler to produce identical results to those produced by the same model developed using a procedural simulation language. Due to a lack of resources to support further efforts, development work on PN-FISCCAT was discontinued prior to completion, so no results are available.

#### 4. LIMITATIONS OF MODELER

There remain two significant aspects of Modeler which limit its utility as a production modeling tool. First, the use of

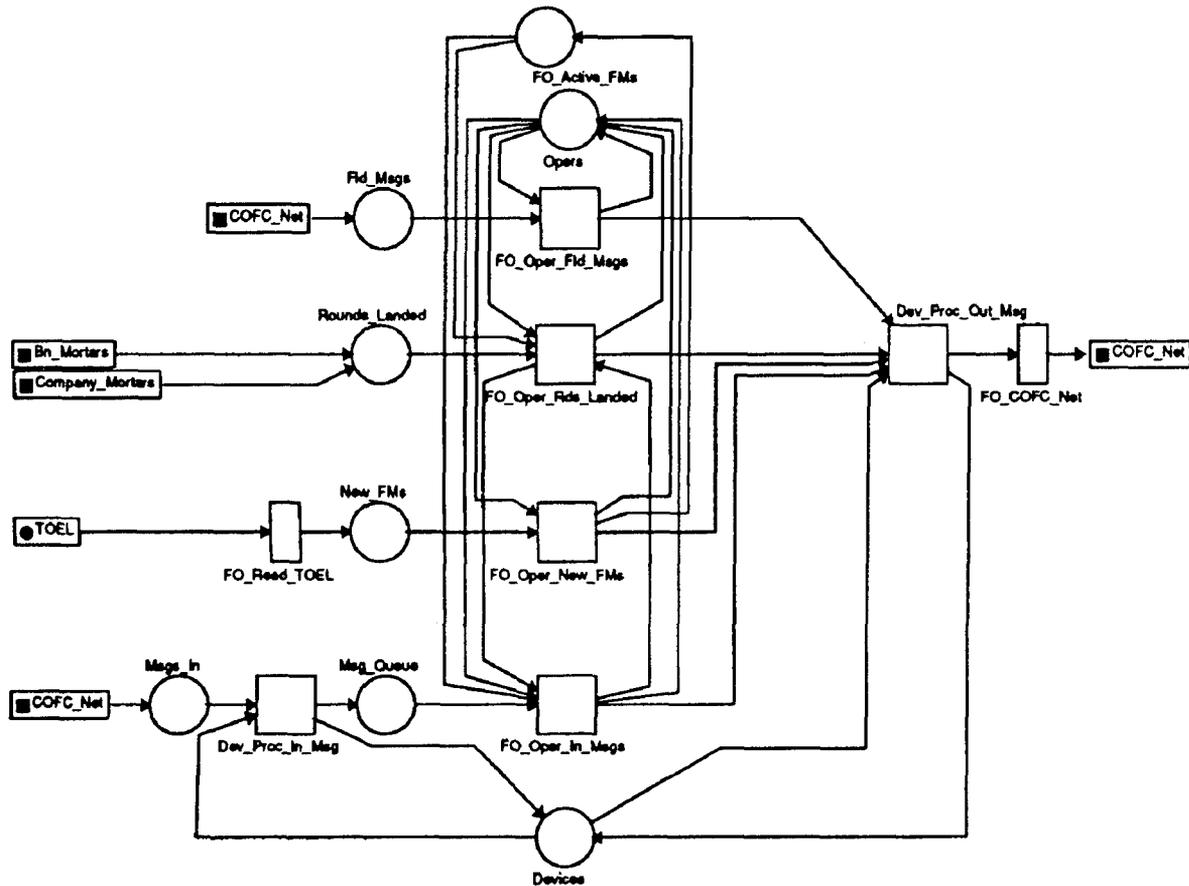


Figure 10. PN-FISCCAT: Forward Observer (FO) Structure.

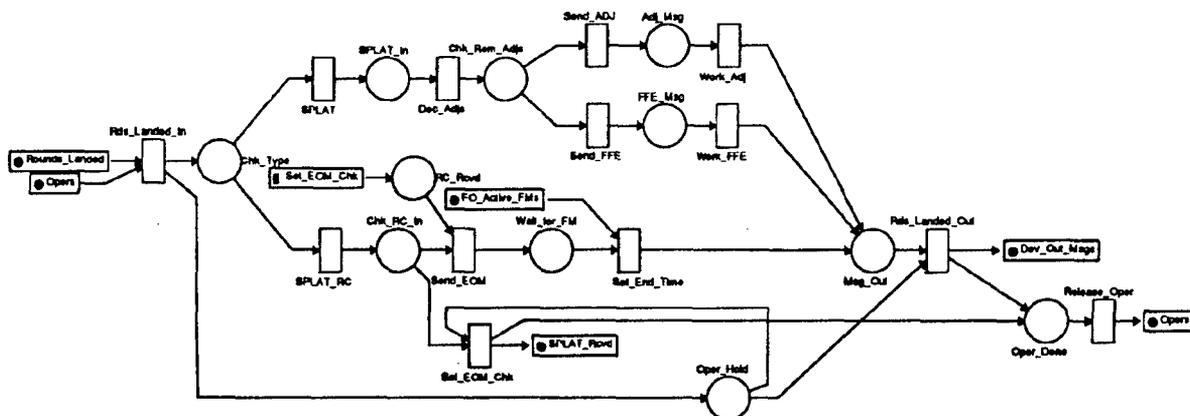


Figure 11. PN-FISCCAT: FO\_Oper\_Rds\_Landed Function.

global data structures (particularly model-wide statistics collection) is extremely cumbersome in Modeler, due to the fact that any process (transition) which is required to change a global data set (add, remove, or modify an element) must have at least one arc connecting it to the place containing that data

set. For example, in FISCCAT, a global list of active fire missions is used widely throughout the model. In PN-FISCCAT, this would require the addition of at least 100 additional arcs. This problem is worked around by adding the attributes of associated fire missions to the message tokens which are passed within the model.

Another facet of this global-data problem arises as a result of the fact that Modeler does not provide the capability to perform run-time file input/output. Summary output data (e.g., one-line records of timestamps associated with individual fire missions) must be retained within the model and written to a file when execution is completed. Similarly, all input records must be read and stored within the model prior to execution. For large models and/or long runs, the processing and memory burden of retaining these data during execution can be quite substantial.

The second negative aspect of Modeler is its poor execution speed. Since Modeler utilizes an X Window System-based user interface, all changes to any visible portion of a model require calls to the X server, which generally redraws the entire window with each call. This behavior presents a burden to Modeler execution speed, since transitions are highlighted as they are fired. This aspect of the speed problem can be mitigated by ensuring that no transitions appear in the top level of the model and exposing only that top level during model execution. It is also possible that future versions of Modeler will include a "batch" option to eliminate the graphics altogether during execution.

Execution speed is also hampered by the use of tokens containing a large number of attributes. Unfortunately, the global data set work-around described above has the direct result of increasing the number of attributes defined for the most commonly used tokens. When designing a Modeler-based simulation, these two considerations must be carefully balanced against one another to prevent excessive loss of execution speed.

Other factors were also observed to have a negative impact on execution speed. First, when the number of tokens in places would become large, execution would slow considerably, indicating that the list maintenance facilities of the Modeler engine are not well optimized. Second, execution time was observed to be adversely impacted by increases in the proportion of calculations which require floating-point operations. In order to mitigate this effect, care should be taken in the selection of the base time unit and other measurement scales such that the use of floating-point values is minimized.

## 5. OTHER ARMY PETRI NET EFFORTS

### 5.1 TRAC-OAC/SAC.

In addition to the ASASNET effort discussed above, a number of other Petri Net models have been developed by or for personnel at TRAC-OAC (now TRAC-SAC). The first, and perhaps best known, of these models is the Command and Control Network Model (C2NET), which was developed under contract by PSE. C2NET represents the processing performed within corps, division and brigade command posts in the intelligence and fire support functional areas. It was developed over the period from August 1990 through September 1991, to support completion of the Command and Control Responsiveness Analysis (Reference 11). C2NET continues in use to the present, and has been modified and enhanced by TRAC-SAC personnel as requirements have evolved.

A high-resolution model of a division command post (DIV CP) was developed during the period from September 1990 through April 1992, by Alphascience, Inc., also under contract to TRAC-OAC. This model was used to evaluate the relative effectiveness of the conceptual Functional Command Post (1990) structure and several possible alternative structures in supporting the C2 functions in a heavy division (Reference 12).

Currently, TRAC-SAC personnel are using Modeler in the in-house development of a number of models in the C2 functional area. These include: a functional model of a Corps Main command post, similar to DIV CP; the Army Battle Command System (ABCS) model; improvements to a C2NET-derived model of the Combat Service Support Control System (CSSCS); and a model for use in the evaluation of prepositioned logistics methodology. Functional descriptions of these models were not available at the time of this writing.

### 5.2 FSTC/NGIC.

Personnel at FSTC (now NGIC) became interested in the use of Modeler in support of their simulation efforts early in Fiscal Year 1992. A contractual effort was recently completed by Alphatech which included a number of further enhancements to Modeler. Along with the Modeler enhancement efforts, Alphatech has developed a baseline artillery Command, Control and Communications (C3) model, which is currently undergoing testing. Upon completion of testing, this model will be used to perform artillery C3 timeline analyses, utilizing system data contained in the Initial Network (INNET) database.

The INNET database contains a great deal of information on the functional and technical characteristics of threat C3 equipment in many functional areas. However, INNET does not contain information related to network throughput or C3 timelines. It is

this gap which the current artillery C3 model and subsequent functional area models are intended to fill.

At present, NGIC personnel are evaluating the utility of Modeler in meeting their simulation requirements. A decision will be made in the near future regarding the methodology to be used in developing subsequent C3 functional area models.

## 6. THE FUTURE OF MODELER AT AMSAA

Due to the fact that Modeler's suitability as a production simulation tool is still quite limited, AMSAA does not intend to utilize Modeler as the production environment for any near-term simulation development efforts. However, Modeler has proven extremely useful as an adjunct tool during simulation development, providing an excellent facility for visually developing and debugging algorithms and program structures. These algorithms and structures can then be translated into the analyst's programming language of choice. Also, since first-order models can be developed and debugged quickly, Modeler is quite useful as a tool in developing "proof-of-principle" demonstration models. Such demonstration models can be developed easily and provide a convenient visual means for conveying the basics of proposed designs.

As enhancements to Modeler continue, AMSAA will continue to monitor its maturation. When Modeler has further matured and its execution speed has been sufficiently improved, consideration will again be given to its potential as a production simulation tool. Until such time, Modeler will see continued use as a design/debugging tool, as well as in the development of "proof-of-principle" first-order models.

## 7. CONCLUSIONS

The general conclusion resulting from this effort is that Petri Nets are most appropriate as a modeling vehicle when the modeled system can be decomposed into subsystems with minimal, well-defined interfaces to each other. This is primarily a result of the global data problem discussed above. Any system in which the use of global data or complex or variable interfaces cannot be avoided is not easily modeled using Petri Nets.

The fact that Petri Nets best represent systems of interacting subsystems makes them highly appropriate for C2 systems, since individual nodes in a real C2 network have access only to data which is local to the node or sent to it via some communications medium. Also suggested as a possible application of Petri Nets is a system which contains a number of components utilizing a fixed set of shared resources (such as a MIL-STD-1553 bus) to obtain data and interact with other components. The global data

problem still presents difficulties, however, since it makes system-wide data collection extremely cumbersome.

The basic elements of Modeler have been found, through the performance validation conducted as part of this effort, to provide valid simulations of simple queues and processes. Therefore, given that sufficient attention is paid to interaction effects in model design, Modeler would appear to provide a valid simulation tool for arbitrarily complex queue/process systems.

The use of Modeler as a tool for the development and execution of models does currently have those drawbacks mentioned above, but there are two principal advantages over a procedural language approach. The first of these advantages is that all interactions among processes are graphically depicted to the developer at all times. Only those interactions explicitly "drawn in" are permitted to occur, preventing problems which frequently arise in procedural codes due to obscure interactions or mis-referenced pointers.

The second, and perhaps more important, advantage seen is a significant reduction in debugging time, since changes take relatively little time and can be executed immediately. These two advantages become increasingly important as a model increases in age and/or complexity, and as the personnel responsible for model maintenance and modification change. Despite the problems noted, these advantages make Petri Nets in general, and Modeler in particular, a potential long-term winner for Army C2 modeling efforts.

## REFERENCES

1. Petri, Carl Adam, "Communication with Automata," in *Supplement I to Technical Report RADC-TR-65-377*, U.S. Air Force Systems Command, Rome Air Development Center, January 1966. Translated by Clifford F. Greene, Jr.
2. Peterson, James L., *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1981.
3. *Modeler I User's Manual, Version 2.1.1*, Alphatech, Inc., Burlington, MA, 31 March 1994. TR-623-1
4. Hack, M., "Decision Problems for Petri Nets and Vector Addition Systems," in *Technical Memo 59, Project MAC*, Massachusetts Institute of Technology, March 1975.
5. Law, Averill M. and W. David Kelton, *Simulation Modeling & Analysis*, McGraw-Hill, Inc., New York, NY, 1991.
6. Gross, Donald and Carl M. Harris, *Fundamentals of Queueing Theory*, John Wiley & Sons, Inc., New York, NY, 1985.
7. Whisler, N.L., *Preliminary Report, AFATDS Performance Modeling, System Level, for Multi Service Advanced Field Artillery Tactical Data System*, AFATDS System Engineering, Systems Technology Division, Magnavox Government & Industrial Products Company, 2 January 1991.
8. *Memorandum, U.S. AMSAA, Subject: Review of ASASNET Model*, 22 July 1993.
9. Wagstaff, Kevin, *Technical Report 467, Modeling and Analytical Support for the Advanced Field Artillery Tactical Data System (AFATDS) Cost and Operational Effectiveness Analysis (COEA)*, U.S. Army Materiel Systems Analysis Activity, August 1988.
10. Ward, Patrick, "Appendix A, Fire Support Command and Control (FSCC) Baseline System's Modeling Requirements," in *Verification and Validation of the Initial Fire Support Command and Control Analysis Tool (FISCCAT) (Draft)*, U.S. Army Materiel Systems Analysis Activity, March 1991.
11. Rameden, David, "Command and Control Responsiveness Analysis," (Technical Report: TRAC-F-TR-0591), TRAC-Operations Analysis Center, September 1991. Volume II: Supporting Analysis (SECRET/NOFORN)

REFERENCES (Continued)

12. Bornman, Louis G. Jr. and Kelley E. Stephens, "Division Command Post (DIV CP) Analysis," (Technical Report: TRAC-TR-0192), TRAC-Operations Analysis Center, April 1992. Volume I: Executive Summary (UNCLASSIFIED)

## LIST OF ACRONYMS

ABCS	Army Battle Command System
AFATDS	Advanced Field Artillery Tactical Data System
AMIP	Army Model Improvement Program
AMMO	Army Model Improvement Program (AMIP) Management Office (now MISMA)
AMSAA	U. S. Army Materiel Systems Analysis Activity
ASAS	All-Source Analysis System
ASASNET	All-Source Analysis System (ASAS) Network Model
BN IUC	Battalion Independent User Center
C2	Command and Control
C2NET	Command and Control Network Model
C3	Command, Control and Communications
COEA	Cost and Operational Effectiveness Assessment
CSSCS	Combat Service Support Control System
DIV CP	Division Command Post
DUSA(OR)	Deputy Undersecretary of the Army for Operations Research
FISCCAT	Fire Support Command and Control Analysis Tool
FO	Forward Observer
FSCC	Fire Support Command and Control
FSTC	U.S. Army Foreign Science and Technology Center (now NGIC)
FTD	U.S. Air Force Foreign Technology Division (now NAIC)
INNET	Initial Network Database
MISMA	US Army Model Improvement and Study Management Agency
NAIC	National Air Intelligence Center
NGIC	National Ground Intelligence Center
OAC	Operations Analysis Center
PN-FISCCAT	Petri Net implementation of FISCCAT
PN-FSCom	Petri Net Fire Support Communications Model
PSE	Potomoc Systems Engineering
SAC	Studies and Analysis Center
SIMTECH	Simulation Technology
STAPN	Stochastic, Timed, Attributed Petri Net
TACFIRE	Tactical Fire Direction System
TOEL	Time-Ordered Events List
TRAC	TRADOC Analysis Center

LIST OF ACRONYMS (Continued)

TRADOC	U. S. Army Training and Doctrine Command
V&V	Verification and Validation

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>
7	Commander Defense Technical Information Center Attn: DTIC-FDAC Cameron Station, Bldg: 5 Alexandria, VA 22304-6145
2	Director US Army Model Improvement and Study Management Agency Attn: L. McGlynn Crystal Square 2, Suite 808 1725 Jefferson Davis Highway Arlington, VA 22202
2	Commander National Ground Intelligence Center Attn: IANGIC-RBE (J. Morrow) 220 Seventh Street NE Charlottesville, VA 22902
1	US Army Training and Doctrine Command Director, TRAC-SAC Attn: ATRC-SAS Ft. Leavenworth, KS 66027-5200
1	Commander National Air Intelligence Center Attn: TACA (T. Mullins) 4115 Hebble Creek Road (Suite 25) Wright-Patterson AFB, OH 45433-5629
1	Commander Simulation, Training, and Instrumentation Command Attn: AMSTI-EC (G. Schow) 12350 Research Parkway Orlando, FL 32826-3276
7	Director US Army Materiel Systems Analysis Activity Attn: AMXSY-CA (3 cys) AMXSY-SL (2 cys) AMXSY-RI Aberdeen Proving Ground, MD 21005-5071

The next page is blank.

AMXSY-CA

## STUDY GIST

SUBJECT: AMSAA TR-559; Use of Petri Nets in the Simulation of Command and Control Systems

### OBJECTIVE:

The principal objective of this study was to evaluate the utility of Petri Nets, and simulation tools based upon Petri Nets, in the simulation of military command and control systems.

### BASIC APPROACH:

The approach taken in this study was comprised of three major efforts:

- Verification of Modeler's simulation engine through comparison of Modeler results to the closed-form solutions of several queueing models
- Comparison of results generated by Modeler-based simulations to those of simulations written in procedural languages
- Exploration of the overall usability of Modeler-based simulations for production studies

### PRINCIPAL LIMITATIONS:

The efforts in this project focused on examining the utility of a Government-owned Petri Net simulation tool, Modeler. A number of commercial Petri Net tools are available, and any of them may now be more mature than Modeler. At the beginning of the efforts in this project, fewer tools were available and Modeler exhibited the greatest maturity at that time. A more recent evaluation of the tools available has not been conducted.

### PRINCIPAL FINDINGS:

Three findings were of particular interest:

- The Modeler simulation tool exhibits reasonable agreement with expected results for all test cases performed.
- The Modeler simulation tool is not yet mature enough to support production simulation efforts. It is, however, well suited to use as a rapid prototyping tool.
- Modeler does exhibit significant promise as a production simulation tool, once it has matured further. Modeler's graphical representation of simulation structure presents significant advantages, particularly when changes to an existing simulation are required.

STUDY GIST (Continued)

IMPACT OF THE STUDY:

The several efforts comprising this study had two principal impacts:

- A "certification" of the All-Source Analysis System (ASAS) Network model (ASASNET) was performed, and a written report of findings was provided to the model proponent.
- A validation of the Modeler simulation engine was performed, and a number of bugs in the software were discovered and brought to the developer's attention. As a result of these inputs, a number of improvements to the reliability and validity of Modeler have been achieved.

SPONSOR:

Director,  
US Army Model Improvement and Study Management Agency  
Attn: L. McGlynn  
1725 Jefferson Davis Highway (Suite 808)  
Arlington, VA 22202

PRINCIPAL INVESTIGATOR: John J. DiLeo, C4I Branch, AMSAA

POINT OF CONTACT FOR COMMENTS/QUESTIONS:

Director,  
US Army Materiel Systems Analysis Activity  
Attn: AMXSU-CA (J. DiLeo)  
Aberdeen Proving Ground, MD 21005-5071

(410) 278-6436 DSN 298-6436  
e-mail: dileo@arl.mil

DTIC ACCESSION NUMBER OF REPORT: Report sent to DTIC, Accession Number not available.