AFRL-IF-RS-TR-2002-97
In-House Final Technical Report
May 2002

ROUTING FOR FUTURE MILSATCOM ARCHITECTURES

Gregory J. Hadynski

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2002-97 has been reviewed and is approved for publication.

APPROVED:

PETER K. LEONG, Chief
Information Connectivity Branch
Information Grid Division
Information Directorate

FOR THE DIRECTOR:

WARREN H. DEBANY JR.
Technical Advisor
Information Grid Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	**2. REPORT DATE** May 2002	**3. REPORT TYPE AND DATES COVERED** **In-House Final, Sep 92 – Dec 96**

4. TITLE AND SUBTITLE

ROUTING FOR FUTURE MILSATCOM ARCHITECTURES

5. FUNDING NUMBERS

PE - 62702F
PR - 4519
TA - 63
WU - 55

6. AUTHOR(S)

Gregory J. Hadynski

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

AFRL/IFGC
525 Brooks Road
Rome, NY 13441-4505

8. PERFORMING ORGANIZATION REPORT NUMBER

AFRL-IF-RS-TR-2002-97

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

AFRL/IFGC
525 Brooks Road
Rome, NY 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

AFRL-IF-RS-TR-2002-97

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Gregory J. Hadynski/IFGC/315-330-4094

12a. DISTRIBUTION / AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***

This report documents the results of the AFRL in-house effort titled “Routing for Next Generation MILSATCOM.” The report is divided into several sections corresponding to the efforts’ main thrusts. While this effort was in existence for several years, its objectives changed over time and the end result is several distinct efforts performed under the umbrella of this effort. When this effort began, its objective was to develop strategies for routing messages in the next generation MILSATCOM environment. The developed routing strategies/algorithms were to be prototyped and tested in computer simulations to determine their performance. At management’s direction, this program was then redefined to investigate the feasibility of Global Grid theatre extension objectives. Through analysis and simulation, this program was to investigate the applicability of SHF and EHF satellite resources to play the role of gateway in a high-speed global network. One other distinct effort that was performed was the result of a request from US Space Command (USSPACECOM) to perform a preliminary validation and accreditation of their NATE C3 simulation model. A report was previously prepared for USSPACECOM but the results were not published in an AFRL/IF Technical Report until now.

14. SUBJECT TERMS

communications networks, satellites, space networks, space communications, modeling & simulation

15. NUMBER OF PAGES

73

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

TABLE OF FIGURES

1.0 INTRODUCTION

This report documents the results of an Air Force Research Laboratory Information Directorate (AFRL/IF) in-house program titled “Routing For Next Generation MILSATCOM”. The report is divided into several distinct sections corresponding to the effort’s main thrusts. While this program was in existence for several years, its objectives changed over time and the end result is several distinct efforts performed under the umbrella of this one effort. Since this effort was in existence for several years, the vast majority of the work documented here was actually performed by AFRL/IF’s organizational predecessor, Rome Laboratory. However, Rome Laboratory technically no longer exists and any references to Rome Laboratory that were in previous drafts of this report have been replaced with AFRL/IF as a result.

When this effort began, its objective was to develop strategies for routing messages in the next generation military satellite communications (MILSATCOM) environment. The developed routing strategies/algorithms were to be prototyped and tested in computer simulations to determine their performance. While this program was redirected midstream to pursue other objectives, the partial results of this portion of the program are documented in Section 2 of this report.

At management’s direction, this program was redefined to support theatre extension objectives of the Global Grid Program. Under this new identity, the objective of this program was to investigate the feasibility of Global Grid’s theatre extension objectives. Through analysis and simulation, this program was to investigate the applicability of super high frequency (SHF) and extremely high frequency (EHF) satellite resources to playing the role of gateway in a high speed global network. Under these general guidelines, several subtasks were initiated to explore various aspects of the concept of space based asynchronous transfer mode (ATM) switches. Results of this portion of the program are documented in Section 3 of this report.

Mid-way through the ATM related tasks, another major task was undertaken. AFRL/IF received a request from the Chief Scientist of U.S. Space Command (USSPACECOM) to perform a preliminary validation and accreditation of their NUICCS Analyst Technical Environment (NATE) command, control, & communications simulation model. Since this model was similar in many ways to the simulator being used for this in-house program and the same principal engineer was to be used, the decision was made to include the NATE Validation as a new task in this program. While a report documenting the results of the NATE Validation was prepared for USSPACECOM, the results were never published in an AFRL/IF Technical Report. For this reason, the results of the NATE Validation are documented in Section 4 of this report.

Upon completion of the NATE Validation task, work resumed on the ATM related tasks started earlier. While by that time most of the earlier planned ATM tasks were overcome by events, I decided to add one final task in that area. Early on, I had considered the use of an existing in-house simulation model, the Multiple Satellite System (MSS) End-To-End Simulation (ETESIM), as a tool for studying ATM satellite networks. Originally, I had determined that there were limitations to the applicability of our ETESIM to the ATM satellite network problem. However, after further thought I realized that a relatively small amount of effort could result in several small but meaningful ATM related changes and a version of the ETESIM could be created which was hardwired for ATM analysis. While I was successful in creating this new ATM version of our ETESIM tool and it was a good learning experience for me, it was nonetheless a wasted task in the end. The hardware platform that ETESIM was hosted on, which was already largely obsolete at the beginning of the task, broke down altogether by the time this task was completed. Given the obsolescence of the hardware, it was not worth the investment to repair it and the entire system was scrapped. Since it seemed pointless to do so, I have not included a separate section in this report to document this task.

2.0 ROUTING ALGORITHM DEVELOPMENT

2.1 ROUTING ALGORITHM DEVELOPMENT OVERVIEW

The main objective of this task was to develop a simple rule-based message routing algorithm for satellite communications networks. Emphasis was intentionally kept on making the algorithm simple in operation. While the algorithm was never intended to provide optimal solutions to any routing problem, its simple operation was intended to greatly reduce the processing necessary to perform the routing function in a large communications network.

This task consisted of five main parts. First, a candidate satellite communications network was defined. Second, the concept of operation of the routing algorithm was defined. Third, the algorithm was implemented in software. Fourth, its performance in the candidate communications network was simulated in a static environment. Fifth, its performance would be simulated in a dynamic environment. The fifth subtask was never done due to this project being redirected to pursue other objectives.

While the main objective of this effort was to develop a routing algorithm, it also resulted in a useful by-product. While implementing the algorithm, it became apparent that a methodology for quickly developing routing algorithms would also be a product of this effort.

2.2 CHOOSING A CANDIDATE COMMUNICATIONS NETWORK

The term satellite communications network is rather broad. Communications requirements can vary widely from network to network. For this reason, before anything else was done under this effort, a candidate communications network was chosen to design to.

In general, the network would consist of earth terminals and satellites. In conventional satellite communications systems, a small number of satellites in geosynchronous orbit are used as relay nodes for earth terminal to earth terminal communications. Only a small number of complex satellites are needed in this type of system, because their altitude generally provides them with a very wide field of view. Another key characteristic of this type of system is that the satellites are stationary with relation to any point on the earth. In this type of system, there are relatively few communications networking problems, essentially none in the area of routing algorithms.

However, there have been several satellite communications systems proposed, as well as a couple that have actually been built, over the past several years that follow a considerably different design philosophy. In these systems, the satellite relays would be placed in a much lower orbit around the earth. The lower altitude of the satellites would mean that the satellites would have a much smaller field of view and more satellites would be necessary to provide complete coverage of the earth. However, each satellite would be much less complex than their geosynchronous counterparts and the larger numbers would reduce the importance of any single satellite to system operation. This would result in a communications system that could be much more survivable.

In this type of system, the term communications network suddenly becomes very applicable. Due to the small field of view of the satellites, messages sent over long distances can no longer be accomplished by simply sending it up to a satellite and that satellite sending it back down to its destination. Instead, the message may be relayed several times. This could be done by sending the message up to a satellite, the satellite sending it back down to an intermediate earth terminal, that terminal sending it back up to another satellite, etc., until the message gets to its destination. However, this could also be done by sending the message up to a satellite, that satellite relaying it to another satellite, etc., until the message reaches a satellite which has the destination earth terminal within its field of view. The latter is the method that is generally considered in this type of system. Therefore, since these satellites will not be stationary with respect to the earth, the communications system in question becomes a communications network with a dynamic topology.

Ultimately, the future military satellite communications architecture is intended to be an integrated environment consisting of both geosynchronous and low earth orbit (as well as any orbit in between) satellite systems and including both military and commercial space assets in a variety of operating frequencies and data rates. Therefore, it was our intention to ultimately also simulate any algorithms that we developed in an integrated environment of this sort. However, it is the low earth orbit systems that are the drivers for satellite communications routing algorithm development and this was the general type of network configuration that was chosen for a baseline architecture in this effort. Fairly arbitrary numbers of 100 satellites in a low earth orbit (750 km) and 10 earth terminals were chosen as network communications nodes. To simplify the static simulation program, the latitudes and longitudes of all satellites and earth terminals were chosen at random by the computer. This is certainly an unrealistic characteristic for a deployed system, but with a sufficiently large and dense constellation, it’s effective as a crude approximation. Additionally, the transmitters on the satellites had a maximum range of 2000 km.

2.3 ROUTING ALGORITHM CONCEPT OF OPERATION

2.3.1 BASELINE ROUTING ALGORITM

As was mentioned earlier, a conscious effort was made to make the operation of the algorithm as simple as possible. The algorithm was to feature distributed operation. Entire paths of messages did not have to be chosen at one time. Each network node along a message path would have the responsibility of choosing only the next node to send the message to. This is significant for various reasons. For one, the problem that each node must solve is now a local one, rather than a global one. Therefore, the problem should be much simpler to solve and require much less information about the rest of the network. Second, a distributed approach is much more tolerant to changes in network conditions. If network conditions change somewhere during the course of a message transmission, the entire original message path does not need to be recalculated. The changed network conditions are simply accounted for in the remainder of the message path.

There were slightly different versions of the algorithm depending on whether a communications node is an earth terminal or a satellite. There were a total of five criteria, or conditions, used to determine the next network node to send any message to. These five conditions are listed below.

1. Is the message destination currently in range?
2. Is satellite node No. __ in the general direction of the destination?
3. Has satellite node No. __ already been used for this message path?
4. Is satellite node No. __ isolated from the network?
5. Is satellite node No. __ congested (message queue full or nearly full)?

The first conditions meant that if the destination for a message could be reached directly by the satellite that was currently holding the message, relay nodes were not necessary at all and the message should be sent directly to the destination. According to the second condition, it would be preferable to not send the message in the opposite direction of the message. While the world is round and the message should get there eventually anyway, it made sense to avoid this if possible. The third condition forbade sending a message back to a satellite that it had already passed through. This was necessary in the algorithm to avoid a message taking an endless loop as a path. The fourth condition was necessary to avoid message paths that were dead ends. Finally, the fifth condition said that it was preferable to avoid satellite network nodes that had message queues that were full or nearly full. This was to help alleviate congestion in the network and to avoid creating bottlenecks in the network.

As was mentioned earlier, there were actually two slightly different versions of the algorithm for the two types of nodes in our network. In the version of the algorithm intended for the earth terminals, conditions 1 and 3 were absent. Condition 1 was not used, because two earth terminals could not communicate directly with each other. Condition 3 was not used, because when the algorithm was being performed by an earth terminal, the earth terminal was the source of the message. Therefore, any satellite that it might consider sending the message to next could not possibly be one which that message had already passed through. The earth terminal version of the algorithm is shown below. The conditions were combined to form rules and for each message sent, each rule was checked sequentially in the order listed here until all of the conditions associated with a rule were satisfied and the rule was subsequently triggered. This rule determined where the message was sent next. For each message, the satellites in the earth terminal's neighbor list were checked sequentially until a satellite relay was found which satisfied the most conditions possible. The first choice would be a satellite that met all three conditions. If this were not possible, the second choice would be a satellite that would be in the right direction, and not be isolated. If this were not possible, the algorithm would settle for the first satellite that wasn't isolated. The algorithm is shown below for a system with n satellites.

```
R1          If          Satellite No. 1 is in the right direction
                        Satellite No. 1 is not isolated
                        Satellite No. 1 is not congested
            Then        send message to Satellite No. 1
```

R2	If	Satellite No. 2 is in the right direction
		Satellite No. 2 is not isolated
		Satellite No. 2 is not congested
	Then	send message to Satellite No. 2
.		
.		
.		
Rn	If	Satellite No. n is in the right direction
		Satellite No. n is not isolated
		Satellite No. n is not congested
	Then	send message to Satellite No. n
R(n+1)	If	Satellite No. 1 is in right direction
		Satellite No. 1 is not isolated
	Then	send message to Satellite No. 1
R(n+2)	If	Satellite No. 2 is in right direction
		Satellite No. 2 is not isolated
	Then	send message to Satellite No. 2
.		
.		
.		
R(2n)	If	Satellite No. n is in right direction
		Satellite No. n is not isolated
	Then	send message to Satellite No. n
R(2n+1)	If	Satellite No. 1 is not congested
		Satellite No. 1 is not isolated
	Then	send message to Satellite No. 1
R(2n+1)	If	Satellite No. 2 is not congested
		Satellite No. 2 is not isolated
	Then	send message to Satellite No. 2
.		
.		
.		
R(3n)	If	Satellite No. n is not congested
		Satellite No. n is not isolated
	Then	send message to Satellite No. n
R(3n+1)	If	Satellite No. 1 is not isolated
	Then	send message to Satellite No. 1
R(3n+2)	If	Satellite No. 2 is not isolated
	Then	send message to Satellite No. 2
.		
.		
.		
R(4n)	If	Satellite No. n is not isolated

```
                 Then          send message to Satellite No. n
```

Since all five conditions would need to be used, the version of the algorithm that would be used by each satellite would be very similar, but slightly more complex. This version of the algorithm is shown below.

```
R1          If            Destination node is reachable directly
            Then          send message to destination node
R2          If            Satellite No. 1 is in right direction
                          Satellite No. 1 is not isolated
                          Satellite No. 1 was not already used
                          Satellite No. 1 is not congested
            Then          send message to Satellite No. 1
R3          If            Satellite No. 2 is in right direction
                          Satellite No. 2 is not isolated
                          Satellite No. 2 was not already used
                          Satellite No. 2 is not congested
            Then          send message to Satellite No. 2
.
.
.
R(n+1)      If            Satellite No. n is in right direction
                          Satellite No. n is not isolated
                          Satellite No. n was not already used
                          Satellite No. n is not congested
            Then          send message to Satellite No. n
R(n+2)      If            Satellite No. 1 is in right direction
                          Satellite No. 1 is not isolated
                          Satellite No. 1 was not already used
            Then          send message to Satellite No. 1
R(n+3)      If            Satellite No. 2 is in right direction
                          Satellite No. 2 is not isolated
                          Satellite No. 2 was not already used
            Then          send message to Satellite No. 2
.
.
.
R(2n+1)     If            Satellite No. n is in right direction
                          Satellite No. n is not isolated
                          Satellite No. n was not already used
            Then          send message to Satellite No. n
R(2n+2)     If            Satellite No. 1 is not isolated
```

```
                          Satellite No. 1 was not already used
           Then           send message to Satellite No. 1
R(2n+3)    If             Satellite No. 2 is not isolated
                          Satellite No. 2 was not already used
           Then           send message to Satellite No. 2
.
.
.
R(3n+1)    If             Satellite No. n is not isolated
                          Satellite No. n was not already used
           Then           send message to Satellite No. n
```

2.3.2 EXTENSIONS OF ALGORITHM FOR INTEGRATED ENVIRONMENT

In order to extend the use of this baseline algorithm for use in the integrated satellite communications environment planned for the future, an additional decision process was needed. Before a message was routed, a type of satellite resource was first selected based on the nature of the message traffic. Types of satellite resources (e.g. UHF LEO, EHF GEO, etc.) were considered rather than actual systems to allow the decision process to include systems that did not exist at the time but conceivably could in the future. Factors taken into consideration in this decision process included priority of the message, data rate requirements of the message, whether or not the message was a long voice transmission, and whether access to polar regions was required. If a message was of low priority and low data rate was sufficient, a UHF satellite resource was chosen as a preference. If high data rate was necessary, a SHF satellite resource was chosen as a preference. If a message was of high priority, an EHF satellite resource was chosen as a preference. In this case, EHF satellite resources are assumed to be highly reliable, but low data rate systems such as Milstar. If a message was not a long voice transmission, a LEO satellite resource was chosen as a preference due to the lower propagation delay associated with LEO systems. If the message was a long voice transmission, a GEO satellite resource was chosen to avoid the need for satellite hand-offs in the middle of the transmission. The entire Satellite Resource Selector (SRS) algorithm is shown below.

Satellite Resource Selector (SRS) Algorithm

```
Rule1      If             destination is beyond range of geo satellite
                          message is of low priority
                          low data rate link is sufficient
                          message will not be a long voice transmission
           Then           preferred resource is a network of UHF LEO sats

Rule2      If             destination is beyond range of geo satellite
                          high data rate link is necessary
```

		message will not be a long voice transmission
	Then	preferred resource is a network of SHF LEO sats
Rule3	If	destination is beyond range of geo satellite
		message is of high priority
		message will not be a long voice transmission
	Then	preferred resource is a network of EHF LEO sats
Rule4	If	source or destination is in a polar region
		message is of low priority
		low data rate link is sufficient
	Then	preferred resource is a network of UHF LEO sats
Rule5	If	source or destination is in a polar region
		high data rate is necessary
	Then	preferred resource is a network of SHF LEO sats
Rule6	If	source or destination is in a polar region
		message is of high priority
	Then	preferred resource is a network of EHF LEO sats
Rule7	If	message is a long voice transmission
		message is of low priority
		low data rate link is sufficient
	Then	preferred resource is a UHF GEO satellite
Rule8	If	message is a long voice transmission
		high data rate is necessary
	Then	preferred satellite resource is a SHF GEO satellite
Rule9	If	message is a long voice transmission
		message is of high priority
	Then	preferred satellite resource is a EHF GEO satellite
Rule10	If	UHF LEO satellite network is preferred resource
		there are no UHF LEO satellites available
	Then	preferred resource is a UHF GEO satellite
Rule11	If	SHF LEO satellite network is preferred resource
		there are no SHF LEO satellites available
	Then	preferred resource is a SHF GEO satellite
Rule12	If	EHF LEO satellite network is preferred resource

```
                         there are no EHF LEO satellites available
          Then           preferred resource is a EHF GEO satellite
```

The original routing algorithm for the earth terminals was then modified slightly to incorporate the SRS decision process into the routing rules. The updated version of the resulting Earth Terminal Uplink Selector (ETUS) algorithm is shown below.

EARTH TERMINAL UPLINK SELECTOR (ETUS) ALGORITHM

For an Earth Terminal which tracks n neighboring satellites:

```
Rule 1      If       Satellite No. 1 is of preferred satellite resource type
                     Satellite No. 1 is in the right direction
                     Satellite No. 1 is not isolated
                     Satellite No. 1 is not congested
            Then     send message to Satellite No. 1

Rule 2      If       Satellite No. 2 is of preferred satellite resource type
                     Satellite No. 2 is in the right direction
                     Satellite No. 2 is not isolated
                     Satellite No. 2 is not congested
            Then     send message to Satellite No. 2
.
.
.
Rule n      If       Satellite No. n is of preferred satellite resource type
                     Satellite No. n is in the right direction
                     Satellite No. n is not isolated
                     Satellite No. n is not congested
            Then     send message to Satellite No. n

Rule n+1    If       Satellite No. 1 is of preferred satellite resource type
                     Satellite No. 1 is in the right direction
                     Satellite No. 1 is not isolated
            Then     send message to Satellite No. 1

Rule n+2    If       Satellite No. 2 is of preferred satellite resource type
                     Satellite No. 2 is in the right direction
                     Satellite No. 2 is not isolated
            Then     send message to Satellite No. 2
.
.
```

```
.
Rule 2nIf              Satellite No. n is of preferred satellite resource type
                            Satellite No. n is in the right direction
                            Satellite No. n is not isolated
             Then           send message to Satellite No. n

Rule 2n+1    If             Satellite No. 1 is of preferred satellite resource type
                            Satellite No. 1 is not congested
                            Satellite No. 1 is not isolated
             Then           send message to Satellite No. 1

Rule 2n+2    If             Satellite No. 2 is of preferred satellite resource type
                            Satellite No. 2 is not congested
                            Satellite No. 2 is not isolated
             Then           send message to Satellite No. 2
.
.
.
Rule 3nIf              Satellite No. n is of preferred satellite resource type
                            Satellite No. n is not congested
                            Satellite No. n is not isolated
             Then           send message to Satellite No. n

Rule 3n+1    If             Satellite No. 1 is of preferred satellite resource type
                            Satellite No. 1 is not isolated
             Then           send message to Satellite No. 1

Rule 3n+2    If             Satellite No. 2 is of preferred satellite resource type
                            Satellite No. 2 is not isolated
             Then           send message to Satellite No. 2
.
.
.
Rule 4nIf              Satellite No. n is of preferred satellite resource type
                            Satellite No. n is not isolated
             Then           send message to Satellite No. n
```

Once a satellite resource type had been chosen for a message and a message had entered a given satellite network, it was assumed that it would remain in that network until the message was delivered to the destination. To not make this assumption and to allow a message to pass through multiple and diverse satellite networks would have the implication that all satellites would be interoperable with each other, regardless of frequency, waveform, data rate, etc., and that seemed unrealistic. Therefore, the link selection process for the satellites remained unchanged.

2.4 IMPLEMENTATION

The C programming language was chosen for implementing the algorithm due to its availability of a C compiler to the project engineer.

As was intended, the algorithm was compact and quick to execute once it was implemented. Only 61 lines of C code were necessary for the baseline earth terminal version and only 114 lines were necessary for the baseline satellite version. It's quite likely that the code could have been made even smaller and quicker by a more experienced C programmer. The extensions for the integrated satellite environment were not coded due to redirection of the effort.

A large reason for the simplicity of the code is that lookup tables were used wherever possible. There were two lookup tables needed for each version of the baseline algorithm. Each version had a table of the latitudes and longitudes of each of the earth terminals. The size of this table depends on how many earth terminals are in the system. If there are m earth terminals in the system, the size of the table will be m by 2. Both the earth terminals and the satellites will also have a neighbor table. There are a finite number of satellites that either an earth terminal or a satellite is in range of at any given time. Therefore it is not necessary to consider every satellite in the system as a potential next node in a message path. One only needs to consider those satellites that are available at the time. The size of these neighbor tables is somewhat arbitrary. Since the number of neighboring satellites that any given node will have will vary with time, the simplest way to implement this table is to put an upper limit on how many neighbors each node will keep track of. In the software developed for this project, each earth terminal kept track of at most 10 neighboring satellites and each satellite kept track of at most 8 neighboring satellites. Each entry in these tables has four fields. Each entry includes a satellite identification number, its latitude and longitude, and a flag indicating whether that satellite is congested or not.

The contents of these neighbor tables would obviously vary with time as the network topology changes. However, all information in these tables could be provided by the neighboring satellites themselves during routine operation. Periodically, each satellite would scan with its antenna the area around it to determine which satellites are its neighbors and where they are. During these times, those neighboring satellites would also report whether they are currently congested with traffic or not and whether or not their own neighbor tables currently have more than one satellite listed in them (whether they are isolated or not). Each network node could then update its own neighbor table. By not including any satellites that have indicated that they were isolated in the table, the need for the routing algorithm to check this is completely removed.

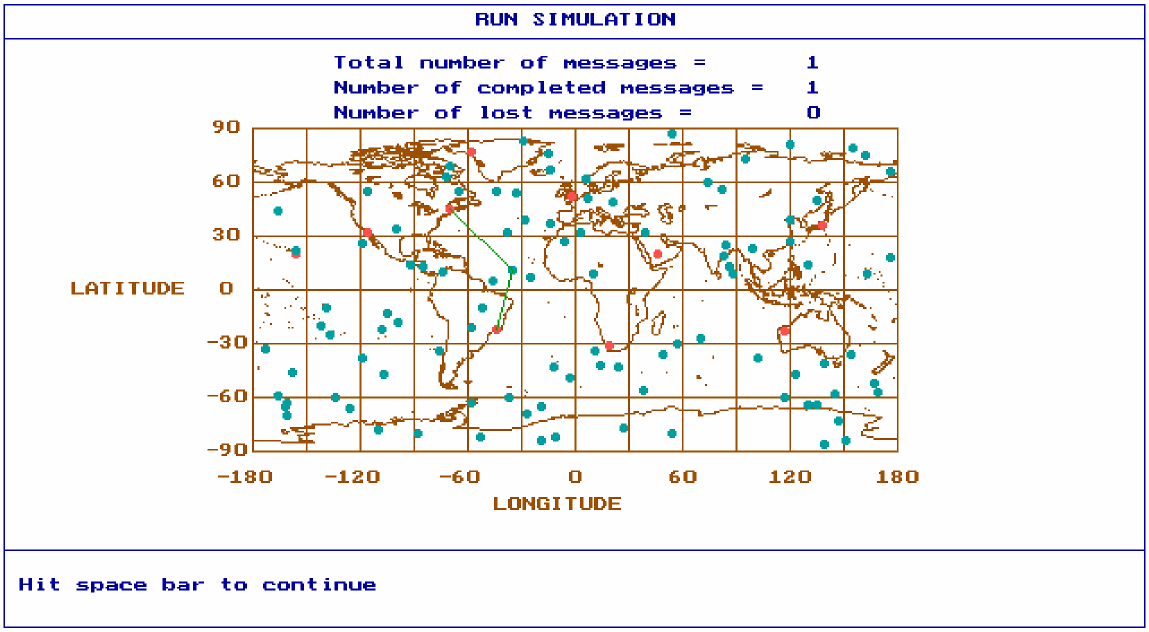
2.5 SIMULATION

In order to test the routing algorithm initially, a fairly simple but flexible static network simulation shell was also developed. The simulator includes a series of easy to use menus that are used to define the ground segment, the space segment, the message traffic, the condition of the network, and the routing algorithm to be used. Since the main purpose of this simulator was to provide a platform for rapid prototyping of routing algorithms for space-based communications networks, there were certainly many compromises made during its development. Most significantly, the success of any message delivery is solely a matter of routing decisions made. Of course, in an actual system the success would also depend upon factors such as the bit error rate on each communications link. Also, the simulator uses a static network topology. Since the satellite systems in question would actually be dynamic in nature, the simulations performed correspond essentially to performance of the routing algorithm during snapshots in time. Once the algorithm was shown to be stable when used in a static environment, the algorithm was intended to be ported to AFRL/IF's Multiple Satellite System (MSS) End-To-End Simulation (ETESIM) for testing in a dynamic environment.

Static simulation results are provided by graphical displays. Examples of these displays are shown in Figures 1 and 2. Each figure shows a grid that represents a rectangular view of the entire Earth. Latitude and longitude are annotated along the axes of the grid. On the grid, the locations of earth terminals are indicated by red dots and the locations of satellites are indicated by blue dots. The intensity of the color of the satellites indicates the queue size onboard those satellites. Light blue indicates a satellite which will be considered to be congested and dark blue indicates a normal status. The magenta lines between network nodes indicate routing decisions made during the simulation. The entire path for one message is displayed at one time. Above the grid, the number of total messages, the number of completed messages, and the number of lost messages are indicated.

Figures 1 and 2 illustrate the performance of the routing algorithm in the same network with the same message. Figure 1 shows the routing decisions made for the case where the network is fully operational and there are no congested nodes. Figure 2 shows the routing decisions made for the case where approximately 30% of the satellites are currently congested. In this figure, the network congestion control feature of the routing algorithm is clearly shown. One of the satellites in the message path shown in Figure 1 is now considered to be congested. As a result, a new path is chosen to avoid the congested area. These figures also emphasize the fact that optimum paths are not being sought. The first path which meets all of our criteria (or as many as possible) is chosen.

However, as advertised, the processing time appears to almost negligible. While a quantitative figure for processing time is currently unavailable, the entire path was calculated and displayed certainly faster than the eye could follow using a standard desktop personal computer.

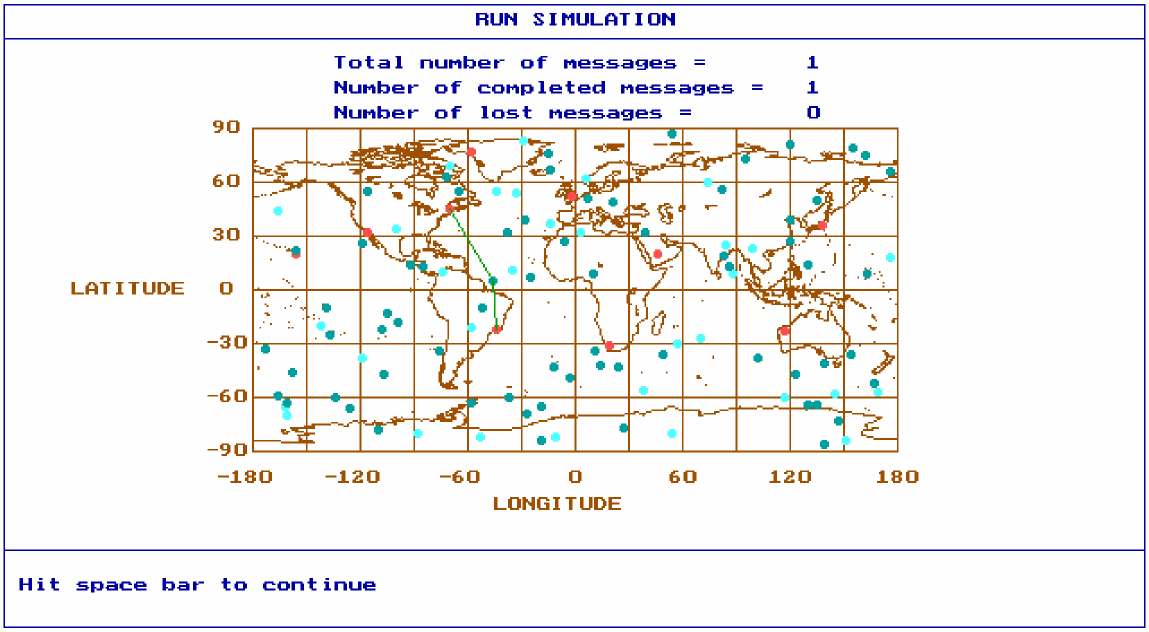


RUN SIMULATION

```
Total number of messages =           1
Number of completed messages =       1
Number of lost messages =            0
```

Hit space bar to continue

Figure 1 - Non-Congested Network Simulation Case



RUN SIMULATION

```
Total number of messages =           1
Number of completed messages =       1
Number of lost messages =            0
```

Hit space bar to continue

Figure 2 - 30% Congested Network Simulation Case

2.6 METHODOLOGY FOR DESIGNING ROUTING ALGORITHMS

While the prime result of this project is a new routing algorithm, it became apparent during the course of this work that a significant by-product of this project was a methodology for fairly quickly and painlessly developing routing algorithms in general. A rule based description of the algorithm's operation proved to be a rapid and easily understood method of developing the top-level description of the algorithm. From this point, it should be a fairly straightforward task to implement the algorithm in an object-oriented programming language. In this case, the C language was used, but it's likely that other object-oriented languages would also be well suited for such a task.

2.7 CONCLUSIONS

The results of this project demonstrate the suitability of rule-based system concepts to the problem of message routing in a communications network. A simple rule-based routing algorithm was developed which possessed the desired qualities of distributed operation and low processing requirements. The algorithm provided admittedly, and intentionally, sub-optimal solutions with great efficiency. By the experience gained in doing this project, it is obvious to the project engineer that this same methodology could be easily be used to provide a myriad of solutions to the same routing problem. A much more optimal solution could have easily achieved at the expense of additional processing and/or data storage. However, I believe that a fully distributed routing algorithm which also handles congestion control in the network with minimal processing and data storage is still quite an achievement.

3.0 ATM ANALYSIS

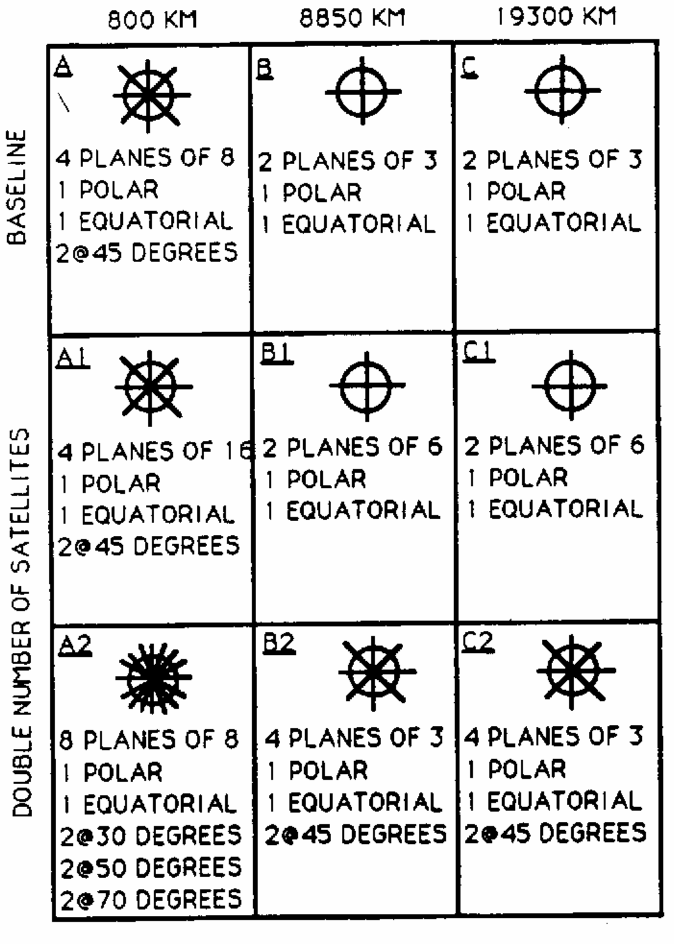
3.1 ATM ANALYSIS OVERVIEW

The general objective of these tasks was to provide modeling & simulation/analysis in support of theatre extension objectives of the Global Grid Program. Global Grid was a program for developing technologies leading to a global high speed communications network. Two key enabling technologies for Global Grid were Asynchronous Transfer Mode (ATM) switches and satellite communications. ATM had become a de-facto standard for switches to which future high speed networks would be built to. Tactical ATM switches were being built by AFRL/IF's Secure Survivable Communications Network (SSCN) Program that would be used in-theatre during military operations. Satellite communications would be needed to communicate with the theatre based "crystal island" ATM networks. At the time, the space segment of this architecture was undecided and conceivably could have included either geosynchronous (GEO) or low earth orbit (LEO) satellites and may have used crosslinks between satellites.

Given this general objective, four tasks were defined. The first was to perform a network connectivity analysis using computer simulations to identify satellite network architectures that would support the Global Grid concept. Options were to include Super High Frequency (SHF) and Extremely High Frequency (EHF) satellite resources in GEO or LEO orbits, with and without crosslink capabilities. The second task would have provided link analyses for the up/downlinks associated with the recommended satellite network architectures. The third task was to investigate the effects of the relatively high bit error rates of satellite communications links on ATM network operations. The final task was to investigate the effects of the relatively high propagation delays of satellite communications links on ATM network operations.

3.2 CONNECTIVITY ANALYSIS

Global Grid scenarios were set up on our MSS End-To-End Simulator and some long-term simulation runs were made for connectivity analysis. For these scenarios, the ground segment consisted of terminals at the five node sites planned for the SSCN Testbed (USAF AFRL/IF, Rome NY; US Army CECOM, Ft. Monmouth NJ; US Navy Nrad, San Diego CA; USAF ACC Langley AFB VA; and DISA, Ft Huachuca AZ) as well as several other terminals located in potential theatre locations. As a baseline space segment, nine different satellite constellations would be considered. The nine constellations chosen had been the subject of investigation of a previous AFRL/IF in-house program. The constellation selection process was described in detail in the final report for that program and will not be repeated here. However, the constellations themselves are summarized below in Figure 3.



	800 KM	8850 KM	19300 KM
BASELINE	A 4 PLANES OF 8 1 POLAR 1 EQUATORIAL 2@45 DEGREES	B 2 PLANES OF 3 1 POLAR 1 EQUATORIAL	C 2 PLANES OF 3 1 POLAR 1 EQUATORIAL
DOUBLE NUMBER OF SATELLITES	A1 4 PLANES OF 16 1 POLAR 1 EQUATORIAL 2@45 DEGREES	B1 2 PLANES OF 6 1 POLAR 1 EQUATORIAL	C1 2 PLANES OF 6 1 POLAR 1 EQUATORIAL
	A2 8 PLANES OF 8 1 POLAR 1 EQUATORIAL 2@30 DEGREES 2@50 DEGREES 2@70 DEGREES	B2 4 PLANES OF 3 1 POLAR 1 EQUATORIAL 2@45 DEGREES	C2 4 PLANES OF 3 1 POLAR 1 EQUATORIAL 2@45 DEGREES

Figure 3 - Baseline Candidate Satellite Constellations

The simulations that were run didn't result in any surprises. Theoretically, each of the satellite constellations used should provide 100% global coverage. The configurations labeled as "baseline" in Figure 3 were originally generated by a closed form solution for full earth coverage found in open literature. The remainder of the configurations was variations of the three "baseline" configurations, introducing redundancy through doubling the number of satellites in the minimal "baseline" versions. As expected, each of the satellite constellations that were simulated provided very nearly continuous coverage at each of the earth terminals over the length of time simulated.

3.3 LINK ANALYSIS

The goal of this task was to do link budget analyses for EHF, SHF, and UHF versions of the satellite constellations studied and determine the communications hardware requirements to support the associated links. It was during this task that the NATE Validation task was added to this program. As a result, the link analysis task was put on hold while the NATE Validation took priority.

Since work on the project's ATM related tasks was preempted for several months so that the NATE Validation could be accomplished, it was appropriate to review the ATM tasks to determine whether it still made sense to do them. By that time, there was no more talk of conceptual space architectures containing satellites in both geosynchronous and low earth orbit. Instead, the Global Grid would, in all likelihood, rely on existing satellite communications resources in geosynchronous orbit. Therefore, it seemed pointless to continue to calculate link budgets for the conceptual satellite constellations described above and this task was effectively canceled. The remainder of the connectivity analysis task was canceled for the same reason.

3.4 BER EFFECT ANALYSIS

This task was started with a literature search to see what had been done in this area already and what this program might be able to add. The result of this literature search was that quite a bit had already been done to study the effects of the relatively high bit error rates associated with satellite links on ATM systems. I found several excellent articles and decided that I could not contribute any original work of any significance to this study area. The reference section of this report will provide references to work in this area.

3.5 PROPAGATION DELAY EFFECT ANALYSIS

This analysis was an expansion of an analysis documented in an article found in the April 1992 issue of IEEE Communications magazine. The article addressed the fundamental relationship between latency and bandwidth of a communications link. The premise of this fundamental relationship is that increased bandwidth will only equate to decreased message transmit times if the queuing plus transmission time delay is greater than the propagation delay of the channel. For every communications link, there is a critical bandwidth beyond which additional bandwidth no longer decreases transmit time and the transmission time becomes limited by the propagation delay, or latency, of the channel. That critical bandwidth is defined as follows:

$$C_{CRIT} = \frac{1000\ b}{(1- r)\ t}$$

In this equation, b is message length in bits, ρ represents the system load, and τ is the propagation delay in milliseconds.

In the case of a local area network, propagation delay is very low and this critical bandwidth is extremely high, making the use of very wide bandwidth technologies a viable option. Take for example a local area network where two network nodes are located .25 miles apart. This distance translates to a propagation delay of approximately 1.3 milliseconds and if we transmit a 1 megabit message across this distance, we would be able to transmit at a rate of at least (0 load case) 745 Gbps before our link became latency limited.

If we consider a cross-country link from New York to Los Angeles instead, the propagation delay is much higher, but we can still take advantage of wide bandwidth technologies. The distance from New York to Los Angeles is approximately 2500 miles, which translates to a propagation delay of 13 milliseconds. If we transmit a 1 megabit message across this distance, we would be able to transmit at a rate of at least (0 load case) 77 Mbps before our link became latency limited.

Finally, we consider a satellite link. Of the most commonly used or planned satellite orbit altitudes, the most common and worst case is the geosynchronous orbit. At an altitude of 22,284 miles, a geosyncronous satellite's altitude is roughly an order of magnitude larger than the distance from New York to Los Angeles. Furthermore, since both an uplink and a downlink would be required at a minimum, the distance between two points on the earth via a satellite would be roughly twice the altitude of the satellite. The commonly used number for propagation delay from earth to satellite to earth is 250 milliseconds. If we transmit a 1 megabit message across this distance, we would be able to transmit at a rate of only 4 Mbps before our links became latency limited.

However, the 4 Mbps limitation stated above is but one example of the critical bandwidth in a satellite system. It corresponded to the worst case geosynchronous orbit and a zero network load condition. Also, since the critical bandwidth is a function of message size as well, the critical bandwidth will increase for messages larger than 1 megabit.

Figures 4 through 10 illustrate the effects of satellite altitude, network load, and message size on the critical bandwidth of the satellite system. Figure 4 plots critical bandwidth vs. load as a function of satellite altitude. Figure 5 plots critical bandwidth vs. load for a 1 MB file relayed through a variety of satellite constellations which have been proposed, and in some cases developed, in recent years. A generic geosynchronous altitude satellite is also included. Figures 6-10 plot critical bandwidth vs. load for a variety of file sizes relayed through a variety of satellite constellations.

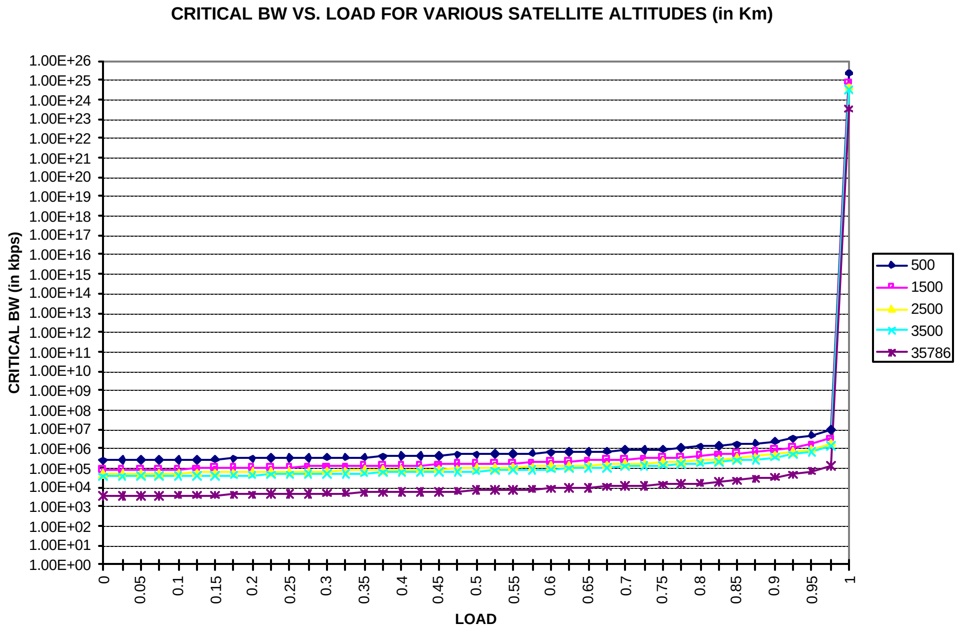


Figure 4 - Critical Bandwidth As A Function Of Altitude

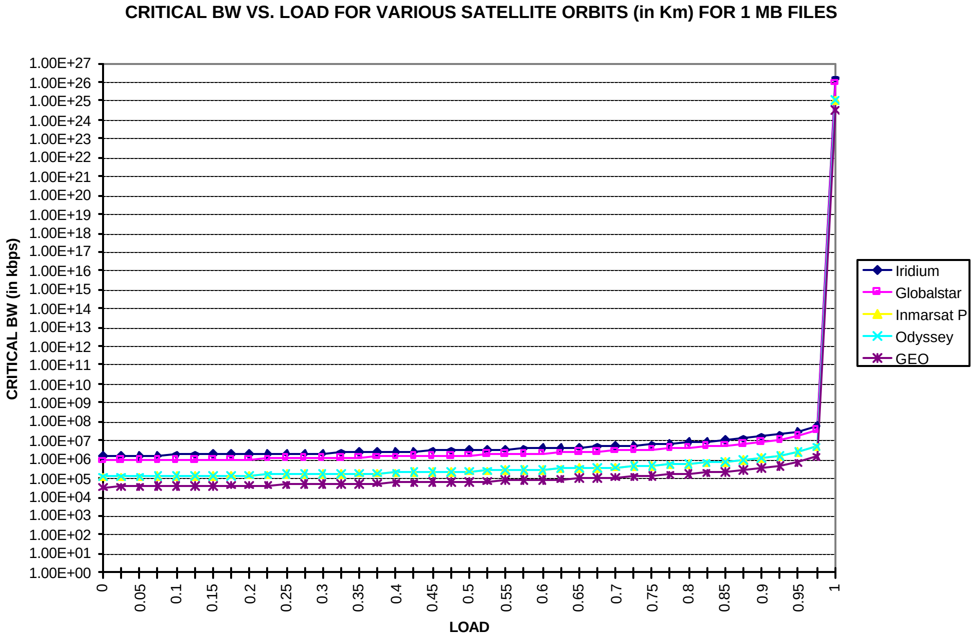


Figure 5 - Critical Bandwidth For Selected Satellite Systems

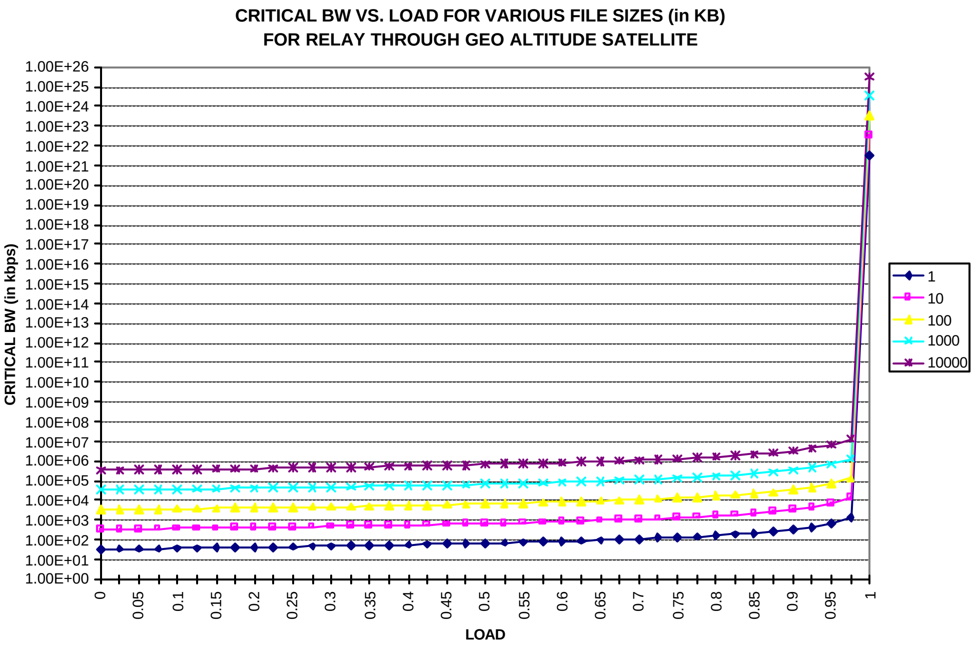


Figure 6 - Critical Bandwidth As A Function Of File Size Through GEO Satellite

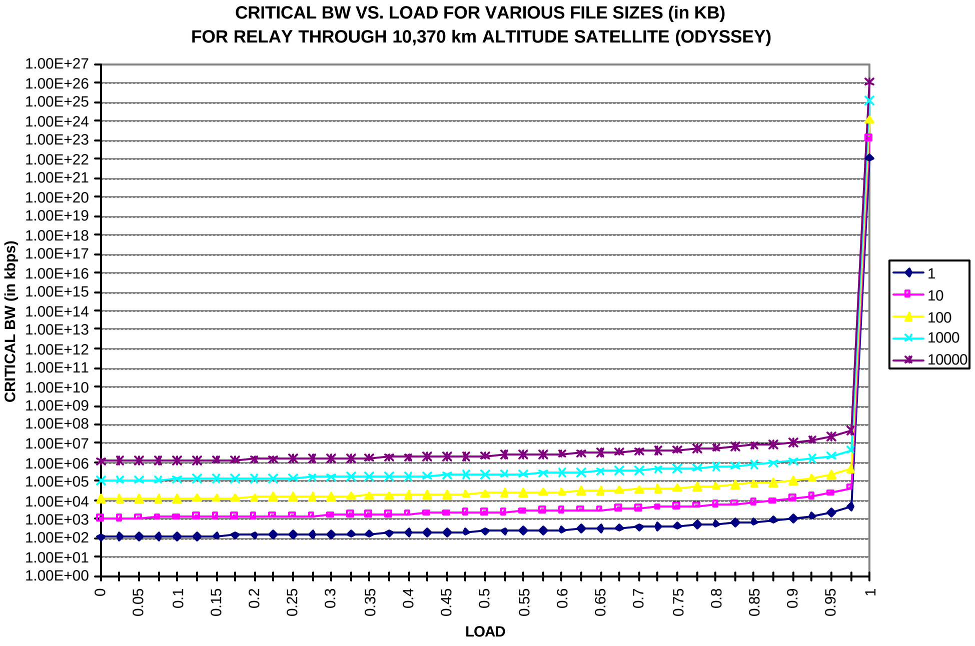


Figure 7 - Critical Bandwidth As A Function Of File Size Through Odyssey Satellite

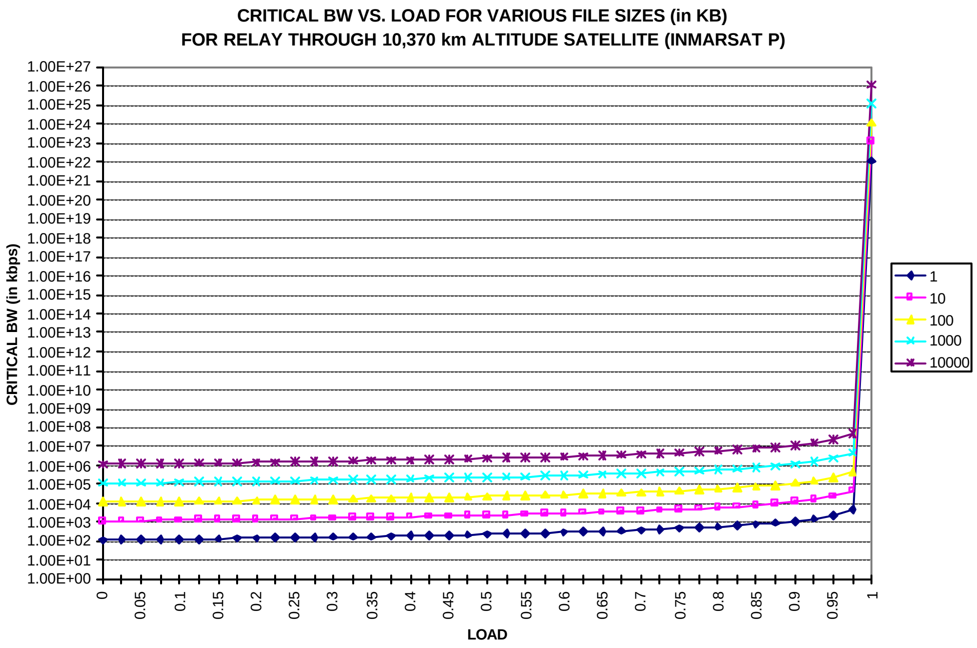


Figure 8 - Critical Bandwidth As A Function Of File Size Through Inmarsat P Satellite

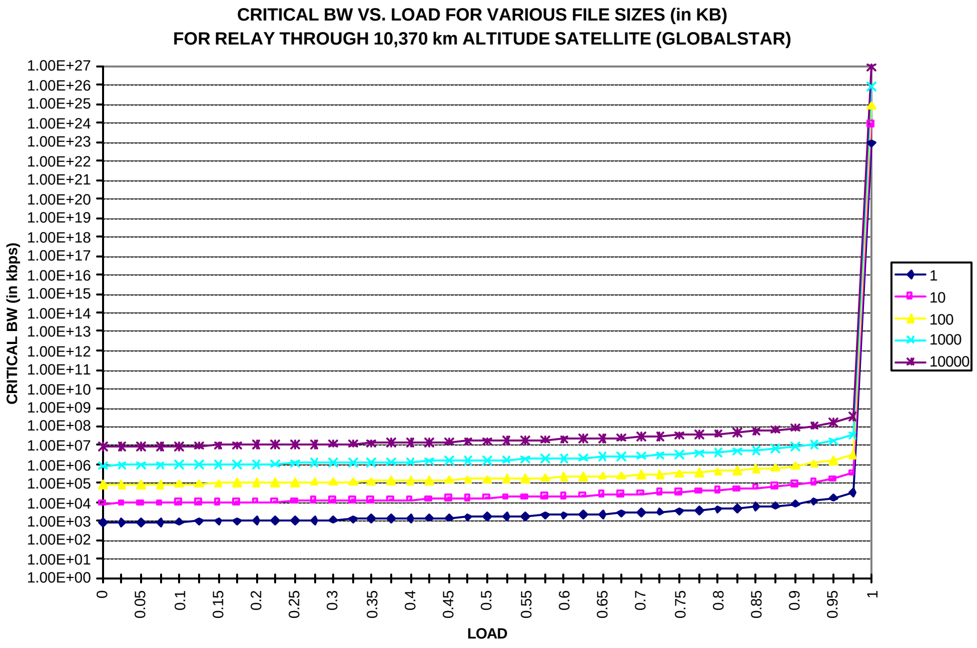


Figure 9 - Critical Bandwidth As A Function Of File Size Through Globalstar Satellite

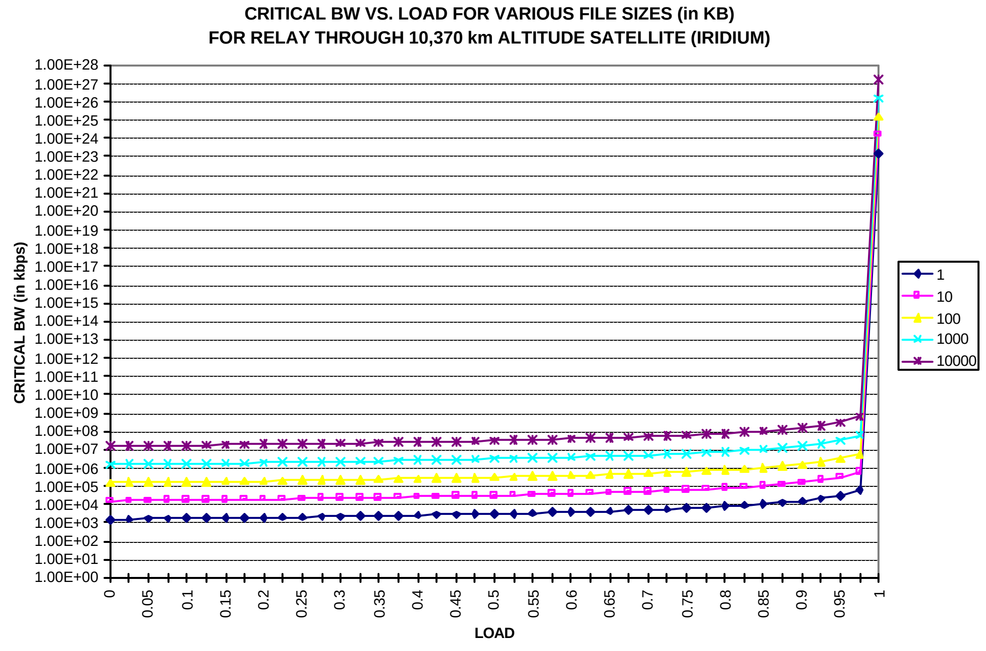


Figure 10 - Critical Bandwidth As A Function Of File Size Through Iridium Satellite

4.0 NATE VALIDATION

The NATE Validation report was originally prepared by Gregory Hadynski and Capt. Michael Mills of AFRL/IF. While there have been no changes in its contents for inclusion in this report, it has been reformatted to form an integral part of this report.

The foreword and acknowledgments for the NATE Validation section of this report were provided by:

Dr. David Finkleman, SES-4
Director of Analysis
North American Aerospace Defense Command and United States Space Command

4.1 FOREWORD

This effort is a milestone in the new military modeling and simulation environment and in joint military efforts. To the best of our knowledge, this is the first disciplined validation of a composite military model. Our commands have implemented aggressively guidance from Congress and the Office of the Secretary of Defense. We are building substantiated confidence in the models and simulations that we use for planning, training, and analysis. AFRL/IF is a knowledgeable, independent agency. Its capabilities and accomplishments qualify the laboratory to examine expertly communications models and computer software. USSPACECOM provided the software and sponsored specific training. This report confirms the value of our relationship. The laboratory found deficiencies in the software. The evaluators exposed uncertainties in elements of the software. We would not have been able to address these matters effectively without this independent validation. By the time we distribute this report, we should have replaced the outdated missile flyout model with a current one whose lineage and documentation are consistent with the rest of the simulation. We have already remedied minor software inconsistencies. We can now claim much greater confidence in analyses that use this tool. At this writing, we have engaged AT&T to conduct a much more comprehensive validation using analysis techniques with a broad commercial basis. This sequence of preliminary “face validation” followed by more comprehensive formal validation may be a paradigm for the future. I hope that this report enables wider implementation of the mandate for model validation.

4.2 ACKNOWLEDGEMENTS

I thank AFRL/IF for accepting this unusual task. We are grateful to Mr. Tony Szalkowski, who advocated this effort within the laboratory, and to Mr. Greg Hadynski and Capt. Mike Mills, who did the work. We appreciate support from the Air Force Studies and Analysis Agency, which directed model development. JS/J8 contributed to funding the NATE model, and the Defense Modeling and Simulation Office is funding formal validation. I appreciate the strong cooperation of the contractors who developed the simulation, SAIC and Autometric, Inc. Col Gordon Long, USAF, and Lt Col Keith James, USAF, spearheaded this

effort within the Unified Commands. Finally, we acknowledge Lt Col Steve Reznick, USAF, and Lt Col Bill Osborne, USAF, who initiated this project during their tenures in USSPACECOM.

4.3 INTRODUCTION

This report documents AFRL/IF's recent preliminary validation of US Space Command's NORAD and USSPACECOM Integrated Command and Control (NUICCS) Analyst Technical Environment (NATE). This work was performed in response to a request made by US Space Command.

The validation of the NATE software contributes to the overall objectives of US Space Command's Project Foretell. The ultimate goal of Project Foretell is to develop the capability to validate reliably the potential of new Air Force Satellite Control Network (AFSCN) system components without the significant expenditures involved in developing and fielding the system components. The first step in achieving this goal is to assemble a modeling and simulation test bed by connecting and, when necessary, validating existing models and simulations of space and C4I systems. NATE is one of the models to be included in Space Command's test bed.

There were four team members involved in the NATE preliminary validation. AFRL/IF acted as the team lead, an independent broker hired to perform an objective evaluation of NATE. US Space Command was the customer, providing a general task description and operational support. Science Applications International Corporation was the developer of both NATE and one of the principal components of NATE, the Strategic Command & Control Architecture Model (STRATC2AM). They provided technical support on the subject of these two models. Finally, the Air Force Studies and Analysis Agency (AFSAA), the government agency responsible for the development of STRATC2AM, provided vital technical support.

This report is basically organized according to the main tasks involved in the NATE validation. The first task was to plan the work necessary for this project. The second task was to validate, if necessary, the individual simulation models within NATE. The third task was to validate the interfaces between the individual models within NATE. The fourth task was to validate the NATE model as a whole. The final task was to prepare a final report. An overview of the NATE simulation environment is also included for those unfamiliar with NATE. Appendix A contains the validation plan. It was included as an appendix rather than reiterating it in the main body of the report. Appendix B contains a list of the documents referenced to perform this analysis. Appendices C & D contain the evaluation matrices which document the validation results and form the basis for much of the discussion in the main body of the report.

Since this report will concentrate on NATE's problems, it is important to put the negative comments in perspective. While NATE may have its problems at this time, it has many good points as well. The communications codes within STRATC2AM are the heart of NATE and its biggest selling point. It is in communications where NATE's commercial competitors fall short. Most commercial communications network programs represent free space communications links simplistically, requiring the user to provide the bit error rate for the links. The most notable exception to this is the OPNET model. OPNET allows the user to create their own link models in code. However, STRATC2AM has a robust set of communications link codes already. Also, NATE's Omni graphical user interface is an excellent visualization tool, providing highly detailed animated displays of global C3I scenarios.

4.4 NATE OVERVIEW

NATE has a storied history beginning in 1974 when work began on the models which eventually evolved into STRATC2AM, one of the four main components of NATE. Today, NATE consists of STRATC2AM, the Omni graphical user interface, the COMET missile flyout model, and the SDP4 & SGP4 orbital propagation models.

STRATC2AM is the official C3 model for AFSAA and has been used for numerous applications such as the Strategic C3 Systems Review, the MILSTAR Program Review, the NATO C3 Architecture Study, the Cheyenne Mountain Upgrade, and the DSCS SCT Upgrade. It supports wide-area network simulations with space nodes, ground nodes and aircraft. Each node model includes C2 node processes and communications transmission equipment (ELF through optical). Communications between nodes are modeled for benign environments, jamming environments, and nuclear environments. One of the more recent STRATC2AM developments has been the addition of the Analytical X-Windows Interface to Simulations (AXIS) graphical user interface.

The development of the NATE environment resulted in the addition of a second, more powerful, graphical user interface known as Omni. Omni is a commercially available product sold by Autometric Inc. which runs on a Silicon Graphics workstation. Omni supports graphic data analysis and provides output in many forms including: pictures, graphs, animated sequences, and text windows. It does this using a mouse-controlled, pull-down menu-driven command system and a multiple, overlapping window environment.

The remaining two NATE components are actually application software for the Omni environment. The COMET and SDP4 & SGP4 models are included in Omni's Astro application package. COMET is a missile propagation model. SDP4 & SGP4 are satellite propagation models originally developed for NORAD. SGP4 is used for modeling the orbits of near-Earth satellites and SDP4 is used for modeling the orbits of deep-space satellites.

A block diagram depicting NATE operation is shown in Figure 11.

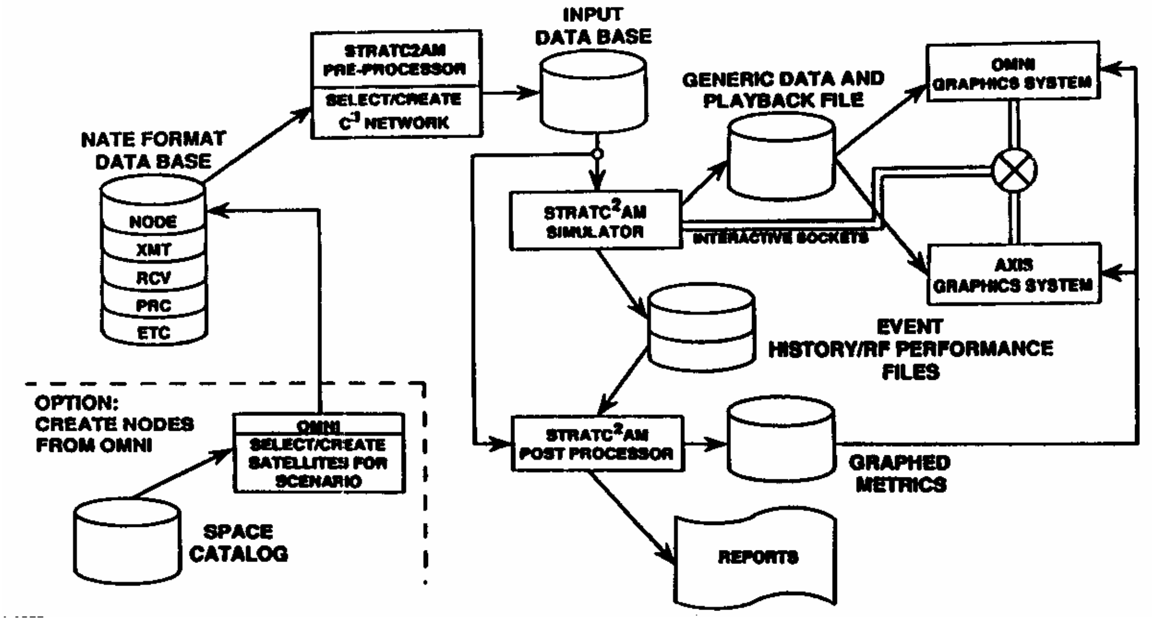


Figure 11: NATE Operation

4.5 PLANNING

4.5.1 LEARNING ABOUT NATE

While learning about NATE may seem like an obvious step in its validation, its importance really does warrant inclusion in this report. Without researching a model, it would be impossible to develop a detailed validation plan. Documenting this research helps to validate the validation process.

AFRL/IF's NATE education began with SAIC visiting AFRL/IF for an introductory briefing and demonstration. This visit was very beneficial and left AFRL/IF with the impression that there were actually many similarities between NATE and models which AFRL/IF uses.

Next, AFRL/IF requested and received full sets of NATE and STRATC2AM documentation. The documentation was reviewed by the engineers who would be working on the NATE validation. The NATE documentation lacked sufficient information on the COMET and SGP4 & SDP4 models. Therefore, additional documentation on these models was requested. The requested documentation on the SGP4 & SDP4 models was received, but for reasons described later documentation on COMET was never received. A full list of the documentation used for the evaluation is included in Appendix B.

Finally, the AFRL/IF engineers working on the project visited SAIC to attend a NATE training course.

4.5.2 FREEZING OF NATE CONFIGURATION

Early in the planning phase, Space Command froze the NATE configuration for the duration of its validation. This served two purposes. First, it told AFRL/IF exactly what configuration it would be using for its analysis. Second, changes to the NATE configuration during its validation would probably require the validation process to be restarted to avoid suspect results. The NATE configuration used during AFRL/IF's analysis consisted of:

STRATC2AM C3 Model (version 2.0)
Omni Graphical User Interface (version 1.3.2)
COMET Missile Flyout Program
SGP4 & SDP4 Satellite Propagation Models
NATE Network Architecture (version 1.0)

4.5.3 SELECTION OF TEST CASES

The selection of test cases for the validation of NATE's interfaces was an arbitrary process to some extent. In testing the interfaces, the main concern was if the data was transferred across the interfaces correctly, and not if the data itself was correct. For this reason, we used an existing scenario called "Simple." Simple is a demo which was created by SAIC to demonstrate the network switching capability of NATE using protocol rule message traffic routing.

It must be pointed out at the start that the selection of particular test cases for the functional validation of NATE as a whole is vitally important. The test case to be used needs to be representative of the application which Space Command ultimately has in mind for NATE. For this reason, Space Command chose to use the existing "DSB" database which describes a strawman surveillance architecture that was created for the Defense Science Board. The DSB database describes a fairly large scale scenario which contains global communications systems, national C2I resources, national information gathering assets, theater assets, and opposing forces.

4.5.4 WRITING NATE VALIDATION PLAN

A detailed validation plan was written and submitted to Space Command for their approval. The plan was approved with minor comments. The resulting plan is included in Appendix A, but is highlighted here for convenience.

The plan divides the NATE validation project into five main tasks: planning, submodel validation, interface validation, NATE validation, and prepare final report. The mechanics for the validation process were adapted from the "Analytical Tool Box Level 1 Face Validation Assessment Plan" prepared by the Martin Marietta Corporation for the National Test Facility. Both plans use a combination of operational effectiveness evaluation matrices and corresponding evaluation criteria to rate the different factors of a simulation model's operational effectiveness.

4.5.5 APPROVAL OF PLAN

Timely feedback from Space Command was critical to the success of this project. Approval of the plan was needed before the validation could proceed and the approval needed to be expedited to avoid unnecessary schedule impacts. Since the evaluation criteria for a software model validation should be closely tied to the customer's intended use of the model, concurrence with Space Command was very important. Fortunately, the plan was quickly accepted with a few minor comments.

4.6 SUBMODEL VALIDATION

Since NATE is actually a composite of existing models, the validation plan was written as a methodology for validating a composite model. The logical approach was to start by validating the submodels. However, in NATE's case, the submodels had undergone validations in the past. For this reason, it was agreed that full submodel validations did not need to be done as part of the NATE validation and that submodel validations would only be done on an "as needed" basis.

Twice during this effort, the need for validation at the submodel level was indicated. In the first, testing the interface between the STRATC2AM post-processor and Omni uncovered a bug which was eventually determined to be in the post-processor. This will be described in greater detail later on during the discussion of the testing of that interface.

It was also determined that a full validation of the Comet missile propagator submodel needed to be done. Unfortunately, neither the time nor the expertise needed to do this was available for this study. Efforts to obtain necessary information on Comet resulted in more questions than answers. According to its original developers, Comet is a simplified missile propagation code which has been around for many years and provides reliable answers to the user who understands the limitations of the model. Its most recent validation took place in late 1989 and early 1990, when a 1987-1988 version was examined and found to be correctly coded. However, in the past, various users of Comet have informally made independent changes to the model while maintaining the original program name. As a result, there are several versions of Comet in existence, only one of which is official. The rest of the versions are unknown quantities to Comet's developers. Unfortunately, the version of "Comet" used in Omni is just such code. Therefore, Omni's version of "Comet" should be considered an unknown quantity in need of further testing.

4.7 INTERFACE VALIDATION

4.7.1 TESTS

As was mentioned earlier, the existing NATE scenario "Simple" was used for the validation of NATE's interfaces. In Simple, low altitude satellites over Korea send missile detection messages to Colorado Springs. Initially, these messages are routed through a constellation of Milstar satellites. However, at 60 second intervals, the low altitude satellites send health and status messages through the Milstar constellation to Colorado Springs. While the health and status messages are being processed by Colorado Springs, the Milstar satellites are not available for the routing of missile detection messages. During these times, the missile detection messages are instead routed through an SDS constellation to Colorado Springs. The following table details interfaces that were tested:

- Omni to STRATC2AM
  - Launched node entry
  - Fixed node entry
  - Moving node entry
  - Satellite entry
  - Communications control file entry
  - Node definition file entry
- STRATC2AM to Omni
  - Communications statistical graphs
    - Graph time delay by message type
    - Graph link message count
    - Graph availability
    - Graph correct message receipt
    - Graph package receipt
    - Graph resource utilization
    - Graph link loading
    - Graph link utilization
    - Graph RF performance
      - Graph probability of receipt
      - Graph signal to interference
      - Graph absorbance
      - Graph scintillation
      - Graph decorrelation
  - Show links
  - Link filters
  - Link shading
- Omni to SGP4 & SDP4
  - Satellite orbit descriptions
- SGP4 & SDP4 to Omni
  - Satellite positions

Table 1: NATE Interfaces Tested

4.7.2 RESULTS

4.7.2.1 OMNI/SGP4 & SDP4 INTERFACES

The first interface that was tested was the interface between Omni and the SGP4 and SDP4 orbital propagation models. Project Spacetrack Report No. 3 which documents the SGP4 and SDP4 models includes examples of results which the models should yield. Using these examples, we were able to compare the results provided by Omni with the results which the models provide. In doing so, we discovered that Omni doesn't allow a user to input the orbital parameters to the same degree of precision which the standard two-card element set description uses. The mean motion loses one digit in the process of inputting the parameters while entering the epoch results in losing three digits.

Overall, the results provided by Omni agreed fairly well with the results contained in the Spacetrack report, but they did not match perfectly. While some error should be expected due to different computer word lengths according to the Spacetrack report, it is difficult to determine if the missing digits of precision were solely responsible for the variance without adding the missing digits and rechecking the results. If the versions used in Omni are later versions than the ones described in the Spacetrack report, then that could account for some of the variance in results. Attempting to provide a quantitative measure of the errors in the Omni results, we determined the magnitude of the position vectors from the xyz coordinates that the SDP4 and SGP4 models provide, and compared the Omni results to the values in the Spacetrack report. For the near earth orbit case, the magnitude of the error was approximately .02 kilometers. For the deep space orbit case, the magnitude of the error was approximately 1.3 kilometers.

Since there is no apparent reason why this precision mismatch could not be fixed, we strongly recommend that a correction be made to the Omni data input window to accommodate the required precision.

4.7.2.2 STRATC2AM POSTPROCESSOR/OMNI INTERFACE

The next interface which was tested was the interface between the STRATC2AM Post-processor and Omni. Omni creates graphs of several STRATC2AM post-processing reports. Omni does this by plotting points from text files generated by the STRATC2AM Postprocessor for each of the statistical reports. Therefore, testing the interface was simply a matter of visually comparing the text files generated by STRATC2AM with the graphs created by Omni.

Although this test sounds trivial (since it isn't likely that Omni can't plot points correctly), when asked to graph statistics for link utilization, link load, and link demand, Omni provided some interesting results. The graphs actually contained a loop. Part of the link utilization graph is shown in Figure 12 to illustrate this.

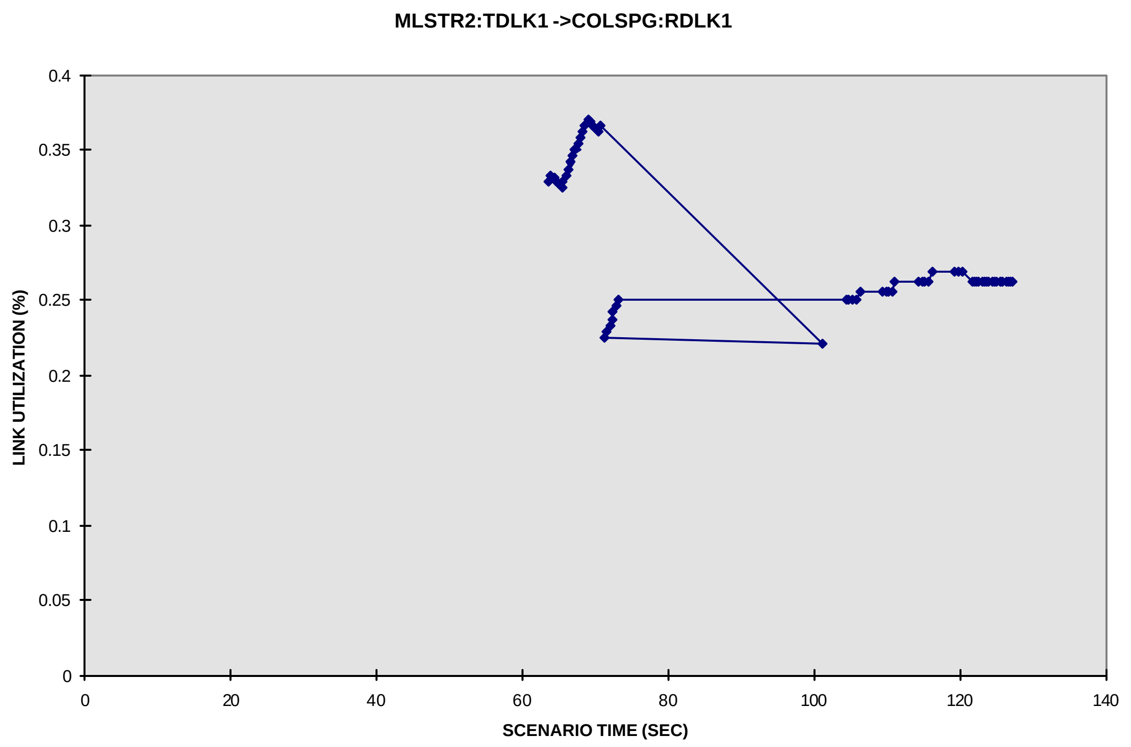


Figure 12: Link Statistics Anomaly Example

Upon further examination, we determined that Omni was doing exactly what it was supposed to do. It plotted the points which STRATC2AM provided. The loop in the graphs was caused by one data point being out of sequence in the file which STRATC2AM created. Therefore, this is not a problem with the interface, but with STRATC2AM itself. Also, it proved to be an intermittent problem. The same link statistics were generated for several other links in the same scenario. None of them exhibited the same problem. According to SAIC, this is a problem they have seen before, but thought had been corrected.

Irregularities were also discovered in plotting the postprocessor's Grouped Packet Receipt report. The first irregularity is that this report has three different names depending on where you look. STRATC2AM generates what it calls a Grouped Packet Receipt report. The Communications Statistical Graph menu in Omni calls it Package Receipt, and the graph is labeled Group Packet Probability of Receipt. At best, this is a minor oversight. At worst, this is truly confusing.

Unfortunately, semantics was not the only problem. This graph did not seem technically correct either. According to the STRATC2AM postprocessor file, with only one data point in this case, the Omni output graph should have had a value of 0 probability of receipt until t = 3.96 seconds, at which time the graph should have had the value 1. When the graph was checked, this isn't what was found. First, the y-axis scale was wrong. Since the data points were probabilities for this graph, the scale should range from 0 to 1. Instead it ranged from 1 to a value that was off the graph and could not be seen. Next, the graph began at 3.96 seconds and there was no value shown at all until 3.96004 seconds. At 3.96004 seconds, the value shown was approximately midway between 1 and the maximum value on the y-axis.

Finally, Omni's use of continuous line graphs to plot some of STRATC2AM's outputs can be very misleading. For example, Omni will indicate positive link utilizations during periods of link inactivity simply because the link is active before and after the period of inactivity. If a line graph is going to be used, it is adviseable that the individual datapoints be highlighted. In many cases, histograms may be even more effective. This is especially true for the link message count statistic. The data which STRATC2AM provides for its link message count statistic was meant to be displayed in a histogram format, but Omni did not have a histogram capability at the time.

4.7.2.3 OMNI/STRATC2AM PREPROCESSOR INTERFACE

The main interface between Omni and the STRATC2AM preprocessor is through what Omni terms a "platform." Within Omni, the user defines fixed sites, satellites, moving nodes, and missiles. A "platform" associates these fixed sites, satellites, moving nodes, and missiles with the communications hardware onboard. From there, Omni will take a set of platforms and either create a new STRATC2AM node definition file or append the platforms to an existing node definition file. Omni will also let the user take the node definition file that they just created and run STRATC2AM without leaving Omni.
This helps to reduce the switching between applications which the NATE user needs to do.

We created one fixed site, one satellite, and one moving node in Omni and tried both creating a new STRATC2AM node definition file with these platforms and appending them to an existing node definition file. The results were consistent between these two approaches. In either case, the values of some of the position variables for the platforms were changed slightly by the interface process. However, the magnitude of these errors was negligible. It is only mentioned here for the sake of completeness. More troubling is the fact that the interface apparently converted the moving node into a launched node.

Perhaps the most significant comment that can be made about the interface between Omni and the STRATC2AM preprocessor is that preprocessor input through Omni is very limited. While new nodes can be created from within Omni, virtually everything else must still be done in the STRATC2AM preprocessor.

4.7.2.4 OMNI/COMET INTERFACE

Although testing the Omni/Comet interface was part of the original validation plan, it was decided that this be postponed. It is recommended that the Comet code in Omni be validated as a submodel prior to testing this interface. Neither the time nor the expertise to accomplish this task was available for this study.

4.7.2.5 STRATC2AM PREPROCESSOR & POSTPROCESSOR/USER INTERFACE

STRATC2AM's pre-processor and post-processor are both adequate functionally, but are definitely not user-friendly. The pre-processor is a command-line user interface which makes data input a very long process. In fact, the interface is so difficult to use that some users actually prefer to use a text editor to create and modify preprocessor files rather than use the pre-processor. Also, inconsistencies in the user interface can be very frustrating. Different keys must be pressed for the same actions depending on where the user is in the interface. The good news is that this should be much less of a problem very soon. The next version of STRATC2AM is to sport a new graphical user interface which looks to be a quantum leap for the STRATC2AM user. Nearly everything that currently must be entered by way of the command-line interface will be able to be entered using simple point and click instructions. The exceptions to this are jammers, events, and user-defined modems which still will require the old pre-processor.

The post-processor is not quite as difficult to deal with as the current pre-processor, but it is still in need of updating. The user request reports by using a menu/command-line interface. The reports themselves are tabular data available in several formats for plotting in other software programs (e.g. Lotus Freelance, Omni, etc.). The post-processor should be modernized to use a graphical interface and create presentation quality tables and graphs. The metrics themselves should also be revisited. Some metrics may no longer be useful and others could be added. For example, a useful metric to add would be connectivity (within line-of-sight and within range) as a function of time. There is currently a report available which shows nodes within line-of-sight as a function of time, but this doesn't tell the whole story.

4.8 NATE VALIDATION

4.8.1 TESTS

NATE, including the DSB database, was evaluated using the evaluation criteria detailed in the validation plan for Space C3 Planning as a yardstick. DSB describes an attack by North Korean forces on South Korea. The North Korean headquarters signals the pre-positioned tank battalions to move toward the South Korean border. In concert with the arrival of the North Korean forces at the border, a SCUD attack is initiated. During the initial phase of the scenario, the North Korean headquarters signals the tank battalions and the national signal intelligence assets intercept the transmission which is then forwarded to the data processing and C2 nodes in CONUS. The interception of this information triggers data being sent back to the theater Air Operations Center to schedule the launch of theater based assets. Concurrently, as national electro-optic and synthetic aperture radar satellites make passes over the area, data is collected and forwarded to CONUS for data processing and as inputs to the C2 nodes. This information is then forwarded to the Air Operations Center. Once the theater airborne assets reach operational altitudes, the information they gather is sent back to the Control and Reporting

Center. When the SCUD is launched from North Korea, it is detected by the DSP satellites. These satellites forward the data to the CONUS Missile Warning Center. The CONUS Missile Warning Center sends data to the theater Air Operations Center which forwards the data to the Patriot battery.

4.8.2 RESULTS

From a communications perspective, the NATE model framework (not including the database) is acceptable for Space Command's intended purpose with a couple of small exceptions. First, the data rate cannot be set by the user for hardwired links currently. Second, there is a bug in the STRATC2AM software which causes datapoints to be recorded out of sequence in some postprocessor reports, resulting in errant statistical plots.

Of course, there is always room for improvement in any model and NATE's communications model is no exception. While it is not absolutely necessary, NATE would also benefit from adding more modern link types (e.g. ethernet, FDDI, ATM). Clever users can probably emulate these types of links with the current NATE, but it would require the user to be a communications expert. Similarly, modeling of techniques for multiple access to resources such as FDMA, TDMA, and CDMA would be useful. Modeling antenna position and slewing would increase realism for satellite and other line-of-sight communications. Currently, the assumption is made that antennas are always pointing in the right direction. At a minimum, the user should be able to choose some nominal antenna slewing time. Also, since modern military communications satellites such as DSCS rely on MBAs for beam steering, beam shaping, and jamming suppression, an MBA model would be a good addition.

Unfortunately, when we define NATE to include the DSB database, the model is less than acceptable. In a way, it doesn't seem fair to criticize the DSB database, because it was not developed with realism in mind. It was designed to provide an unclassified demonstration of NATE's capabilities and it does that. However, simulation results are only as good as the data that a user puts into a model and Space Command needs to know that simulation scenarios using the DSB database will not give them realistic results. In fact, the DSB database would not provide results at all unless the scenario is run using the unclassified development version of NATE which ignores communications link calculations. When this was discovered, SAIC modified the database so that it would at least provide some results when run on the official version of NATE. While the DSB database may be a fair representation of the types of resources that will be available in future conflicts (Milstar, DSCS, imaging satellites, Patriot batteries, etc.) the resources are not accurately described in the database. Without discussing exact specifications of the operational systems, we can still give a few examples. The operating frequencies defined for the Milstar transmitters are off by anywhere from 40 to 70 GHz. This has very real implications in trying to predict the attenuations on uplinks and downlinks due to the atmosphere and rain. In general, the bandwidths specified are quite wide. In fact, one of the TDRSS satellite transmitters has a 100 GHz bandwidth while the operating frequency is only 7.5 GHz. This is worse than inaccurate. This is physically impossible and NATE should give error messages for mistakes of this nature.

Again, the main point of these examples is to stress that one of Space Command's top priorities regarding NATE needs to be to develop a realistic database of present and projected resources to base their scenarios on. Unfortunately, this is currently a non-trivial task. However, the graphical user interface for the preprocessor with the next version of STRATC2AM should make this a much more manageable process.

The environment is adequately modeled for Space Command's purposes at this time. Environmental effects modeled presently are atmospheric loss, rain loss, jamming, and nuclear burst effects. There has also been talk recently of 'tacticalizing' STRATC2AM to make it more applicable to tactical scenarios. While it is probably not necessary for Space Command's purposes, terrain modeling would greatly benefit the tactical users of STRATC2AM.

From Space Command's perspective, another area where NATE is currently lacking is in sensors. NATE does not currently model sensors. Sensors can be "faked" to some extent by using Omni. Omni allows the user to attach a very simplified model of a sensor to any platform. It allows the user to display a cone with a user specified field of regard to emulate a sensor. However, this is only a visual tool. If a missile flies directly through a sensor's field of regard, nothing will happen. There is no connection between Omni and STRATC2AM allowing the generation of missile warning messages triggered by such an event. Currently, this is a manual process. The user visually detects a sensor event, notes the time, and modifies the STRATC2AM preprocessor file to add missile warning messages at that time. At a minimum, this process could be automated. If more realism is desired, a link could be made to an existing sensor model such as the Mission Effectiveness Model (MEM). Whatever form it takes, a fully functional sensor model should be a fairly high priority for NATE.

Threat resolution and discrimination could currently be emulated by the NATE user similar to way that sensors can be emulated, by a manual process. The main difference would be that the process would be even clumsier. Since messages in NATE do not contain any data, different message types would need to be created to indicate threat resolution and discrimination results. The effort that would be required for NATE to actually model resolution and discrimination would not be trivial, and unless it is a critical issue to Space Command, it would probably not be the best use of scarce NATE development dollars.

Tracking and data fusion cannot be done currently with NATE and the effort associated with adding this capability would be considerable. Algorithms for performing the tracking and data fusion would obviously need to be added. However, there is still the underlying limitation of NATE that messages do not currently contain data. Fortunately, this is probably not a high priority item.

Surveillance and intelligence are also not currently included in NATE, and in all likelihood they never will be. Adding intelligence data to NATE would add not one but two levels of complexity to the model. Not only would it be necessary to pass data in messages, it would be necessary to judge whether or not the data was genuine. Since this is probably beyond the scope of Space Command's intended use for NATE, it was probably unfair to include it as an evaluation factor. On the other hand, it is fair because it would be part of the scenario in real life.

It's notable to point out that we've just mentioned three things in a row which NATE cannot really do presently due partially to its inability to pass data in messages. Therefore, the addition of the capability of passing, interpreting, and acting on data could potentially have very widespread benefits. Despite the fact that STRATC2AM stands for STRATegic Command and Control Architecture Model, it is primarily a communications model. Passing data instead of bits in messages, NATE could become a true C3 model.

Some aspects of electronic warfare (EW) are addressed fairly well while others are not addressed at all. For scenarios which only include the jamming of communications links, NATE's EW model is adequate. Since the jammer antenna is assumed to always be pointing in the right direction, modeling jammer antenna pointing would provide added realism. However, the argument could be made that simplifications resulting in a worst case situation is not necessarily a bad thing. Unfortunately, countermeasures associated with sensors are not addressed at all. Of course, since NATE does not really have a sensor model, this is understandable.

Random processes are handled acceptably in NATE with the exception of determining equipment availability. It's good that the reliability of equipment is addressed, but it's implemented in an awkward manner. Probability of failure is specified for entire classes of equipment instead of individual pieces of equipment. What this means is that every piece of equipment having the same class will fail at the exact same time. Since this is unlikely to ever happen in real life, it seems more realistic to ignore probability of equipment failure altogether. The only way to make this feature work properly is for the user to not use more than one of any one class of equipment, thereby not taking advantage of the convenience of being able to define classes. This should be fixed.

4.9 CONCLUSIONS & RECOMMENDATIONS

The stated purpose of this project was not to do an in-depth validation of NATE's individual models. It was given that these models had been sufficiently validated already. Our prime objective was to determine whether any of the individual models had been "broken" in the process of integrating them together. This was done through extensive testing of the interfaces between the individual NATE components. The results of these tests showed that the individual models had not been "broken", although they were "chipped" in a few places. These "chips" can be found below under the heading of FIXES. We have also provided a frank assessment of NATE's suitability to Space Command's intended use.

While this report is full of NATE's problems, NATE is basically a very sound communications simulation model. This would be a much longer document if we also included everything in NATE that was right. Almost all of the mistakes which were found in the design appear to be relatively minor and definitely fixable. Since a few of the mistakes were in the commercial product Omni, the biggest problems with fixing them may be contractual ones. Its most obvious weak point is a very old-fashioned and user-unfriendly pre-processor user interface and this will be nearly eliminated in the next release of STRATC2AM (due for release any time now). Its next biggest weak point is that realistic databases need to be built up to base simulation scenarios on. The reason why this is such a problem currently is that the existing pre-processor would make this a very time-consuming task best left to STRATC2AM experts. If the new graphical pre-processor is as good as it appears to be, this will be a much more manageable task that could be done by the users instead of the developer. For the intended use of NATE the next highest priority change should be the addition of a real sensor model. At that point, NATE should be fairly well suited for Space Command's purposes.

Of course, there will always be something that a user will want to add to any model and NATE is no exception. Below is a list of enhancements which we would currently recommend for NATE. Beyond the suggestions to build realistic databases and add a sensor model which have already been discussed, we do not recommend them in any particular order. Also, as was mentioned above, a list of recommended fixes is included. Since making a fix to a model means that there is something incorrect in the model which could be compromising simulation results, the fixes should be considered higher priority items.

4.9.1 FIXES

1. The Omni satellite data entry windows should be fixed to accommodate the precision required for standard two-card orbit descriptions.
2. STRATC2AM pre-processor should be fixed to accept user-specified data rates for hardwired links.
3. STRATC2AM post-processor should be fixed to ensure that time sequencing of data points is correct.
4. The irregularities in the plotting Grouped Packet Receipt report in Omni should be fixed.
5. Continuous line graphs in Omni should be fixed to highlight the data points.
6. The error in Omni that causes moving nodes to be interpreted as launched nodes when written to STRATC2AM node definition files should be fixed.
7. The equipment probability of failure model should be fixed to allow the independent failures of members of an equipment class.

4.9.2 ENHANCEMENTS

1. Develop a realistic database that can provide system "building blocks" that users can use with confidence to build scenarios with.
2. Add real sensor capabilities to NATE.
3. Develop a new STRATC2AM post-processor with a point-and-click type interface.
4. Add modern link types (e.g. ethernet, FDDI, ATM, etc.)
5. Validate the "Comet" missile flyout model in Omni to determine if it conforms to the real Comet model. If it does, validate Omni's interface to Comet.
6. Complete the graphical user interface pre-processor which the next version of STRATC2AM will use.
7. If possible, create unclassified "development" version of STRATC2AM which does not ignore link calculations. Perhaps this could be done by eliminating the nuclear codes and only using generic modems (e.g. DPSK, CFSK, NFSK, and CPSK).
8. Add connectivity statistics metric.
9. Add resource multiple access models (e.g. FDMA, TDMA, and CDMA).

10. Add MBA model.
11. Add capability to pass, interpret, and act on data in messages.

4.9.3 DRAFT AIR FORCE INSTRUCTION 16-1002

DRAFT AIR FORCE INSTRUCTION 16-1002 defines validation as a rigorous and structured process of determining the extent to which an M&S accurately represents the intended "real world" phenomena from the perspective M&S use, and may identify improvements. It has two main components: structural validation, which includes an internal examination of M&S assumptions, architecture, and algorithms in the context of the intended use; and output validation, which determines how well the M&S results compare with the perceived "real world."

Although the instruction was not approved at the time of the NATE validation, we feel it is important to comment on the degree to which this validation addressed the above definition. The instruction further specifies the following validation methodologies that could be commented on. Upon signature of the instruction, these comments would allow reference of the report to the instruction.

Face validation process that determines whether an M&S, on the surface, seems reasonable to personnel knowledgeable about the system being modeled; (addressed by this report)

Comparison with historical data or with results from other M&S already accredited for use in similar applications; (not addressed)

Comparison with developmental test data or other engineering test data; (to some extent)

Comparison with operational test data, other field tests, or operational data; (Appendix A - NATE Validation and Accreditation Plan requested Desert Shield Data Bases and the DSP Missile Warning Dissemination data, however the only test data available was the Defense Science Board database and the "simple" scenario.)

Peer review, where functional area SMEs analyze M&S internal representations and outlouts; (accomplished to some extent by AFRL/IF’s review)

Independent review of the entire M&S, or specific functions, by a designated committee or other agents that are independent of the M&S developer. (this was the accomplished by AFRL/IF’s review)

4.9.4 FUTURE NATE CONFIGURATIONS AND VALIDATION

Now that this preliminary validation of NATE has been completed, it is a fair question to ask what happens when NATE’s configuration changes. Will this validation be worthless then? Will it need to be completely redone? That would depend upon how much NATE's configuration changes. Obviously, if all of NATE's components are replaced, then the answer is yes. However, if a component is being added, then the answer is probably no.

For example, if a sensor model is added as has been suggested, NATE's validation would need to be updated, but would not need to be completely redone. First, the sensor model would be validated if it has not already been validated. Second, the interfaces between the sensor model and NATE's other components would be tested. It should be noted that this may not be a trivial step. Ironically, the interfaces for the current configuration were fairly easy to test due to the fact that STRATC2AM is still old-fashioned in many ways. The interface from the post-processor to Omni was via a textual file containing a list of data points. If this interface was by a more direct means, it could have been much more difficult to test. Finally, the new NATE would be evaluated against the Space C3 Planning criteria used for this validation. The evaluation of the new NATE would be done by updating the appropriate parts of the old evaluation.

If STRATC2AM were to be replaced by a different model, the bulk of this validation would need to be redone. STRATC2AM is the heart of NATE and the effects of its replacement on NATE's validation would be widespread. The new communications model would need to be validated if it had not already been validated. Most of the interfaces would be changed, requiring new testing. Finally, much of the functional evaluation (against the Space C3 Planning criteria) would need to be redone to reflect the new communications model.

4.10 APPENDIX A - NATE VALIDATION AND ACCREDITATION PLAN

The NATE Validation and Accreditation Plan was originally prepared by Gregory Hadynski and William Brennan of AFRL/IF. While there have been no changes in its contents for inclusion in this report, it has been reformatted to form an integral part of this report.

4.10.1 OVERVIEW

United States Space Command has recently initiated a program to perform a validation and accreditation of the NUICCS Analyst Technical Environment (NATE) simulation model. The formal validation and accreditation will be performed by AT&T under contract to USSPACECOM with funding by the Defense Modeling and Simulation Office (DMSO). However, by USSPACECOM request, AFRL/IF will work with the Air Force Center for Studies and Analysis Agency (AFSAA) and Science Applications International Corporation (SAIC) to provide an informal validation of the NATE model in the interim. AFSAA is the agency responsible for the development of one of the principal parts of the NATE model, the Strategic Command and Control Architecture (STRATC2AM) model. AFSAA and SAIC have performed validation of the STRATC2AM model as an integral part of its development cycle and will ensure that the NATE validation goes hand-in-hand with their efforts. This plan is an AFRL/IF document and will concentrate on the AFRL/AFSAA/SAIC informal validation effort.

4.10.2 STRATEGY

The NATE validation and accreditation initiative is especially challenging in that NATE itself is a composite of simulation models. NATE consists of four main components: the STRATC2AM C3 simulator, the Omni graphic interface, the COMET missile propagator, and the SGP4 & SDP4 satellite propagators.

It is logical to begin the validation process of a composite model by decomposing the model to its major components and performing a validation process on each of the major components. Fortunately, the individual models that NATE consists of have already been sufficiently validated to eliminate much of this step. STRATC2AM, COMET, and SGP4 & SDP4 have all stood up well to scrutiny over a long period of time. A recent memorandum from Curt Smith of SAIC to Lt. Col. Keith James of USSPACECOM provides an excellent overview of the history of STRATC2AM validation. The STRATC2AM Analyst's Manual provides actual validation data. COMET and SGP4 & SDP4 have been used by NORAD for many years. Since Omni is a graphic interface rather than a model, it cannot really be validated alone. For these reasons, the individual models within NATE will be validated under this effort on an "as needed" basis determined by validation results of the NATE model as a whole.

Having done this, we need to validate the "glue" which holds these models together. Again we can decompose the problem by individually addressing the interfaces between each of the models. Our goal in this step is to ensure that validated data generated by any one of the individual models is accurately transferred to the appropriate model. This step is essentially checking data at the beginning (databases and pre-processors), midpoint, and endpoint (GUI and post-processors) in the NATE environment and provides greater confidence in the eventual output of NATE. While all of the interfaces to be tested are internal to NATE, most of them should be relatively simple to test. The majority of the interfaces are between STRATC2AM and NATE and testing these interfaces would entail a straightforward comparison of the outputs of STRATC2AM to the outputs of NATE.

The last step is to validate the NATE model as a whole. This would be done by using simulation scenarios of complexity adequate for testing NATE's main functions. These scenarios will be provided by USSPACECOM, but could include the detection of ballistic missiles and the dissemination of threat warnings, Desert Storm scenarios, or perhaps everyday conditions. While "real world data" for communications systems is never absolutely repeatable due to the probabilistic nature of communications, our goal is to determine levels of confidence that can be put in NATE results. NATE will be evaluated against an extensive set of criteria to determine these levels of confidence.

4.10.3 VALIDATION PROCESS

For guidance on the validation process, the AFRL/IF validation team will look to Dr. Paul Davis's report on Generalizing Concepts and Methods of VV&A for Military Simulations as well as the most current Air Force guidance and practical experience. The figure shown below is taken from Dr. Davis's report and illustrates the validation process which will be generally followed. Using this philosophy, a combination of empirical evaluation, theoretical evaluation, and other comparisons are used to determine levels of confidence in the model.

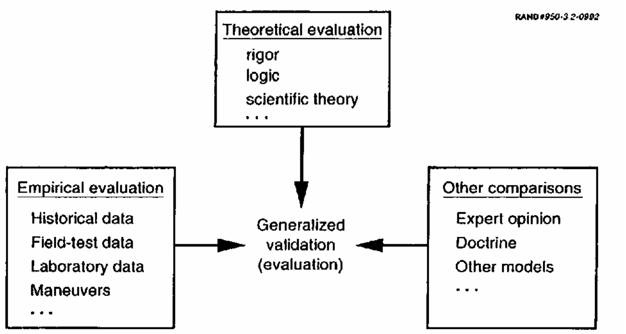


Figure 13 - Validation As A Holistic Process

Additional guidance for AFRL/IF in performing Validation and Accreditation of NATE can be found in Figure 2 Taxonomic View of VV&A (found in the same report mentioned above). The blocks of the diagram which are applicable to this validation program are highlighted. The branch of this tree of importance to AFRL/IF's validation effort is "Generalized Validation (Evaluation)". All three branches under "Generalized Validation (Evaluation)" are of interest to some extent. NATE results will be evaluated against “real world” data where possible. Where “real world” data is not available, results will be evaluated against results from other simulations or against theory. Finally, engineering judgement may be used to determine the reasonableness of results.

This methodology of validation agrees with the direction of USSPACECOM to assure that NATE represents the physical and operational processes it was developed for, through comparison of "one dimensional" results from NATE to those of the independent models. For example, does STRATC2AM give the same answers alone as it does when incorporated in NATE? This also addresses the validation of the Graphical User Interfaces (OMNI and AXIS) in providing complete and accurate data presentation.

NATE will be validated through careful comparison of selected "real world" cases. Independent test data obtained from the Desert Shield Data Bases and the DSP Missile Warning Dissemination are two example test cases to be used for the empirical evaluation of NATE. Assessment of selected NATE output data to determine physical accuracy and soundness will be accomplished with particular attention given to communications delays, dropout, and surveillance coverage as a function of time.

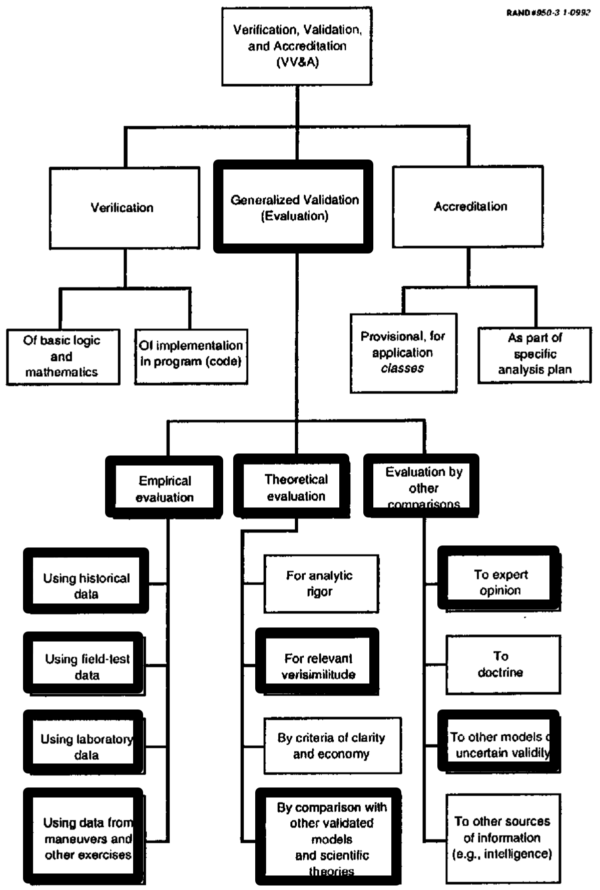

RAND#950-3 1-0992
Verification, Validation, and Accreditation (VV&A)
Verification
Generalized Validation (Evaluation)
Accreditation
Of basic logic and mathematics
Of implementation in program (code)
Provisional, for application classes
As part of specific analysis plan
Empirical evaluation
Theoretical evaluation
Evaluation by other comparisons
Using historical data
Using field-test data
Using laboratory data
Using data from maneuvers and other exercises
For analytic rigor
For relevant verisimilitude
By criteria of clarity and economy
By comparison with other validated models and scientific theories
To expert opinion
To doctrine
To other models of uncertain validity
To other sources of information (e.g., intelligence)

Figure 14 - A Taxonomic View of VV&A

The mechanics of the validation process will follow the process described in the Analytic Tool Box (ATB) Level 1 Face Validity Assessment Plan developed by Martin Marietta Corporation for the National Test Facility (NTF) at Falcon Air Force Base. While the Face Validity Assessment Plan was written as a means to evaluate a simulation model for acceptance into NTF's Analytic Tool Box, it's process is quite applicable to our task of evaluating the NATE simulation model for acceptance into USSPACECOM's planned modeling and simulation testbed. More specifically, for the NATE preliminary validation effort, the use of face validity assessment evaluation matrices will be adopted.

4.10.4 FACE VALIDITY ASSESSMENT EVALUATION MATRICES

The assessment process is made repeatable through the use of a set of evaluation matrices and checklists. These lists capture the primary factors that contribute to the M&S quality and operational effectiveness as well as chart the progress made toward validation. Each primary factor or area is further decomposed into subfactors. The subfactors themselves are evaluated by means of a set of criteria and exploratory questions that either probe the information or directly lead to evaluative answers. While many areas of the matrix are quite explicit, others are somewhat more difficult to assess and may require further refinement by the team. These subfactors are evaluated by the team through a consensus process and then combined into an overall evaluation. This consensus approach significantly reduces the subjectivity of the assessment.

A non-numerical rating system with four possible scores ("Exceeds Guidelines," "Meets Guidelines," "Minor problems Noted," and "Major Problems Noted") is used to rate the primary factors and subfactors for operational effectiveness. The principal objection to using a numerical scoring/weighting system to combine all of the ratings into a single "overall score" is that an "overall score" does not capture the variations in the importance and completeness of the evaluation that can be performed in the separate factors and therefore may present a misleading picture of results of the process.

The evaluation material is partitioned into two different matrices, with each including and evaluation criteria:

a. Operational Effectiveness-Integration/Data management
b. Operational Effectiveness-Space C3 Planning (Not a standard matrix in the TB Face Validity Assessment Plan. Created for the USSPACECOM mission)

4.10.5 TASK DESCRIPTIONS

Task 1:

A plan for performing the informal validation of the NATE model will be developed. The document which you are reading constitutes the most current version of the plan. It will be revised throughout the effort as required. The final version of this plan will be incorporated, along with the validation results, into an AFRL/IF Technical Report upon completion of the effort.

Included in this task will be AFRL/IF efforts to learn the NATE system and its associated components.

Feedback from US Space Command is critical for the completion of this task and the consequent execution of the remaining tasks. Approval of the plan must be expedited to avoid schedule impacts. Since the evaluation criteria should be closely tied to the customer's intended use of the model, concurrence with Space Command is very important. Scenarios of interest to Space Command should also be identified during the planning period.

Task 2:

Individual models which constitute the NATE model will be validated as required. Since the individual models have already undergone considerable validation, activity in this task is expected to be limited.

Task 3:

The interfaces which bind the individual NATE models will be validated. The NATE interfaces which will be validated are listed in Table 1. The evaluation matrix and accompanying evaluation criteria for this task can be found in Appendix A.

MAIN NATE INTERFACES:

- Omni to STRATC2AM
  - Missile entry
  - Platform entry
  - Sensor entry
  - Site entry
  - Moving object entry
  - Communications control file entry
  - Node definition file entry
- STRATC2AM to Omni
  - Communications statistical graphs
    - Graph time delay by message type
    - Graph link message count
    - Graph availability
    - Graph correct message receipt
    - Graph package receipt
    - Graph resource utilization
    - Graph link loading
    - Graph link utilization
    - Graph RF performance
      - Graph probability of receipt
      - Graph signal to interference
      - Graph absorbance
      - Graph scintillation
      - Graph decorrelation
  - Show links
  - Link filters
  - Link shading
- Omni to COMET
  - Missile records
- COMET to Omni
  - Missile positions
- Omni to SGP4 & SDP4
  - Ephemeris descriptions of satellites
- SGP4 & SDP4 to Omni
  - Satellite positions

Table 1. Main NATE Interfaces

Task 4:

A validation of the NATE model as a whole will be performed. The exact scenarios to be simulated are TBD at this time. They will be provided by USSPACECOM. The evaluation matrix and accompanying evaluation criteria can be found in Appendix B.

Task 5:

An AFRL/IF Technical Report will be written and published which will document the validation plan, test results, and quantitative levels of confidence in the NATE model.

Schedule:

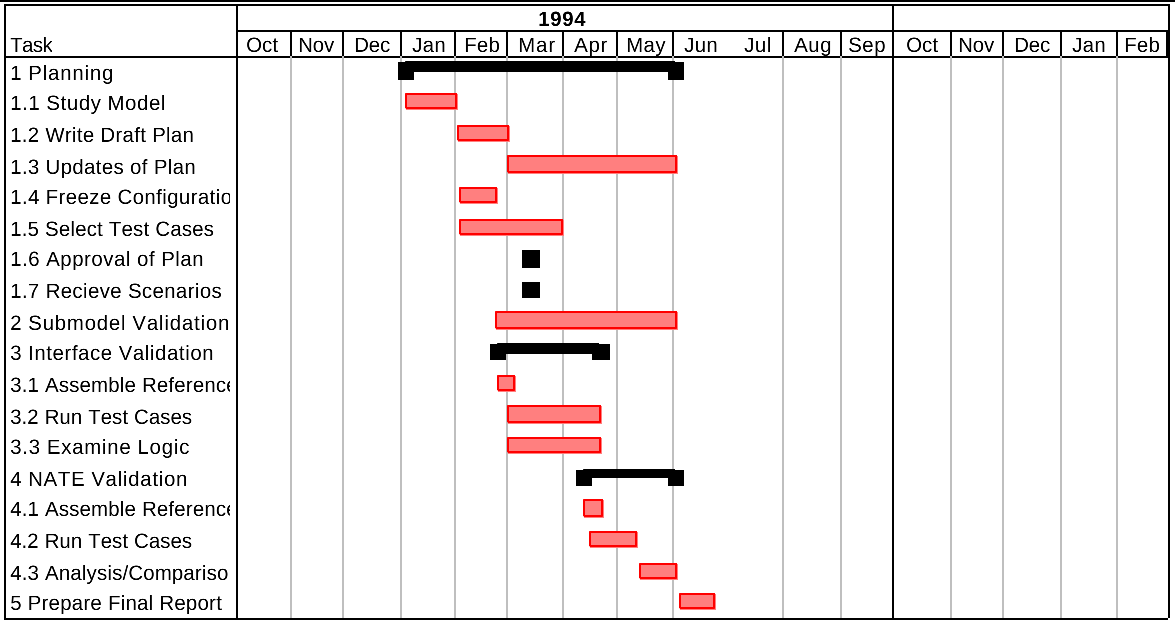
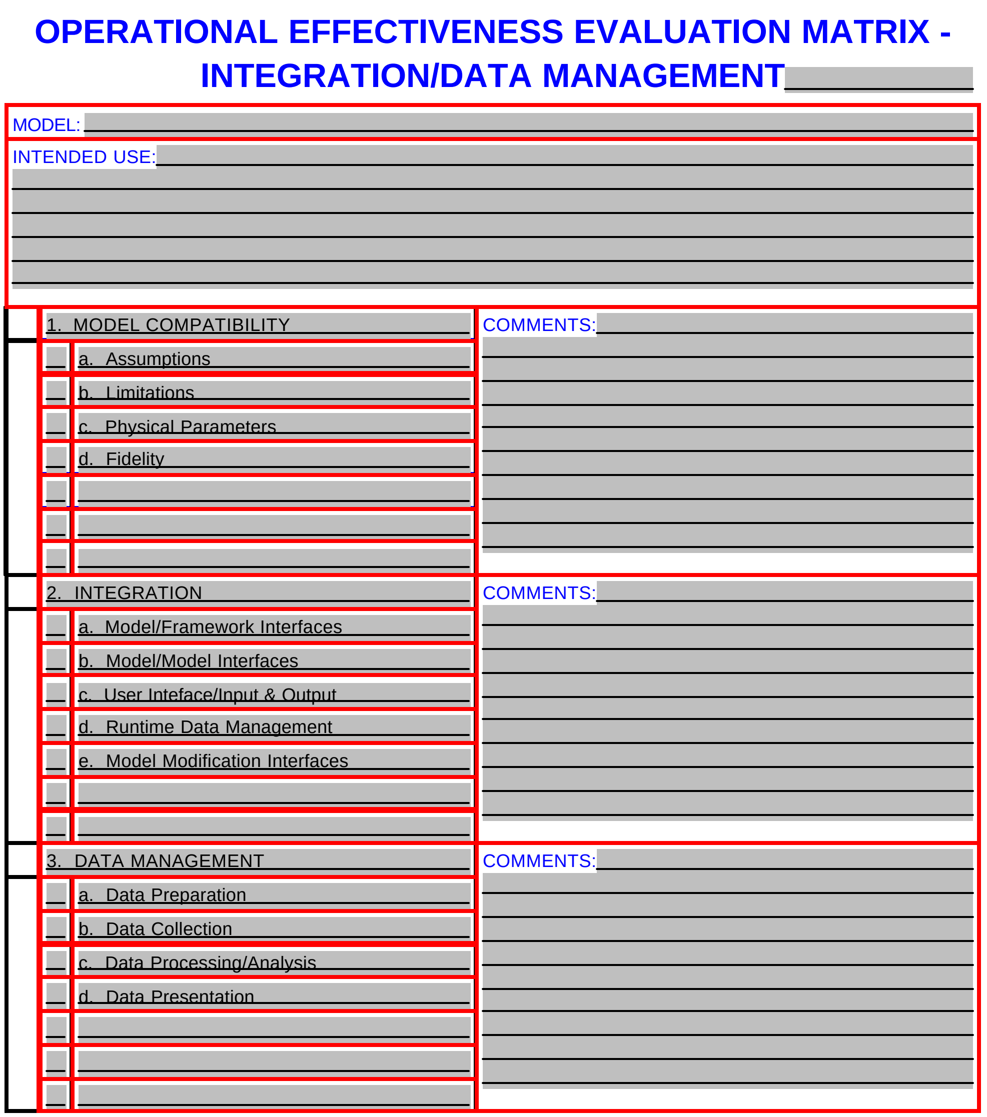


Figure 15 - NATE Validation Schedule

4.10.7 APPENDIX A - EVALUATION MATERIAL FOR OPERATIONAL EFFECTIVENESS-INTEGRATION/DATA MANAGEMENT



OPERATIONAL EFFECTIVENESS EVALUATION MATRIX - INTEGRATION/DATA MANAGEMENT ____________

MODEL: ____________

INTENDED USE: ____________

1. MODEL COMPATIBILITY

__ a. Assumptions

__ b. Limitations

__ c. Physical Parameters

__ d. Fidelity

__ ____________

__ ____________

__ ____________

COMMENTS: ____________

2. INTEGRATION

__ a. Model/Framework Interfaces

__ b. Model/Model Interfaces

__ c. User Inteface/Input & Output

__ d. Runtime Data Management

__ e. Model Modification Interfaces

__ ____________

__ ____________

COMMENTS: ____________

3. DATA MANAGEMENT

__ a. Data Preparation

__ b. Data Collection

__ c. Data Processing/Analysis

__ d. Data Presentation

__ ____________

__ ____________

__ ____________

COMMENTS: ____________

Figure 16 - Integration/Data Management Evaluation Matrix

4.10.7.1 EVALUATION CRITERIA FOR OPERATIONAL EFFECTIVENESS - INTEGRATION/ DATA MANAGEMENT

1. MODEL COMPATIBILITY. For an integrated phenomenology model framework, the various constituent M&S should be able to represent the phenomenologies at the level of detailed resolution expected in order to be used as intended.. The M&S, when integrated together to form a composite solution, should have data interfaces and source data commonality that are compatible in accuracy and functional use. If one model provides macro level results while another provides micro level results, one cannot expect to achieve micro level results when used in composite. The integrated model framework should also provide compatibility across the family of constituent M&S with regard to assumptions and limiting constraints. If assumptions are in conflict, or limitations of one model are being violated by the use of another model, then the results of model integration have less value. It should be possible through model demonstration, interviews with developers and IV&V personnel, and review of documentation to obtain a reasonably good subjective understanding of the level of model compatibility within the integrated model framework. For the model to "meet guidelines" the integration of the constituent M&S must be compatible in assumptions, limitations/constraints, use of physical parameters, and there must be consistency between model and real object fidelity.

a. Assumptions

How compatible are the model assumptions with respect to the functionality, performance, and intended environment being modeled?

How adequate are assumptions regarding intended use and accuracy of model results?

How adequate are assumptions regarding use and accuracy of source data?

How adequate are assumptions regarding model integration to achieve composite results?

How adequate are assumptions regarding the model representation fidelity of phenomenologies?

b. Limitations

How compatible are the model limitations with respect to the functionality, performance, and intended environment being modeled?

How clearly are the limitations stated?

What impact do the limitations have on use of models for critical decision-making?

What are the conditions under which limitations apply?

What is the level of fidelity between phenomenology and models?

c. Physical Parameters

How compatible are the physical parameters with the assumptions, limitations, and phenomenologies of the models being used?

What is the relationship of physical parameters to phenomenology models?
What is the source of physical parameter values?
How is the integration of physical parameters accomplished across various models?

d. Fidelity

How trusted is the model fidelitv- the modeling results compared with actual phenomenologies being modeled?
Is there similarity of results where functional redundancy exists across models?
Was there sufficient certification/validation of model source data?
Was there sufficient certification/validation of model algorithms?
Is there consistency of model results with measured phenomenology data?

2. INTEGRATION. For an integrated phenomenology model framework, the various constituent M&S must be adequately integrated through model to model interfaces and model to framework interfaces. The integration depends upon the data used by the M&S (source- and scenario- specific), shared among M&S, and collected from model execution for execution display and post-execution analysis. The integrated model framework should be reviewed to ensure an appropriate level of detail is provided by each of the M&S as integrated within the framework. The documentation should explicitly describe all interfaces between the integrated model framework and external sources, between any one model and the integrated model framework, and between cooperating M&S. Demonstrations should illustrate how M&S are integrated together to achieve desired results. For the model to "meet guidelines" there must be sufficient model/framework scene generation interfaces, model/model interfaces for operational use, user interfaces for input and output of information, and supporter interfaces for accomplishing model modifications.

a. Model/Framework Interfaces

How adequate are the models (e.g., phenomenology- terrain, cloud, horizon, earthlimb, aurora, space, nuclear, boost, midcourse, vent, debns) within the integrated model framework (e.g., scene generation- framework scenario specifications, transformation, construction component, framework phenomenology model data libraries component, framework model execution component, framework scene generation component)?
Is there consistency in the level of detail for model/framework interfaces?
Is there consistent data fidelity and source of data?
Are there sufficient modes of operation (e.g., scene generation- models and flames, models only, frames only, information) that affect which interfaces are used and the fidelity of the results?

b. Model/Model Interfaces

How adequate are the interfaces between models or modules?
Is there consistency in the level of detail for model/model interfaces?

Is there consistent data fidelity and source of data?

What capabilities are provided for model data (input, output) to be used by other models without excessive assumptions, constraints, and transformations?

c. User Interface/Input & Output

How adequate are the user interfaces for accomplishing an efficient scenario construction, model execution, and report generation?

How adequate is the scenario generation interface provided the human user?

How adequate is the framework and its model components to efficiently handle data input during execution of the models?

How adequate is the framework for entry and modification of various types of data including: physical data constants, threat data, phenomenology data?

How adequate are the display interfaces: pre-execution, execution, and post-execution?

How adequate is the framework in providing a user interface to specify the form and content of model results for use in analysis?

d**.** Runtime Data Management

How adequate is the management of data (input, display, and output) during the execution of the selected set of models **within the** integrated model framework?

How efficient is the data management; e.g., does it take a long time to enter data?

Does it take a long time to process data?

Does it use excessive machine and human resources to complete its modeling function?

Is the information presented during execution simple and understandable to the human operator?

Is there sufficient consistency in the information presented for input, display, or output?

Is there capability provided for storage of data for later use during analysis?

e. Model Modification Interfaces

How adequately does the model support modifications to its component elements?

Is there allowance for modifications to physical parameters (constants)?

Is there allowance for modifying accuracy levels in results?

Is there sufficient means provided for modifications to enhance models?

Is there sufficient means provided for modifications to correct defects?

Is there sufficient means provided for modifications to adapt to changes in threat, environment, or other model/framework interfaces?

Is there sufficient means provided for modifications to integrate with new models?

3. DATA MANAGEMENT. For an integrated phenomenology model framework, the various constituent M&S should consistently handle the management of data flowing into and out of the integration framework. These individual M&S may not have originally been designed without such integration considered, so it is imperative that upon integration the data management be baselined and forrealized. The efficient management of the data through the use of preprocessors, scenario generators, report generators, off-the-shelf analysis tools, and databases can aid in providing a synergistic data management capability. This synergism depends upon how well common data requirements have been identified, how well the data is prepared prior to model execution, the mechanisms available during execution to- capture significant data necessary for other model use and/or post-execution analysis, the analysis processes used to derive information from the data collected, and the presentation' representations of the data to capture the significance of the modeling results. Models without the mechanisms to adequately manage input, nantime internal, nantime display, and output data will not be able to adequately support the required technical decision-making process. For the model to "meet guidelines" it must adequately use procedures for collecting, storing, displaying, and analyzing data during the preparation, runtime execution, and postexecution processing activities related to the integrated model framework.

a. Data Preparation

How adequate are the provisions in the process for preparing data used by the integrated model framework and its constituent models?

Are input data matrices sufficient?

Is there a user interface for building the appropriate input data files?

Are there adequate preprocessors, such as scenario construction tools, scenario transformation tools, and certified data sources?

b. Data Collection

How adequate is the process for collecting data used by the integrated model framework and its models for inputs and post-execution analysis?

Are there automated data collection instrumentation within the models?

Is there provision of controls for selecting and limiting the data collected and the form of the data collected?

c. Data Processing/Analysis

How adequate is the processing and analysis of data collected during the execution of the integrated model framework and its models?

How well have off-the-shelf tools been integrated into the data processing/analysis process?

Is there sufficient control of the processing/analysis methods to properly reflect desired fidelity of results?

How adequate is the use of processing/analysis tools during pre-execution, execution, and post-execution?

d. Data Presentation

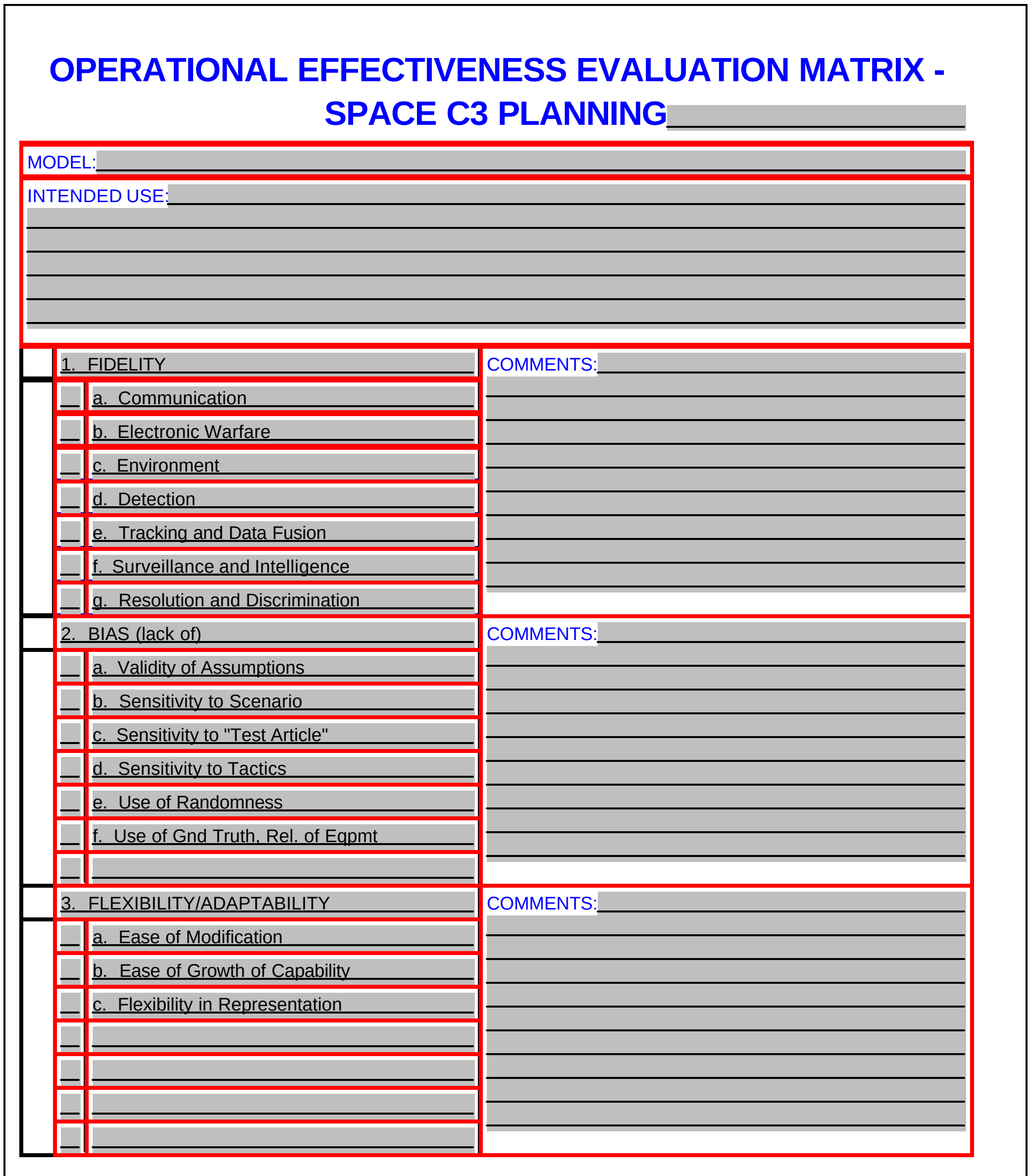
How adequate is the process of presenting data during execution (e.g., data displays) and post-execution analysis?

Are there sufficient graphical capabilities during preparation, execution, and analysis?

Is there capability to provide levels of data presentation from summary to very detailed depending upon the audience?

Is there means for automated integration of model execution, data collection, and data presentation?

4.10.8 APPENDIX B - EVALUATION MATERIAL FOR OPERATIONAL EFFECTIVENESS-SPACE C3 PLANNING



OPERATIONAL EFFECTIVENESS EVALUATION MATRIX - SPACE C3 PLANNING

MODEL:

INTENDED USE:

1. FIDELITY

COMMENTS:

a. Communication

b. Electronic Warfare

c. Environment

d. Detection

e. Tracking and Data Fusion

f. Surveillance and Intelligence

g. Resolution and Discrimination

2. BIAS (lack of)

COMMENTS:

a. Validity of Assumptions

b. Sensitivity to Scenario

c. Sensitivity to "Test Article"

d. Sensitivity to Tactics

e. Use of Randomness

f. Use of Gnd Truth, Rel. of Eqpmt

3. FLEXIBILITY/ADAPTABILITY

COMMENTS:

a. Ease of Modification

b. Ease of Growth of Capability

c. Flexibility in Representation

Figure 17 - Space C3 Planning Evaluation Matrix

4.10.8.1 EVALUATION CRITERIA FOR OPERATIONAL EFFECTIVENESS for SPACE C3 PLANNING

These criteria are intended for rating medium-fidelity architecture models.

1. FIDELITY

a. Communications
Communications in the model should have realistically described equipment and comm processes which are of medium fidelity as noted below. In addition, the comm processes must be integrated with the other processes (i.e., affect and are affected by them). An overall score of "exceeds" guidelines requires that the simulation model the type, timing, and failure of all messages (data) as a function of capacity, traffic, routing, and environment. An overall score of "meets" guidelines require that the model represents the explicit transfer of information for key decisions with delays and connectivity explicitly modeled. An overall score of "minor problems noted" should be used when the model represents communication delays, but with some deficiencies, inconsistencies, omissions, or incomplete integration with other processes. An overall score of "major problems noted" should be used only connectivity is modeled or communications are not integrated with other processes.

b. Electronic Warfare
Electronic Warfare (EW) is an increasingly important aspect of air warfare. Effects of countermeasures should be explicitly included. An overall score of "exceeds" guidelines requires that the simulation explicitly model sensors, comm, and weapons for both sides, reactively. Sensor and communications degradation is based on accurate geometric modeling of interference (jamming and self-jamming). An overall score of "meets" guidelines require that the model represents major effects of countermeasures to key sensors, comm, and weapons in a balanced fashion. Uses variables and mechanisms with due regard to physical law. An overall score of "minor problems noted" should be used when the model represents some effects, but (for example) in scripted form, without explicit modeling of physical effects. An overall score of "major problems noted" should be used ff EW is one-sided only or is not included.

c. Environment
Environmental factors should be included in a medium-fidelity simulation. These factors should include phenomenology which affect communications links. An overall score of "exceeds" guidelines requires that the simulation model such phenomenology as: atmospheric attenuation, attenuation due to rain, jamming, nuclear burst effects, dust storms, smoke, clouds, wind, temperature, vegetation, man-made terrain features, and dynamic terrain. The simulation should also include environmental effects on ground equipment, degradation of sensors, and weapon probability-of-kill (Pk). Terrain

features, both man-made and natural should be well represented. An overall score of "meets" guidelines require that atmospheric attenuation, attenuation due to rain, jamming, and nuclear burst effects be accurately modeled. An overall score of "minor problems noted" should be used when the model ignores some of the effects of the environment on communications links. An overall score of "major problems noted" should be used if the effects of the environment are ignored or incorrect.

d. Detection

Detection is a necessary component of a meaningful engagement model. Model capabilities should consider sensors and their supporting systems, and interface them effectively into support of other systems functions. An overall score of "exceeds" guidelines requires that the simulation model accurately model all sensor types' performance at varying range in the presence of natural and threat environments, with due consideration to sensor capabilities, field of regard, field of view, and sensor slew rates. Sensor modeling should consider, where significant: sensors controlled by sensor manager(s); sensor loading and power management; sensor, processor and platform reliability; sensor processing and delays due to sensors. An overall score of "meets" guidelines requires that the model should be able to depict sensor performance with minor simplifying assumptions, based upon environment and sensor capability. Field of view, field of regard, and slew rates are modeled. Sensors are controlled by sensor manager when necessary. An overall score of "minor problems noted" should be used sensor functionality is over-simplified function of environments and capabilities. Effective use of multiple sensor or overloading of narrow field of view sensors is not addressed. An overall score of "major problems noted" should be used if the model does not include major sensor types.

e. Tracking and Data Fusion

Threat object trajectory (current, projected, and in some cases, original) is a principal input to allocation and engagement functions; it must be done to the level of fidelity required within the simulation to provide a credible basis for subsequent determinations. An overall score of "exceeds" guidelines requires that the simulation model should be able to form credible tracks for threat objects based on sensor capabilities and fusion of such data from all sensors viewing the threat, Tracks should include error covariances for position and velocity. Filter(s) should be provided to refine tracks and predict trajectory, launch, and impact points. An overall score of "meets" guidelines requires that the model be able to form tracks based upon sensor capability, and consider data from all sensors to reduce uncertainties. Tracks need to include position co-variance error and impact point prediction. An overall score of "minor problems noted" should be used if track formation is solely based on time of viewing of the threat and/or track quality is based on sensor type only and not on geometries. Track prediction uncertainty is not provided. An overall score of "major problems noted" should be used when the tracks are based on perfect knowledge.

f. Surveillance and Intelligence

Long term and global information provides the background for situation assessment-Modeling should at desired time requires information management. An overall score of "exceeds guidelines" requires that all indicate quality and timeliness of information. Provision of correct information to appropriate game elements sources of information included in the model should reflect the effects of overload, bottlenecks, environmental variations, conflicting information content, data fusion and so on. A capability is provided to model false information on decoys and misidentified elements. An overall score of "meets guidelines" requires that the impact of principal information sources is explicitly modeled. Delays are accounted for. An overall score of "minor problems noted" indicates that some information necessary for decisions is not explicitly modeled, is instantaneously available to all assets, or has no impact on the decision process. An overall score of "major problems noted" indicates that surveillance and intelligence information is not explicitly considered.

g. Resolution and Discrimination

Determination of the number of threat objects (resolution) and the discrimination of real targets from associated objects (discrimination) is a vital part of the ballistic missile defense, affecting both the use and waste of interceptors and the quality of sensors needed. An overall score of "exceeds" guidelines requires that the simulation be able to resolve the number of objects based on accuracy of sensor sensitivities, range to objects, background, dispersion, and size of threat objects. Also able to model discrimination of threat objects based on the actual capability of the sensor. Sensors controlled so that they will continue to view the object until resolution and discrimination is complete. Uses physical discriminators (i.e., tumble rate, etc.). An overall score of "meets" guidelines requires that the model be able to model resolution based on sensor sensitivities, range, and dispersion of threat objects. Able to discriminate objects based on sensor time of viewing and parametric description of threat object signatures. An overall score of "minor problems noted" should be used if track formation is only able to model the effects of resolution and discrimination. An overall score of "major problems noted" should be used when the model does not consider discrimination or resolution.

2. BIAS (LACK OF).

a. Validity of Assumptions

How far removed from “real world” is the simulation due to assumptions?

b. Sensitivity to Scenario

How sensitive is the simulation to scenarios?

c. Use of Randomness

How adequately is randomness handled?

How are random numbers and probabilities used?

How is Monte Carlo handled?
Are there separate random streams by function or only one?

3. FLEXIBILITY/ADAPTABILITY

   a. Ease of Modification
      How adequately is modification of software handled?
      How easily can the software be modified for different analyses?
      Is the user permitted to modify the source code?

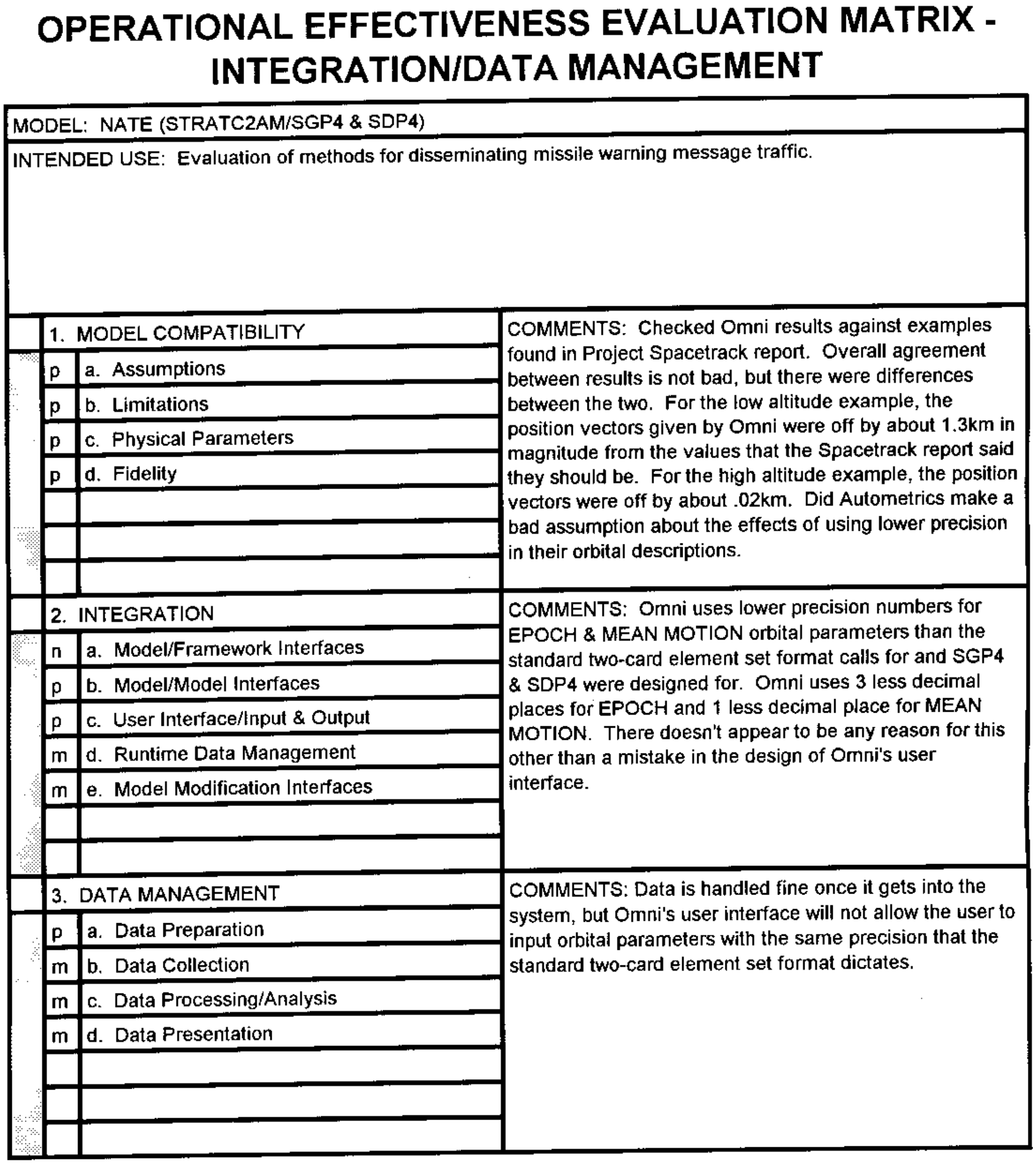
   b. Ease of Growth of Capability
      How adequately is growth capability handled?
      How good is the simulation's potential for growth?
      How good is the simulation's acceptance of new algorithms?
      Is the user permitted to add or change features?

   c. Flexibility in Representation
      How adequately is flexibility of representation handled?
      How well does the software lend itself to large threat analysis?
      How well does the software lend itself to long duration engagements?
      How well does the software lend itself to architecture trades?

   d. Flexibility in Hardware Requirements
      How flexible is the model in terms of hardware requirements?
      How many different computers will it currently operate on?
      Is the model portable to computers not currently supported?

4.11 APPENDIX B - INTEGRATION/DATA MANAGEMENT EVALUATION MATRICES

Key to rating symbols:
n = not applicable
e = exceeds guidelines
m = meets guidelines
p = minor problems noted
P = major problems noted



OPERATIONAL EFFECTIVENESS EVALUATION MATRIX - INTEGRATION/DATA MANAGEMENT

MODEL: NATE (STRATC2AM/SGP4 & SDP4)

INTENDED USE: Evaluation of methods for disseminating missile warning message traffic.

	1. MODEL COMPATIBILITY
p	a. Assumptions
p	b. Limitations
p	c. Physical Parameters
p	d. Fidelity

COMMENTS: Checked Omni results against examples found in Project Spacetrack report. Overall agreement between results is not bad, but there were differences between the two. For the low altitude example, the position vectors given by Omni were off by about 1.3km in magnitude from the values that the Spacetrack report said they should be. For the high altitude example, the position vectors were off by about .02km. Did Autometrics make a bad assumption about the effects of using lower precision in their orbital descriptions.

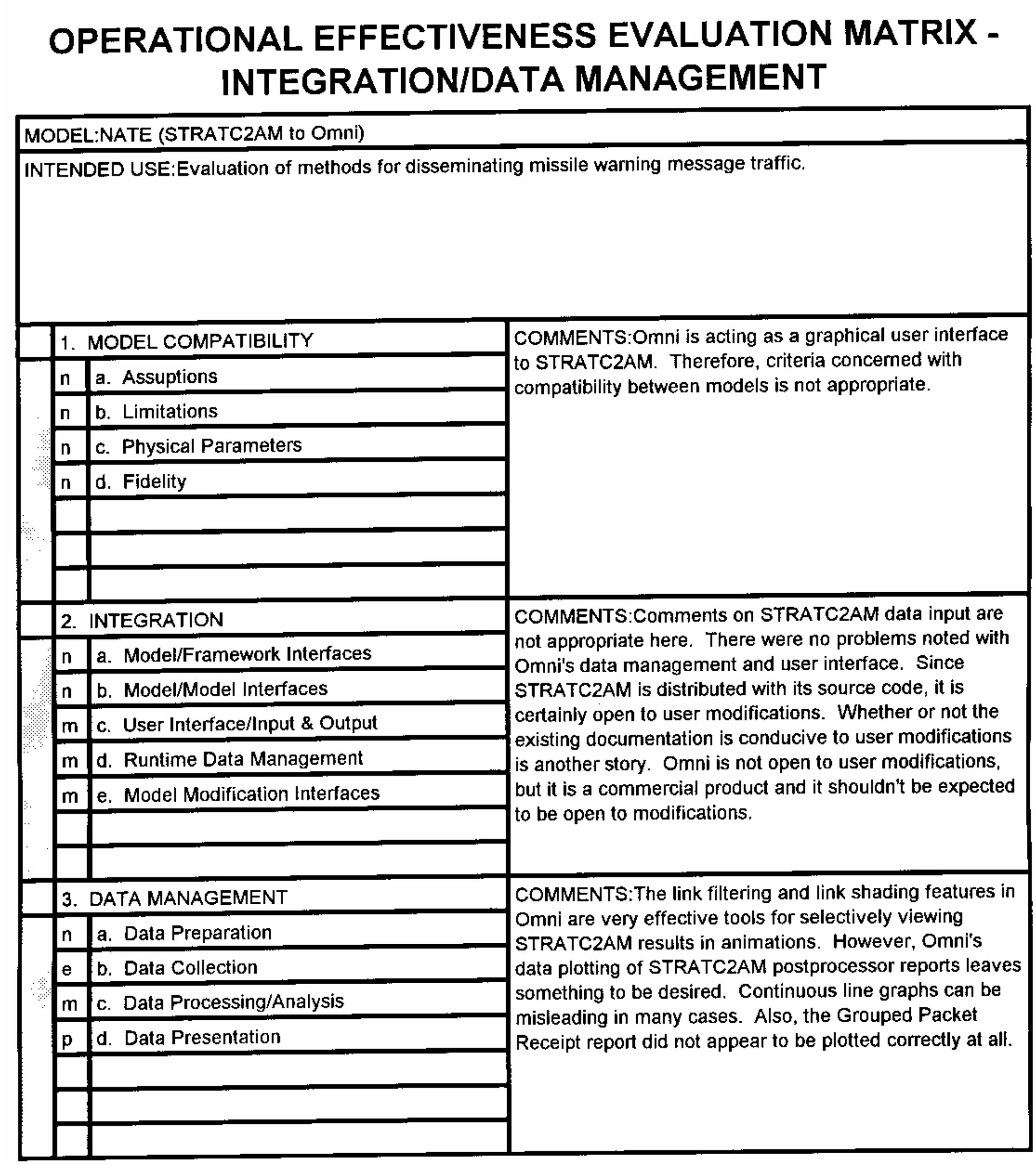
	2. INTEGRATION
n	a. Model/Framework Interfaces
p	b. Model/Model Interfaces
p	c. User Interface/Input & Output
m	d. Runtime Data Management
m	e. Model Modification Interfaces

COMMENTS: Omni uses lower precision numbers for EPOCH & MEAN MOTION orbital parameters than the standard two-card element set format calls for and SGP4 & SDP4 were designed for. Omni uses 3 less decimal places for EPOCH and 1 less decimal place for MEAN MOTION. There doesn't appear to be any reason for this other than a mistake in the design of Omni's user interface.

	3. DATA MANAGEMENT
p	a. Data Preparation
m	b. Data Collection
m	c. Data Processing/Analysis
m	d. Data Presentation

COMMENTS: Data is handled fine once it gets into the system, but Omni's user interface will not allow the user to input orbital parameters with the same precision that the standard two-card element set format dictates.

Figure 18 - Integration/Data Management Evaluation Matrix (Interface 1)



OPERATIONAL EFFECTIVENESS EVALUATION MATRIX - INTEGRATION/DATA MANAGEMENT

MODEL:NATE (STRATC2AM to Omni)

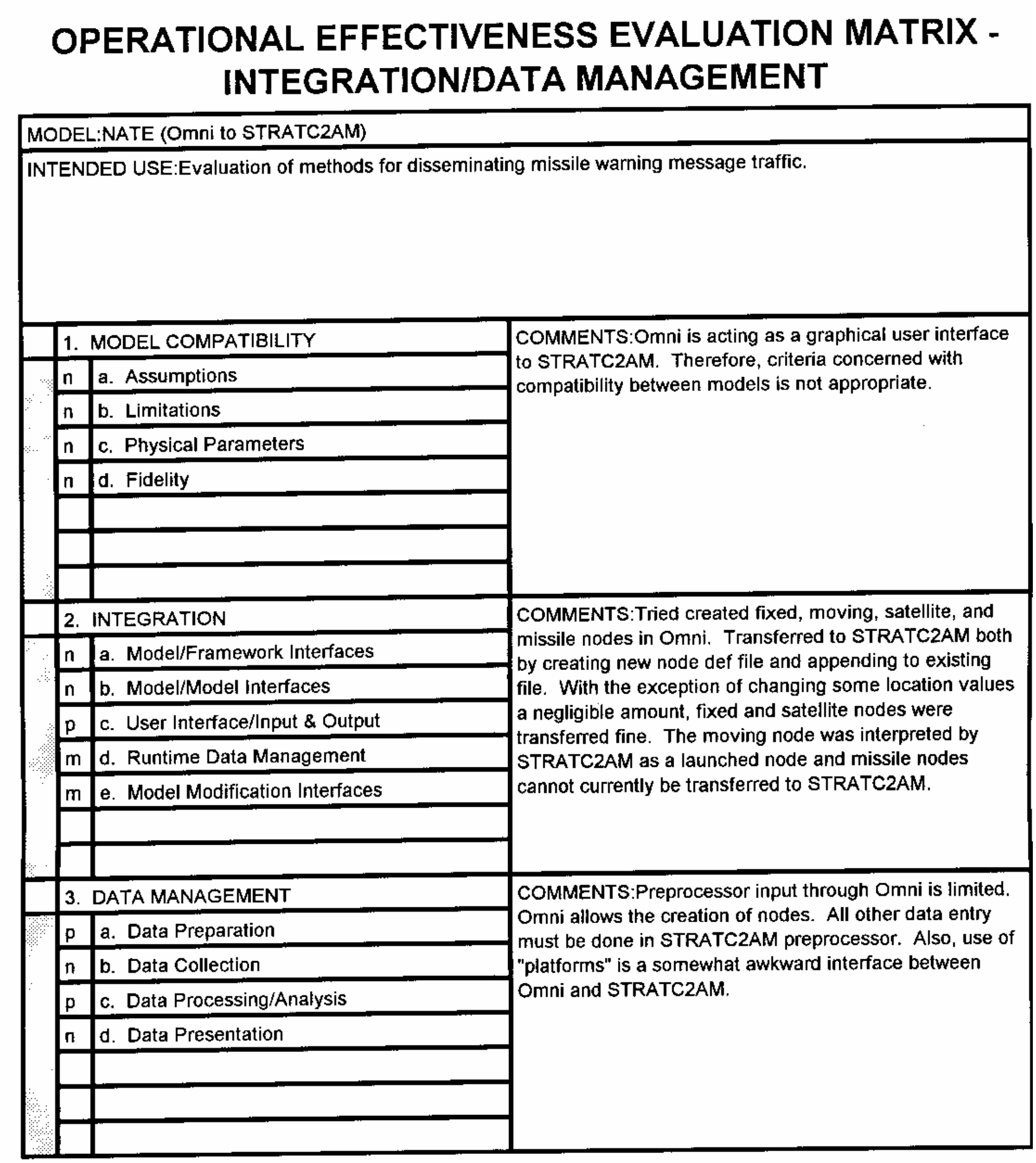
INTENDED USE:Evaluation of methods for disseminating missile warning message traffic.

	1. MODEL COMPATIBILITY	COMMENTS:Omni is acting as a graphical user interface to STRATC2AM. Therefore, criteria concerned with compatibility between models is not appropriate.
n	a. Assuptions	
n	b. Limitations	
n	c. Physical Parameters	
n	d. Fidelity	

	2. INTEGRATION	COMMENTS:Comments on STRATC2AM data input are not appropriate here. There were no problems noted with Omni's data management and user interface. Since STRATC2AM is distributed with its source code, it is certainly open to user modifications. Whether or not the existing documentation is conducive to user modifications is another story. Omni is not open to user modifications, but it is a commercial product and it shouldn't be expected to be open to modifications.
n	a. Model/Framework Interfaces	
n	b. Model/Model Interfaces	
m	c. User Interface/Input & Output	
m	d. Runtime Data Management	
m	e. Model Modification Interfaces	

	3. DATA MANAGEMENT	COMMENTS:The link filtering and link shading features in Omni are very effective tools for selectively viewing STRATC2AM results in animations. However, Omni's data plotting of STRATC2AM postprocessor reports leaves something to be desired. Continuous line graphs can be misleading in many cases. Also, the Grouped Packet Receipt report did not appear to be plotted correctly at all.
n	a. Data Preparation	
e	b. Data Collection	
m	c. Data Processing/Analysis	
p	d. Data Presentation	

Figure 19 - Integration/Data Management Evaluation Matrix (Interface 2)



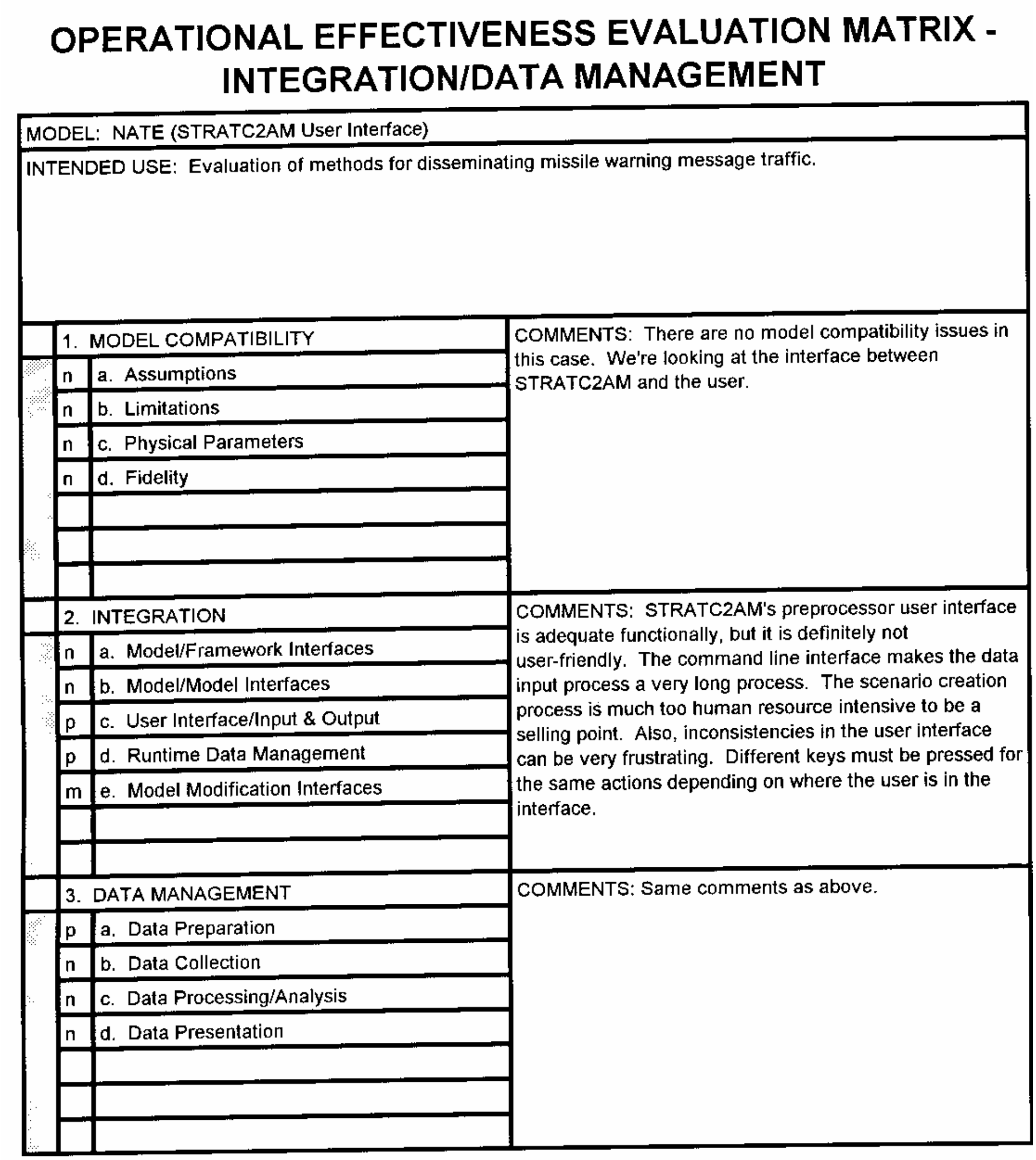
OPERATIONAL EFFECTIVENESS EVALUATION MATRIX - INTEGRATION/DATA MANAGEMENT

MODEL:NATE (Omni to STRATC2AM)

INTENDED USE:Evaluation of methods for disseminating missile warning message traffic.

	1. MODEL COMPATIBILITY	COMMENTS:Omni is acting as a graphical user interface to STRATC2AM. Therefore, criteria concerned with compatibility between models is not appropriate.
n	a. Assumptions	
n	b. Limitations	
n	c. Physical Parameters	
n	d. Fidelity	
	2. INTEGRATION	COMMENTS:Tried created fixed, moving, satellite, and missile nodes in Omni. Transferred to STRATC2AM both by creating new node def file and appending to existing file. With the exception of changing some location values a negligible amount, fixed and satellite nodes were transferred fine. The moving node was interpreted by STRATC2AM as a launched node and missile nodes cannot currently be transferred to STRATC2AM.
n	a. Model/Framework Interfaces	
n	b. Model/Model Interfaces	
p	c. User Interface/Input & Output	
m	d. Runtime Data Management	
m	e. Model Modification Interfaces	
	3. DATA MANAGEMENT	COMMENTS:Preprocessor input through Omni is limited. Omni allows the creation of nodes. All other data entry must be done in STRATC2AM preprocessor. Also, use of "platforms" is a somewhat awkward interface between Omni and STRATC2AM.
p	a. Data Preparation	
n	b. Data Collection	
p	c. Data Processing/Analysis	
n	d. Data Presentation	

Figure 20 - Integration/Data Management Evaluation Matrix (Interface 3)



OPERATIONAL EFFECTIVENESS EVALUATION MATRIX - INTEGRATION/DATA MANAGEMENT

MODEL: NATE (STRATC2AM User Interface)

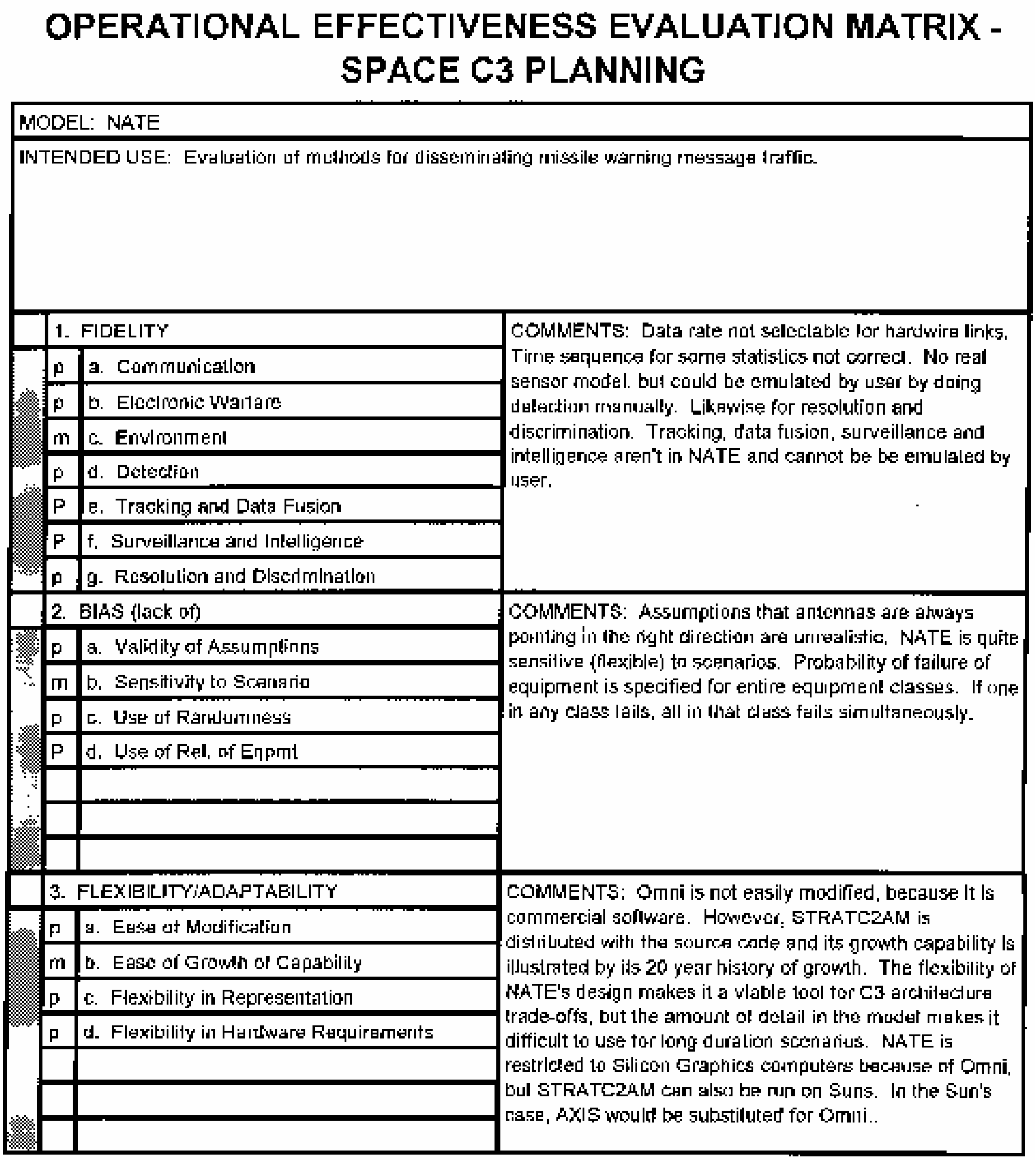
INTENDED USE: Evaluation of methods for disseminating missile warning message traffic.

	1. MODEL COMPATIBILITY	COMMENTS: There are no model compatibility issues in this case. We're looking at the interface between STRATC2AM and the user.
n	a. Assumptions	
n	b. Limitations	
n	c. Physical Parameters	
n	d. Fidelity	
	2. INTEGRATION	COMMENTS: STRATC2AM's preprocessor user interface is adequate functionally, but it is definitely not user-friendly. The command line interface makes the data input process a very long process. The scenario creation process is much too human resource intensive to be a selling point. Also, inconsistencies in the user interface can be very frustrating. Different keys must be pressed for the same actions depending on where the user is in the interface.
n	a. Model/Framework Interfaces	
n	b. Model/Model Interfaces	
p	c. User Interface/Input & Output	
p	d. Runtime Data Management	
m	e. Model Modification Interfaces	
	3. DATA MANAGEMENT	COMMENTS: Same comments as above.
p	a. Data Preparation	
n	b. Data Collection	
n	c. Data Processing/Analysis	
n	d. Data Presentation	

Figure 21 - Integration/Data Management Evaluation Matrix (Interface 4)

4.12 APPENDIX C - SPACE C3 PLANNING EVALUATION MATRIX

Key to rating symbols:
n = not applicable
e = exceeds guidelines
m = meets guidelines
p = minor problems noted
P = major problems noted



OPERATIONAL EFFECTIVENESS EVALUATION MATRIX - SPACE C3 PLANNING

MODEL: NATE

INTENDED USE: Evaluation of methods for disseminating missile warning message traffic.

	1. FIDELITY	COMMENTS: Data rate not selectable for hardwire links. Time sequence for some statistics not correct. No real sensor model. but could be emulated by user by doing detection manually. Likewise for resolution and discrimination. Tracking, data fusion, surveillance and intelligence aren't in NATE and cannot be be emulated by user.
p	a. Communication	
p	b. Electronic Warfare	
m	c. Environment	
p	d. Detection	
P	e. Tracking and Data Fusion	
P	f. Surveillance and Intelligence	
p	g. Resolution and Discrimination	
	2. BIAS (lack of)	COMMENTS: Assumptions that antennas are always pointing in the right direction are unrealistic. NATE is quite sensitive (flexible) to scenarios. Probability of failure of equipment is specified for entire equipment classes. If one in any class fails, all in that class fails simultaneously.
p	a. Validity of Assumptions	
m	b. Sensitivity to Scenario	
p	c. Use of Randomness	
P	d. Use of Rel. of Eqpmt	
	3. FLEXIBILITY/ADAPTABILITY	COMMENTS: Omni is not easily modified, because it is commercial software. However, STRATC2AM is distributed with the source code and its growth capability is illustrated by its 20 year history of growth. The flexibility of NATE's design makes it a viable tool for C3 architecture trade-offs, but the amount of detail in the model makes it difficult to use for long-duration scenarios. NATE is restricted to Silicon Graphics computers because of Omni, but STRATC2AM can also be run on Suns. In the Sun's case, AXIS would be substituted for Omni..
p	a. Ease of Modification	
m	b. Ease of Growth of Capability	
p	c. Flexibility in Representation	
p	d. Flexibility in Hardware Requirements	

Figure 22 - Space C3 Planning Evaluation Matrix

5.0 SUMMARY

This report described the results of three different initiatives pursued under the umbrella of one AFRL/IF in-house effort.

When this effort began, its objective was to develop strategies for routing messages in the next generation MILSATCOM environment. Towards this objective, several routing algorithms were produced in a rapid prototyping fashion using a rule-based approach. The first generation of algorithms developed was tested successfully in a static network environment. The second generation of algorithms was never tested due to a redirection of this overall effort. Testing of the algorithms in a dynamic environment was not done for the same reason.

The second thrust of this effort was to perform several ATM related tasks in support of theatre extension objectives of the Global Grid concept. These tasks were interrupted for approximately a year to pursue the NATE validation initiative under this effort. As a result, some of the ATM related tasks were overtaken by events and were canceled because they were no longer relevant. This report described the results of all those tasks. In the cases where the tasks ended up being canceled, partial results are presented.

The third thrust of this effort was to perform a preliminary validations and accreditation of USSPACECOM's NATE C^3 simulation model. The results of this initiative showed USSPACECOM that their NATE model was basically a very sound communications simulation model. Almost all of the mistakes that were found in the design were relatively minor and fixable. USSPACECOM has since had these mistakes corrected.

6.0 REFERENCES

6.1 ROUTING ALGORITHM DEVELOPMENT

[1] Patrick Winston, *Artificial Intelligence, Second Edition,* Addison-Wesley Publishing Company, Inc., 1984.

[2] James R. Wertz and Wiley J. Larson (editors), *Space Mission Analysis And Design*, Kluwer Academic Publishers, 1991.

[3] Dimitri Bertsekas and Robert Gallager, *Data Networks*, Prentice-Hall, Inc. , 1987.

[4] Harris Corporation, "BM/C3 Communications Network Design," Final Technical Report, 1990.

[5] Harris Corporation, "Communications Networking For SDI Applications," Final Technical Report, 1987.

[6] GTE Government Systems Corporation, "Communications Networking For SDI Applications," Final Technical Report, 1987.

[7] Contel Federal Systems, Inc., "Multiple Satellite System End-To-End Simulator," Final Technical Report, 1990.

[8] Syed A. Shah and Patricia P. Arellano, "AI-Based Network Management for the Defense Switched Network," Military Communications Conference 90 Proceedings, pp 1058-1061.

6.2 ATM ANALYSIS

[1] Leonard Kleinrock, "The Latency/Bandwidth Tradeoff in Gigabit Networks," IEEE Communications Magazine, April 1992.

[2] D.M. Chitre, D.S. Gokhale, T. Henderson, J.L Lunsford and N. Matthews, "Asynchronous Transfer Mode (ATM) Operation Via Satellite: Issues, Challenges and Resolutions," International Journal of Satellite Communications, Vol. 12, 1994.

[3] D.J. Kennedy and M.B. Nakhla, "Burst Error Characterization of FEC Coded Digital Channels," International Journal of Satellite Communications, Vol. 10, 1992.

[4] S.Agnelli, S. Defever, F. Rodriguez and H. Wannman, "Optimization of the Physical Link Performance For ATM Satellite User Access," Proceedings of 3rd European Conference on Satellite Communications, 2-4 November 1993.

[5] J. Butterworth and G. Nourry, "Test Plan for an investigation of the Transport of ATM-Based Traffic By Ku-Band Satellite Bearers," Workshop on the Global Grid concept sponsored by TTCP STP-6, 16 January 1995.

6.3 NATE VALIDATION

[1] Autometrics Inc., *Omni Software User's Manual*

[2] SAIC, *STRATC2AM Version 2.0 Analyst's Manual Vol. 1 Propagation Modeling and Message Processing*, October 1993

[3] SAIC, *STRATC2AM Version 2.0 Analyst's Manual Vol. 2*, October 1993

[4] SAIC, *STRATC2AM Version 2.0 Protocol Rules Version 1.0 Pre-Processor User's Manual*, December 1992

[5] SAIC, *STRATC2AM Version 2.0 Protocol Rules Version 1.0 Pre-Processor User's Manual Addendum*, December 1992

[6] SAIC, *STRATC2AM Version 2.0 Protocol Rules Version 1.0 Simulator User's Manual*, December 1992

[7] SAIC, *STRATC2AM Version 2.0 Protocol Rules Version 1.0 Postprocessor User's Manual*, December 1992

[8] SAIC, *STRATC2AM C3 Models Documentation STRATC2AM Menu Management System User's Guide v1.0*, October 1993

[9] SAIC, *STRATC2AM C3 Models Documentation User's Guide to the NSSA Simulator v3.1*, October 1993

[10] SAIC, *STRATC2AM C3 Models Documentation SKYMAP User's Manual*, October 1993

[11] SAIC, *STRATC2AM C3 Models Documentation STRATQUICK User's Guide*, October 1993

[12] SAIC, *STRATC2AM C3 Models Documentation STRATLINK User's Guide*, October 1993

[13] Foreign Aerospace and Technology Center , *STRATC2AM C3 Models Documentation Analytical X-Windows Interface to Simulations (AXIS) v1.8,* October 1993

[14] SAIC, *STRATC2AM Version 2.0 Preprocessor User's Manual*, August 1993

[15] SAIC, *STRATC2AM Version 2.0 Simulator User's Manual*, August 1993

[16] SAIC, *STRATC2AM Version 2.0 Postprocessor User's Manual*, August 1993

[17] SAIC, *Strategic Command and Control Architecture Model (STRATC2AM) User Training*, 15-19 November 1993

[18] Aerospace Defense Center, *SPACETRACK REPORT NO. 3, Models for Propagation of NORAD Element Sets*, December 1980

[19] SAIC, *NATE Database Development - A Phased Approach* , February 1994

[20] SAIC, *Task Completion Report for the Surveillance Architecture Evaluation Support Task*, 30 November 1993