MPS FOLLOW-ON HARDWARE REPORT

(Preliminary)

(CDRL A1003)

GEODYNAMICS CORPORATION

EUROPE

MISSION PLANNING SYSTEM (MPS)/

INTELLIGENCE COMPUTER SYSTEM (ICS)

SYSTEM SET (MISS)

MPS FOLLOW-ON HARDWARE REPORT

(Preliminary)

(CDRL A1003)

Prepared For:

HQ USAFE/DOO/DOA/INY

Ramstein Air Base, FRG

31 December 1987

Contract No.: F61546-87-C-0041

Prepared By: Geodynamics Corporation

Systems Control Technology, Inc.

A-1

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS NA
2a. SECURITY CLASSIFICATION AUTHORITY NA	3. DISTRIBUTION / AVAILABILITY OF REPORT UNLIMITED
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE NA	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NA	5. MONITORING ORGANIZATION REPORT NUMBER(S) NA
6a. NAME OF PERFORMING ORGANIZATION GEODYNAMICS CORPORATION AND SYSTEMS CONTROL TECHNOLOGY, INC 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION HQ USAFE/DIRECTORATE OF OPERATIONS ANALYSIS
6c. ADDRESS (City, State, and ZIP Code) 2300 GENG ROAD PALO ALTO CA 94303-0888	7b. ADDRESS (City, State, and ZIP Code) HQ USAFE/DOA APO NY 09094-5001
8a. NAME OF FUNDING/SPONSORING ORGANIZATION HQ USAFE/DO/IN 8b. OFFICE SYMBOL (If applicable) HQ USAFE/DO	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F61546-87-C-0041
8c. ADDRESS (City, State, and ZIP Code) APO NY 09094-5001	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
MISSION PLANNING SYSTEM (MPS)/INTELLIGENCE COMPUTER SYSTEM (ICS) SYSTEM SET (MISS)

12. PERSONAL AUTHOR(S)

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
PRELIMINARY	FROM 870910 TO 871230	87 Dec31	97

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
15	03	
12	05	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Automated Planning System

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

This document was the result of three months investigation to determine hardware and software capabilities that would be required to support a USAFE follow-on automated mission planning system (MPS). The report defines the processing requirements for: main memory; mass storage capabilities; external peripherals; and security and reliability considerations. This report was developed by Geodynamics Corporation and Systems Control Technology, Inc for HQ USAFE Deputy Chief of Staff for Operations and Intelligence. Hardware requirements were based primarily upon an empirical sizing analysis of the MPS Follow-On system functions, data bases, input, and outputs as they were currently implemented in two existing mission planning systems: the USAFE Force Level Automated Planning System(FLAPS) and the Tactical Air Forces (TAF) Mission Support System (MSS). The capabilities of these two systems were extrapolated to meet the performance requirements demanded upon the MPS Follow-On system. The study also determined the estimated minimum and maximum hardware processing capabilities to include data transfer speeds that would be required to meet USAFE's mission planning stated needs.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL JACK L. WINGER	22b. TELEPHONE (Include Area Code) 049-6371-47-6911 22c. OFFICE SYMBOL HQ USAFE/DOA

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

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

AAA..................Anti-Aircraft Artillery

ACO..................Airspace Coordination Order

AGL..................Above Ground Level

AOB..................Air Order of Battle

ASCII................American Standard Code for Information Interchange

ATO..................Air Tasking Order

BLIT.................Block Image Transfer

bps..................Bits Per Second

C3CM BMDA............Command, Control and Communications Counter-Measures Battlefield Management Decision Aid

CAB..................Course Arrow Box

CBU..................Cluster Bomb Unit

CHUM.................Chart Update Manual

CMF..................Combat Mission Folder

CONDUIT ....Consolidated Network to Distribute Unit Intelligence and Tasking

CONSTANT SOURCE......Intelligence Data Source

CPU..................Central Processing Unit

DMPI.................Designated Mean Point of Inpact

DPA..................Dynamic Programming Algorithm

DTC..................Data Transfer Cartridge

DTC L/R..............Data Transfer Cartridge Loader/Reader

DTED.................Digital Terrain Elevation Data

EC...................Electronic Combat

EIFEL..................Electronik Information und Fuhrung Systeme Fuer die Einsatzbereitshaft der Luftwaffe (Multi-national automated tactical command, control, and information system)

EO/IR..................Electro Optical/Infrared

EOB....................Electronic Order of Battle

FLAPS..................Force Level Automated Planning System

FOV....................Field of View

FPO....................Overall floating point operation, as used in this report (includes data transfers from local memory)

FPOS...................FPOs per second

GOB....................Ground Order of Battle

I/O....................Input/Output

ICS....................Intelligence Computer System

IINCOMNET ....Intratheater Intelligence Communications Network

IMOM...................Improved Many-on-Many Programs

IP.....................Initialization Point

Kbps...................Kilobits Per Second

LAN....................Local Area Network

LOCE...................Limited Operational Capability Europe

LRU....................Line Replaceable Unit

MIPS...................Million Instructions Per Second

MISS...................MPS/ICS System Set

MOA....................Minimum Observable Altitude

MOB....................Missile Order of Battle

MFLOPS.................Million Floating-point Operations Per Second

MPS....................Mission Planning System ms..................Millisecond

MSS.................Mission Support System

NIB.................Navigation Information Box

NOB.................Navel Order of Battle

nm..................Nautical Mile

ROZ.................Restricted Operating Zone

SAM.................Surface to Air Missile

TAF.................Tactical Air Forces

TDA.................Tactical Decision Aids

TOBS................Terrain Observability

TOT.................Time on Target

UPS.................Uninterruptable Power Supply

USAFE...............United States Air Force Europe

WCCS................Wing Command and Control System

WFZ.................Weapons Free Zone

WORM................Write Once Read Memory

1.0 INTRODUCTION

This report presents the hardware requirements for the United States Air Force Europe (USAFE) Unit Level Mission Planning System (MPS) Follow-On, hereafter referred to as the MPS Follow-On. This report defines the hardware capabilities necessary to perform the MPS Follow-On system functions. Processing requirements, main memory requirements, mass storage requirements, peripheral requirements, as well as security and reliability requirements are specified.

These hardware requirements are based primarily upon an empirical sizing analysis of the MPS Follow-On system functions, data bases, inputs, and outputs as they are currently implemented in two existing mission planning systems, the USAFE Force Level Automated Planning System (FLAPS) and the Tactical Air Force (TAF) Mission Support System (MSS). The capabilities of these systems were extrapolated to meet the performance requirements levied upon the MPS Follow-On. Other existing and planned TAF systems were also studied.

This report is separated into five sections. Section 2 lists reference documents upon which this report is based. Section 3 first introduces the MPS Follow-On system functions and their interfaces. Then the results of the sizing analysis are summarized, namely, the MPS Follow-On hardware requirements. Section 4 describes the system functions and details the sizing analysis that was performed on them. The final section presents the overall system sizing analysis, pulling together the separate sizing analyses of each system function.

2.0 REFERENCES

1. "Analysis of Videodisk and Color Hardcopy Technology for Mission Planning Applications," SCT Technical Report prepared for HQ USAFE/DO, 11 December 1987.

2. USAFE Mission Planning System, Contract F61546-87-C-0041, 20 April 1987.

3. Concept of Operations, Tactical Air Forces Mission Planning System, 3 March 1987.

4. Statement of Operational Need, Tactical Air Forces Mission Support System, TAF, 312-87.

5. Tactical Unit Level Automation Plan, Headquarters Tactical Air Command, 9 July 1987.

6. Mission Support System Analysis, Project No. 6130, Working Paper No. 273323, The Mitre Corporation, 31 July 1987.

7. FLAPS User's Manual Volume II Data Base Specification, October 7, 1987, Contract No. F61546-85-C-0054.

8. PEN-AIDS Upward Compatibility Design Document, October 9, 1984, Prepared for Georgia Technological Research Institute.

9. Intelligence Computer System (ICS) Specifications, Geodynamics Corporation, 29 January 1988, Contract No. F61546-87-C-0041.

3.0 MPS FOLLOW-ON HARDWARE REQUIREMENTS

This section presents the MPS Follow-On hardware requirements based upon the sizing analysis of the MPS Follow-On system functions described in Sections 4 and 5. First, the system functions and their interfaces are introduced. Then the processing, storage, and peripheral requirements are presented. Also presented are the security, reliability and maintainability, and environmental requirements.

3.1 MPS Follow-On System Functions and Interfaces

The MPS Follow-On has been decomposed into 20 separate system functions. The functions and their interfaces are identified and summarized in this section.

The wing and squadrons utilize similar data sources to perform their operations. The Wing Intelligence Computer System (ICS) receives force level inputs--the Air Tasking Order (ATO), the Airspace Coordination Order (ACO), weather information and Intelligence inputs (see Figure 3.1-1). The Intelligence inputs include threat data such as the Electronic Order of Battle (EOB), Air Order of Battle (AOB), Missile Order of Battle (MOB), Ground Order of Battle (GOB), and Naval Order of Battle (NOB). One of the functions of the wing ICS is to receive this data from various operations and Intelligence sources, such as EIFEL, IINCOMNET, CONSTANT SOURCE, and WCCS. Another wing ICS function is to prepare this data for use at the squadrons. The data is transferred from the wing ICS to the squadron ICS systems via the WCCS or CONDUIT communication links. The squadron ICS processes the unit tasking, threat, airspace coordination data and weather information received from the wing ICS and then passes this processed data on to the squadron MPS Follow-On. The MPS Follow-

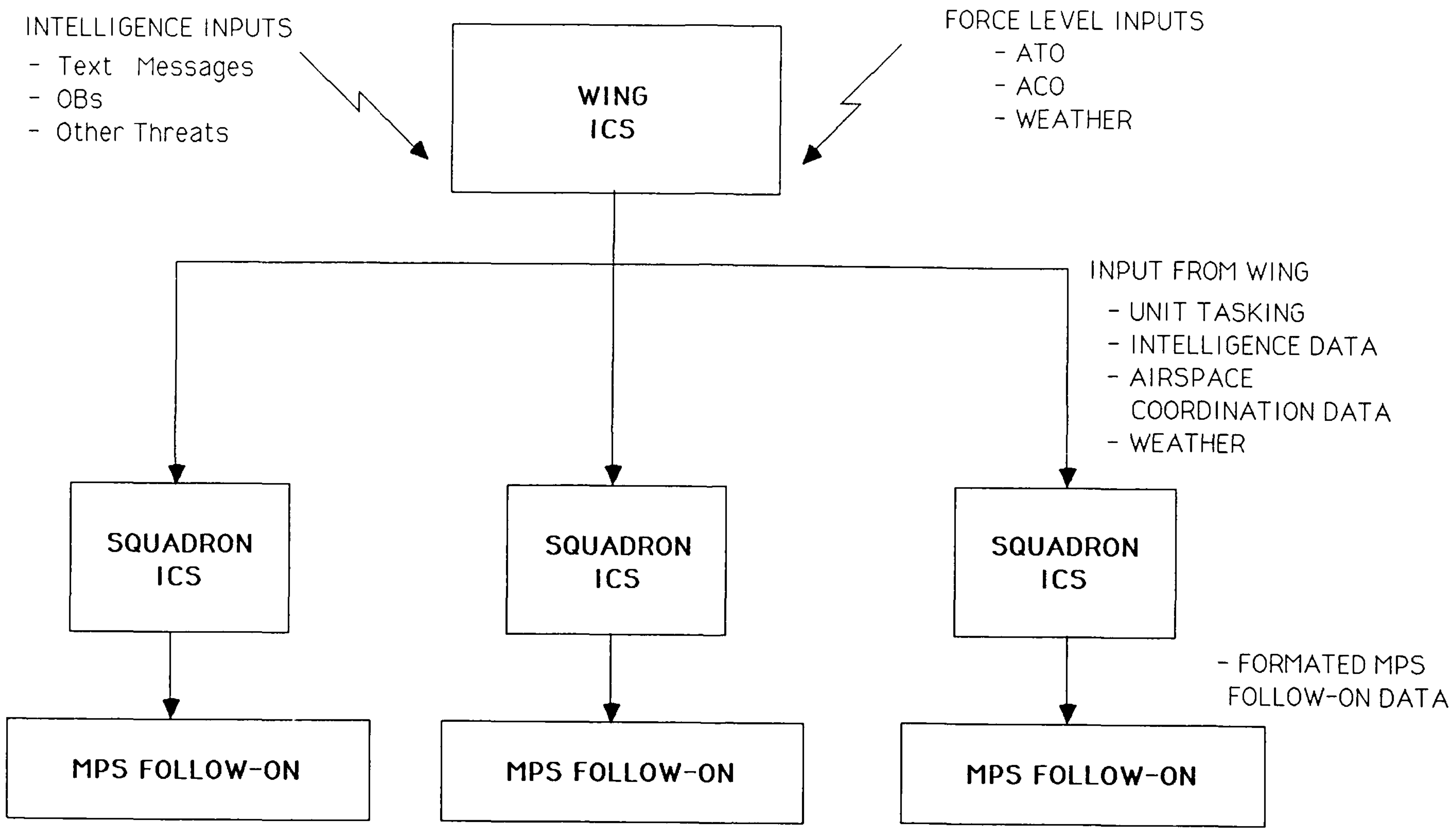

Figure 3.1-1 Wing-Squadron Communications

On uses this data to plan individual routes and to produce combat mission folders. These interactions are shown in Figure 3.1-1.

3.1.1 MPS Follow-On Functional Design

Twenty MPS Follow-On system functions have been identified. They are shown in Table 3.2-1. Each of these twenty functions is described in detail in Section 4.

Table 3.1.1-1 MPS Follow-On Functions

1. Threat Data Input
2. Mission Tasking Input
3. Airspace Coordination Data Input
4. Weather Data Input
5. Relative Threat Lethality Processing
6. Route Generation
7. Route Evaluation and Threat Analysis
8. Flight Plan Generation
9. Combat Mission Folder (CMF) Generation
10. Radar Prediction
11. Electro-Optical/Infrared (EO/IR) Predictions
12. Electronic Combat (EC) Asset Modeling
13. Onboard EC Modeling
14. Three-Dimensional Modeling
15. Conventional Weapons Delivery
16. Nonconventional Weapons Delivery
17. Digital Map and Imagery Display
18. Data Transfer Cartridge Loader/Reader (DTC L/R) Interface
19. User Interface
20. Data Base Management

Figure 3.1.1-1 is a high level interface diagram for the twenty functions listed in Table 3.2-1. The mission planning process would begin with receipt of tasking and the input of intelligence data from the squadron ICS. The intelligence data will be processed into the threat data base. The threat then will be processed into the relative threat lethality grid, or statespace. The statespace is used for route generation and route evaluation. Threat data will only be processed when necessary. A separate threat update will not be necessary for each route. Typically, the same statespace will be used to plan several routes.

Once the statespace has been processed, then the route planning may begin. Based on the received taskings, the route generation function will be executed to generate routes. A combination of optimization algorithms and manual inputs will be used to generate the best possible route. Weather data and airspace coordination data, also received from the squadron ICS, will be input to the MPS Follow-On data base. These inputs will be considered during route generation. Weather can severely impact the effectiveness of Electro-Optic and Infra-red (EO/IR) weapons. Software will be available to predict detection and lock-on ranges for these weapons. Target area planning will be done using conventional or non-conventional weapons delivery software.

The route evaluation function will evaluate the survivability of individual routes. The effects of standoff and onboard EC jamming should be included in the route evaluation.

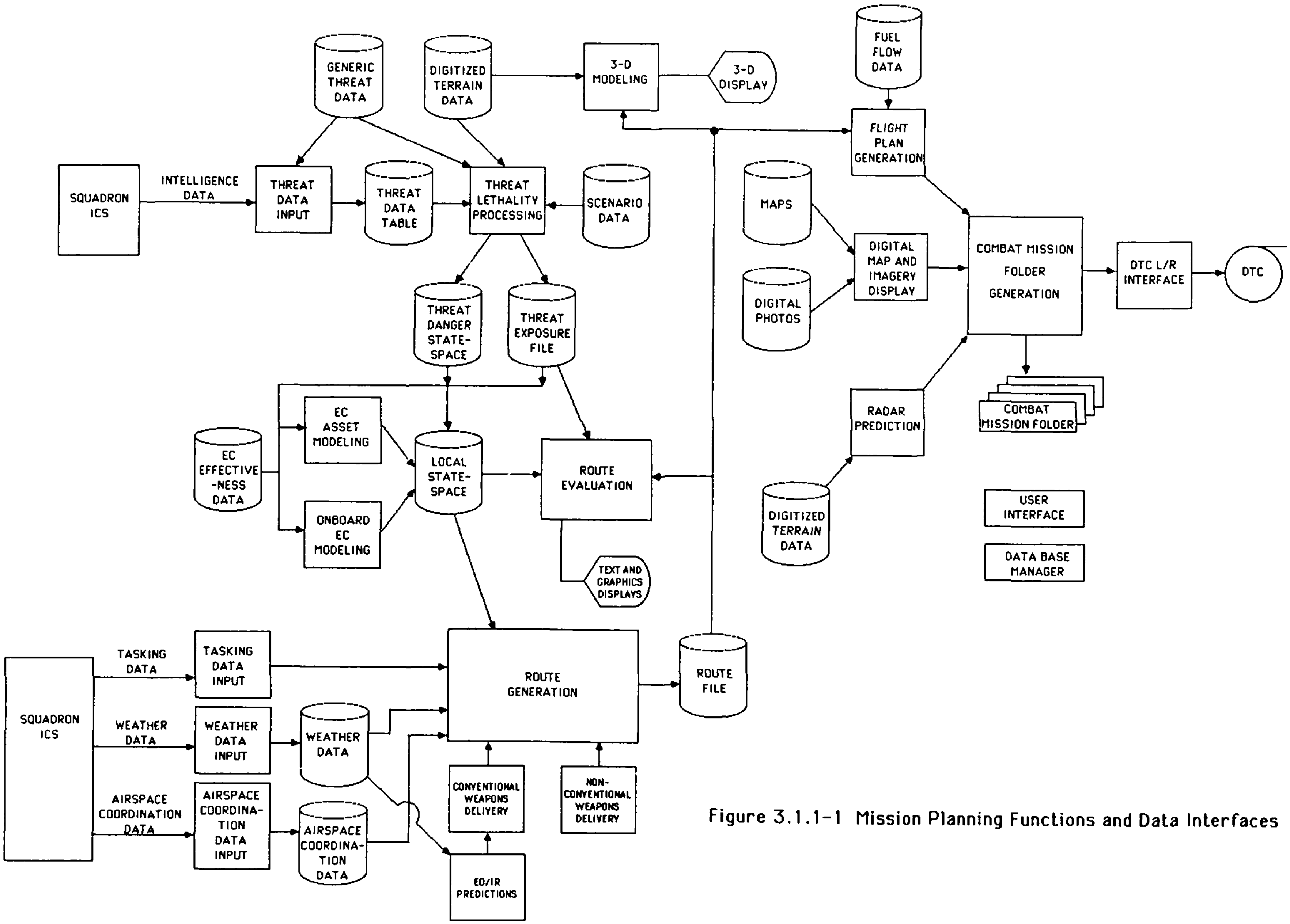

Figure 3.1.1-1 Mission Planning Functions and Data Interfaces

After the route has been planned, the system will be used to produce a CMF for the route. This will include a detailed flight plan (Form 691 in USAFE) and strip charts. The strip charts will be based on digital map images overlaid with route and flight data. Radar predictions will be produced for important navigation points along the route. Three dimensional (3-D) perspective views may also be prepared.

Finally, the route and weapons initialization data will be transferred to the Data Transfer Cartridge Loader/Reader (DTC L/R) or similar device. This peripheral device will write the route data to a Data Transfer Cartridge (DTC). The DTC will be used to initialize the flight computer onboard the aircraft.

3.1.2 MPS Follow-On Interfaces

The MPS Follow-On expects to receive data in a format it can automatically process. This data includes threat, mission tasking, weather, and airspace coordination data. A squadron ICS can be used to receive this data from the wing ICS via the WCCS or CONDUIT communications links and then convert the data into the format expected by the MPS Follow-On. The MPS Follow-On may need an interim capability to directly receive and process data from EIFEL, IINCOMNET, WCCS, CONDUIT, and CONSTANT SOURCE if the installation of the MPS Follow-On precedes that of the ICS.

The requirements for threat, mission tasking, airspace coordination, and weather data are discussed in Sections 4.1, 4.2, 4.3, and 4.4. The data which must be input into these functions can be stored in fairly short text files. For example, a file containing an EOB update of 100 threats will be about 15200 bytes long. If this file were transmitted over a line at 9600 bits per second, the transfer would take about thirteen seconds. This is typical of the data files discussed in Sections 4.1, 4.2, 4.3, and 4.4.

Consider a sample mission planning data transmission problem. The squadron ICS will transmit an EOB update containing 100 threats. The ICS wing will also transmit one mission tasking message, ten airspace coordination avoidance areas, and ten weather areas. This will require a total of 19400 bytes. At 9600 bits per second, this transmission will require sixteen seconds.

There are no requirements for how much time these data file transfers may take, beyond the 30 minute time requirement (specified by HQ USAFE/DO) for completing a mission plan. However, it appears that a link operating at 9600 bits per second will be sufficient. A link operating at 28800 bits per second will allow a 200% growth potential.

Reference 4 states that a capability to receive digital photo images over the LAN is required. This requirement was not considered in the analysis above. It is difficult to assess the impact of this requirement. A high resolution color digital image can require as much as 1.3 megabytes of storage. At 9600 bits per second, it would take eighteen minutes to transmit a single image. Transmission of a large number of these images almost certainly would require a high capacity communications link (much higher than 9600 bits per second). A low resolution black and white image can require as little as 154 kilobytes. At 9600 bits per second, such a file would require 128 seconds to transfer. It takes eight times as long to transfer a single image as it takes to transmit the mission planning data in the example above. The requirement to transfer images has a much greater impact on the transmission requirements than any of the mission planning data files.

3.2 MPS Follow-On Hardware Requirements

Table 3.2-1 summarizes the processing, main memory, mass storage, and peripheral requirements derived from the sizing analysis of the MPS Follow-On system functions presented in Sections 4 and 5. These requirements include a 200% growth margin where appropriate. This table also summarizes the security, reliability and maintainability, and environmental requirements defined for the MPS Follow-On.

Table 3.2-1 Summary of MPS Follow-On Requirements (cont'd)

COLOR PRINTER INTERFACE

STORAGE	IMAGE CAPTURE TIME	COLORS	MAGNIFICATION	CONTROL
ONE 1280 x 1024 PIXEL IMAGE MIN.	<15 SECONDS	256 MINIMUM 4096 DESIRED	X1, X2, X3	RS-232C

OPTICAL DISK

FORMAT	DISK DIAMETER	AVERAGE ACCESS TIME	CONTROL
NTSC VIDEO (LASERDISC)	12"	≤1.5 SECONDS	RS-232C

LINE PRINTER

LINE WIDTH	PRINT SPEED	DATA/CONTROL
5" MINIMUM	≥360 CHARACTERS/ SEC	RS-232C AT 9600 BAUD, OR CENTRONICS PARALLEL

SECURITY

TEMPEST CERTIFIED HARDWARE. COMPUTER CAPABLE OF PROCESSING AT TOP SECRET LEVEL.

Table 3.2-1 Summary of MPS Follow-On Requirements

GENERAL PURPOSE COMPUTER

DISK ACCESS TIME	DISK CAPACITY	DISK TRANSFER RATE	CPU SPEED	INTERFACES	MAIN MEMORY	TERMINALS
≤20 ms	200MB	≥0.5MB/sec	≥1.0 MFLOPS	8 serial at 28.8K baud 2 parallel	16 MB	1 FOR SYSTEM ADMINISTRATION

INTERNAL IMAGE PROCESSOR

IMAGE INPUT	CPU SPEED	IMAGE MANIPULATION	VIDEO/GRAPHICS ROTATION	MEMORY
NTSC VIDEO 1/30 SEC CAPTURE	8 MIPS	BLIT AT 12M PIXELS/SEC	YES	2Kx2Kx8

COLOR PRINTER

RESOLUTION	PRINT TIME	NO. OF COLORS	MAGNIFICATION	INTERFACE
300 DOTS/INCH	≤60 SECONDS FOR 8.5x11" ≤90 SECONDS FOR 11x17"	256 MINIMUM 4096 DESIRED	X1, X2, X3	CENTRONICS PARALLEL DATA AND CONTROL, OR DIRECT RGB DATA INPUT WITH RS-232C CONTROL

COLOR GRAPHICS MONITOR

RESOLUTION	NO. OF COLORS
1280 x 1024	256 MIN 4096 DESIRED

Table 3.2-1 Summary of MPS Follow-On Requirements (cont'd)

RELIABILITY AND MAINTAINABILITY

(A)	MISSION RELIABILITY (24 HRS/DAY FOR 30 DAYS)	90.0 PERCENT
(B)	UPTIME RATIO	99.9 PERCENT
(C)	MEAN TIME BETWEEN CRITICAL FAILURES	6834 HOURS
(D)	MEAN DOWNTIME	2.0 HOURS
(E)	MEAN TIME BETWEEN MAINTENANCE (PREVENTIVE)	1000 HOURS
(F)	COMBINED FAULT DIAGNOSTICS (BUILT-IN TEST, MANUAL TEST, TECHNICAL TEST)	100 PERCENT

MAINTENANCE ACCOMPLISHED BY REMOVAL AND REPLACEMENT OF LINE REPLACEABLE UNITS (LRUs). LINE REPLACEABLE UNITS TO INCLUDE THE COMPUTER; DISK DRIVES, SUCH AS THE OPTICAL DISK DRIVE; TERMINAL; VIDEO MONITOR; DTC L/R AND THE COLOR GRAPHICS AND LINE PRINTERS

ENVIRONMENT

(A) TWO-MAN PORTABLE

(B) ± 10% VOLTAGE FLUCTUATIONS

(C) 10 MINUTE UNINTERRUPTABLE POWER SUPPLY

(D) USE LOCAL POWER SOURCES

4.0 MPS FOLLOW-ON PROCESSOR AND STORAGE REQUIREMENTS

This section details the main types of operations to be performed by the MPS Follow-On processor so that an estimate of the processing and storage requirements can be made. The FLAPS approach for threat lethality processing, route generation and route evaluation are assumed for the MPS Follow-On and serves as the basis for this section's calculations. The operations are combined, using a typical scenario, to obtain the overall requirements.

FLAPS operates upon a large data base and it is important to place the time taken by each function in perspective. The following guidelines give an overview of the time consuming operations. The additional perspective of a typical scenario will be provided below.

> Disk I/O for FLAPS operations typically takes more time than the computation, even when the I/O is performed efficiently.
>
> FLAPS computations are primarily performed in floating point arithmetic.
>
> The large data bases require that the data be stored on disk and be brought into main memory for processing, since it is not practical to store all the data in main memory.
>
> Disk I/O is several orders of magnitude slower than main memory I/O, but disks are far more efficient at working with large blocks of sequentially stored data than with randomly accessed data. It is therefore vitally important that enough main memory is available to store the data "locally" (quickly accessed by the

CPU), that the computer operating system allows a large block of memory to be used by any one job, and that the software is written to take advantage of disk I/O in large blocks.

A generalized performance measure, or benchmark, is needed to determine the required performance.

A note here is appropriate to explain the assumptions used in determining the definition of a floating point operation. Different operations such as adds, multiplies, and divides may take differing relative amounts of time, depending upon the architecture of the computer being used, as well as the method of programming. That is, the use of data values stored in main memory, or in a register within the computer central processing unit (CPU), may be much faster than programming the computer to use data retrieved from a disk. The computer may require separate programming operations to (1) move the data from their storage locations into the central processing unit; (2) operate on the data; and (3) move the computed data from the processor to a new location in memory. Operations such as these, would be found in the "assembly language" for a computer. These "lower level" operations would not all be seen in a higher level language such as FLAPS uses, but are what are generally referred to in the hardware specifications for a computer. That is, if the computer is specified as being able to perform a certain number of million floating point operations per second (MFLOPS) such as additions, the data to be operated upon are assumed to already reside in the central processing unit.

When benchmarking a particular complex process, such as route generation, the programming may vary between software modules, such that the data are accessed from memory in differing ways. An "average" floating point operation (FPO) is used here to determine the CPU time taken to bring each data operand to the

CPU from main memory or from a register, operate on the data, and place the computed data back in local memory. Disk operations are removed from the FPO times. Thus, an FPO, which as defined here includes "move" operations, is about equal to 4 or 5 FLOPs as would be seen in the specification for a computer.

Typical CPU operations used in the functions that take the largest amounts of time in FLAPS, are floating point additions, multiplies, and comparisons. These three CPU operations each generally take about the same amount of time, and so the benchmark CPU operation (FPO) is defined by performing a series of additions.

Using this benchmark program on a DEC VAX 11/785 resulted in a performance of 280,000 FPOs per second, or about 1.3 MFLOPS.

4.1 Threat Data Input

This function receives threat data from the squadron ICS. The threat data must be broken into pieces (parsed) and entered into the MPS data base.

This function also includes a threat data filter. The filter maps threat systems into the known threat models. The filter also checks new threat information against the current threat data base. Typically, much of the intelligence information data will prove to be redundant. That is, a single threat system may be reported several times. A significant reduction in processing time can be achieved by not adding these redundant reports to the threat data base. The output of this process is a table of threat data (see Figure 4.1-1).

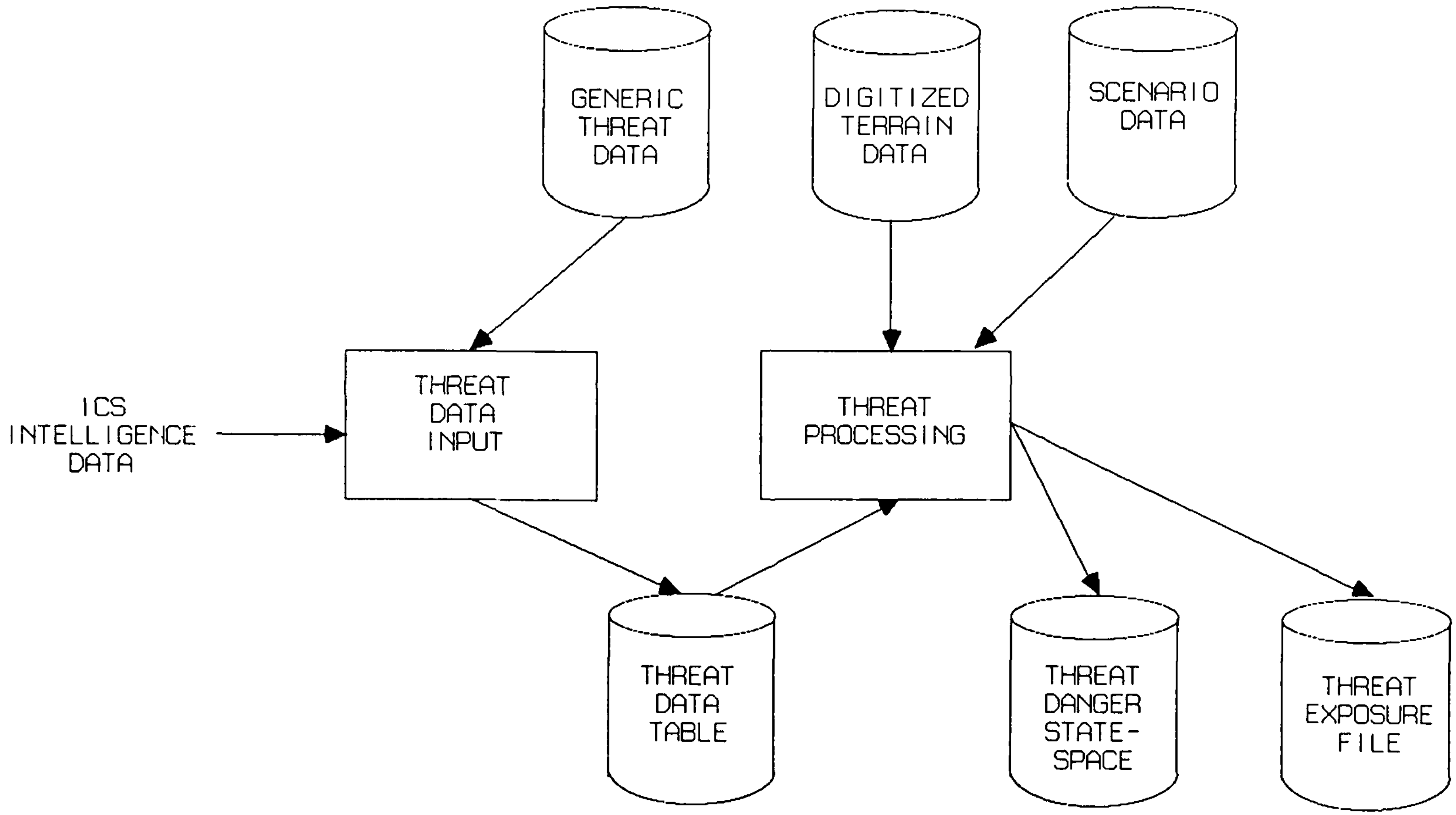

Figure 4.1-1 Threat Processing

The requirements stated below are based on the assumption that the ICS format data function will provide the threat data to the MPS Follow-On in a form that can be input directly into the MPS Follow-On.

4.1.1 Processing Requirements

The process of parsing the input data and loading it into the data base is transactional in nature and requires a minimum amount of processing. The filtering function also requires a very small amount of processing.

4.1.2 Main Memory Requirements

Main memory requirements for the threat data input function are minimal. However, several large arrays are needed by the threat data filter. Based on FLAPS, the filter requires 50,000 bytes of main memory.

4.1.3 Volume of Input Data

Each threat data record in the MPS Follow-On data base manager will require approximately 152 bytes (based on FLAPS). Multiple threat records will be periodically input to the MPS Follow-On system from the ICS. Assuming a threat update of one hundred threats, the input will be approximately 15200 bytes.

Assuming that the threat data is in ASCII (text) form, and not in binary form, and assuming that the data can be transferred to the MPS Follow-On at 9600 bits per second, the time required to input the threat update is 12.6 seconds. This is adequate.

A serial data port capable of data transfer at 9600 bits per second meets the requirements for threat data input. Special high speed data transfer capability is not required for this function.

4.2 Mission Tasking Input

This function receives tasking data for the aircrews from the squadron ICS. This data may include a target and designated mean point of impact (DMPI) specification, a weapons load, a time on target (TOT), and other data concerning coordination with other units. The tasking data must be interpreted (parsed) and entered into the MPS Follow-On data base.

The requirements stated below are based on the assumption that the ICS format data function will provide the tasking data to the MPS Follow-On in a form that can be input directly into the MPS Follow-On.

4.2.1 Processing Requirements

The process of parsing the input data and loading it into the data base is transactional and requires a minimum amount of processing.

4.2.2 Volume of Input Data

Each tasking data record in the MPS Follow-On data base manager will require approximately 200 bytes (based on FLAPS). Multiple tasking records will be periodically input to the MPS Follow-On system from the ICS. Assuming a tasking update of ten taskings, the input will be approximately 2000 bytes.

Assuming that the mission tasking data is in ASCII (text) form, and not in binary form, and assuming that the data can be transferred to the MPS Follow-On at 9600 bits per second, the time required to input the mission tasking update is 1.7 seconds. This is adequate.

A serial data port capable of data transfer at 9600 bits per second meets the requirements for mission tasking input. Special high speed data transfer capability is not required for this function.

4.3 Airspace Coordination Information Input

This function receives data concerning restricted airspace and other information critical to mission planning. This data may include Restricted Operating Zones (ROZs), Weapons Free Zones (WFZs), transit corridors, and tanker and standoff EC orbits. The source of this data is the squadron ICS. The airspace coordination data must be interpreted (parsed) and entered into the MPS Follow-On data base.

The requirements stated below are based on the assumption that the ICS format data function will provide the airspace coordination data to the MPS Follow-On in a form that can be input directly into the MPS Follow-On.

4.3.1 Processing Requirements

The process of parsing the input data and loading it into the data base is transactional in nature and requires a minimum amount of processing.

4.3.2 Volume of Input Data

Each tasking data record in the MPS Follow-On data base manager will require approximately 200 bytes (based on FLAPS). Multiple updates will be periodically input to the MPS Follow-On system from the ICS. Assuming an update of fifty records, the input will be approximately 10,000 bytes.

Assuming that the airspace coordination data is in ASCII (text) form, and not in binary form, and assuming that the data can be transferred to the MPS Follow-On at 9600 bits per second, the time required to input the airspace coordination update is 8.3 seconds. This is adequate.

A serial data port capable of data transfer at 9600 bits per second meets the requirements for airspace coordination information input. Special high speed data transfer capability is not required for this function.

4.4 Weather Data Input

This function receives data concerning weather which may impact the aircrew mission. This includes terminal area weather, enroute weather, and target area which may impact weapons delivery. The source of this data is the squadron ICS. The weather data must be interpreted (parsed) and entered into the MPS data base.

There are two possible formats for weather data. In the current version of FLAPS (4.1), weather is represented as polygons or lines which may be displayed. Constraints may be placed on these polygons to restrict specific types of weapons and aircraft from being used inside them. Each weather polygon requires approximately 200 bytes of storage.

The second format is based upon gridded weather data consistent with future versions of the Air Weather Service's Tactical Decision Aid (TDA) program. Gridded weather would contain forecast weather information at multiple pressure altitudes. In the future, gridded weather data may be available to force level and unit level planning systems via satellite. Gridded weather data is not currently available. The gridded weather data version of TDA is not operational at this time either. This option is included here only to suggest a future growth option.

4.4.1 Processing Requirements

The process of parsing the input data and loading it into the data base is transactional in nature and requires a minimum amount of processing.

Assuming that weather data is input in polygon form, each weather data record in the MPS Follow-On data base manager will require approximately 200 bytes (based on FLAPS). Multiple weather data records will be periodically input to the MPS Follow-On system from the ICS. Assuming a weather data update of one hundred polygons, the input will be approximately 20,000 bytes. Assuming that the weather data is in ASCII (text) form, and not in binary form, and assuming that the data can be transferred to the MPS Follow-On at 9600 bits per second, the time required to input the threat update is 16.7 seconds. This is adequate.

A serial data port capable of data transfer at 9600 bits per second meets the requirements for weather data input. Special high speed data transfer capability is not required for this function.

A gridded weather data file will be much larger than the polygon data discussed so far. A gridded weather file will be in the range of one or more megabytes. At 9600 bits per second it takes about seven minutes to transfer one megabyte. This is probably not acceptable. If gridded weather data is used in the future, then a high speed data input capability will be required.

4.5 Relative Threat Lethality Processing

This function performs terrain masking and relative threat lethality computations on the threat data base. The output of this process is a relative threat lethality array or "statespace" and a terrain masked exposure array for each threat (as shown in Figure 4.1-1). Relative threat lethality will be computed at several different altitudes. The statespace is used for minimum threat route generation, route evaluation, EC asset modeling, and onboard EC modeling.

4.5.1 Requirements

The requirements for relative threat lethality processing are summarized below. These requirements are taken from references (3) and (4).

(1) Capability to process 100 threats per hour.

(2) The system must compute threat lethality contours and line of sight coverage for enemy threats as a function of ingress and egress altitude, type of aircraft, and aircraft speed.

(3) The system must be capable of 3-D route optimization. It must be able to recommend routes which optimize aircraft survivability independent of artificial altitude boundaries.

The processes required to meet these requirements are described in the next two subsections.

4.5.2 Terrain Masking

Terrain masking is a straightforward but time consuming process. It is processor intensive and, as implemented in FLAPS, it is also I/O intensive. Computer I/O speed (to a hard disk) must be considered along with processor speed to determine if timing requirements are satisfied.

The following is a brief and non-technical summary of the terrain masking algorithm. The threat location, elevation above the terrain, and maximum radius (R) for a threat are known from the threat and threat model data bases. The maximum radius, R, is the maximum radius to be considered for all altitudes. Terrain data is read in for a square window covering a circle or radius R centered at the threat location.

A recursive algorithm computes masking effects along rays beginning at the threat location and running out to R. Masking effects are computed as Minimum Observable Altitude (MOA) above the terrain at given points along each ray. MOA is the lowest altitude above the terrain at which the threat has a line of sight to the point. Note that MOA is not tied to any arbitrary altitude boundaries. MOA data (oriented radially) is stored temporarily in the MASK array (which acts as a buffer). The number of points on each ray and the number of rays are determined by program parameters. There will always be at least one MOA point for each statespace cell at the outer edge of the threat circle. There will, in general, be many MOA points per cell near the center of the threat. After all of the radially oriented MOA points have been computed, the data is transformed into a rectangular grid. Typically, four MOA points per statespace cell are computed and stored using bilinear interpolation. That is, bilinear interpolation is used to transform the radially oriented MOA data into rectangularly oriented data. This data is stored in the Terrain Observability (TOBS) array. A block of MOA data is stored for each threat.

Radar threats are masked, using the terrain, at various altitudes. The time taken to mask a threat is roughly proportional to the square of the threat radius, as shown in Table 4.5.2-1. The masking operation is predominantly floating point arithmetic. The numbers in Table 4.5.2-1 were gathered using a processor with a floating point performance of 1.7 MFLOPS.

MOA data can be readily retrieved. Because the MOA data is not tied to any arbitrary boundaries, it is possible to display exposure contours for individual threats at any AGL altitude. This is done by reading in the MOA data for a threat, computing contour lines at the points where MOA equals the AGL clearance altitude, and plotting the result. It is also possible to evaluate a route against threat exposure. This is done by reading in the MOA data for each threat (one at a time) and comparing leg clearance altitude against MOA for the portions of the route that go through clearance altitude against MOA for the portions of the route that go through each threat. Any leg clearance altitude may be used. Different legs may have different clearance altitudes. The result of this process is total exposure time to each threat.

4.5.3 Threat Lethality Processing

Threat lethality processing is the process of combining the relative threat lethality templates (or footprints) with the MOA data and storing the results in the statespace. Threat templates are stored on disk for each type of SAM or AAA threat of interest. These templates may be dependent on altitude and/or aircraft type. However, it is unusual to use different templates for different types of tactical aircraft.

Table 4.5.2-1 Terrain Masking

RADIUS	TOTAL CPU TIME (SEC)	I/O (WORDS)	TIME SPENT BY CPU IN I/O* (SECONDS)	TIME SPENT BY CPU IN MATH (SECONDS)	# FPOs PERFORMED BY CPU (MFPO)**	WALLCLOCK TIME (SECONDS)
10	1.8	28,269	.74	1.056	.295	2.477
20	6.17	98,706	2.60	3.57	1.0	8.472
40	21.15	298,589	7.86	13.29	3.7	33.531
60	46.75	616,635	16.23	30.52	8.5	74.086
100	127.85	1,593,795	41.94	85.91	24.05	221.164

* 38,000 I/O WORDS/CPU-SEC
** 280,000 FPO/SEC

Existing programs (FLAPS and the current MSS) that use this relative threat lethality approach, compute, and store lethality values for each statespace cell at predetermined altitude levels. There is a reason for this. The statespace is a summary of the effects of all threats within the scenario. Each statespace cell contains the relative threat lethality to the aircraft if it flies through that cell (at a certain altitude). This is a single number, regardless of how many threats have a line of sight to the cell.

Once the statespace has been processed, then most planning functions can proceed quickly, independent of the number of threats. Operations like displaying threat lethality contours, route optimization, and route evaluation (to determine leg and route lethality), depend on the size of the statespace and not on the number of threats. For example, suppose a statespace is one hundred cells by one hundred cells (at a 4.5 nm by 4.5 nm cell size). The amount of time it takes to generate or evaluate a route is the same regardless of whether there are ten threats or ten thousand in the scenario. Of course, the amount of time it takes to compute the statespace depends directly on the number of threats, but most route planning operations do not.

The process of building a statespace is as follows. The MOA data and associated threat template for each threat are read from disk. For each statespace cell within the threat's maximum radius and for each AGL altitude level of interest, the danger is computed. If the AGL altitude is less than the MOA for that cell, there is no danger for that cell. If the AGL altitude is greater than or equal to the MOA, then the danger is computed from the threat template and AGL altitude. This danger is added to the current value for the cell. This process is repeated for each cell, each AGL altitude level, and each threat. Note that the statespace does not maintain any record of which threats contributed to the danger in a cell.

The process will be referred to as a "Statespace Add." This is an important operation. This process is repeated in reverse when a threat is removed from the scenario or is affected by EC. The approximate number of mathematical and I/O operations for a statespace add are shown in Table 4.5.3-1. Note that a statespace add is much faster than a terrain masking operation.

Route evaluation will be discussed below, but a few remarks are relevant here. A route evaluation to determine leg by leg and total route lethality can be made using a precomputed statespace. Sometimes this is not sufficient. In order to determine which threats have contributed to the danger along a route, a detailed route evaluation is required. This procedure is very similar to a Statespace Add. The process requires that the exposure to each threat be computed, as described above in the terrain masking section. After the MOA data has been used to determine that a given threat exposes a route, the threat model is read in from disk. The exposed part of the route is traced through the threat model and the danger is accumulated. This danger can then be reported along with the exposure time. Note that the terrain masking did not need to be done again, although the relative lethality computations did have to be repeated. It is a good approximation to assume that a detailed route evaluation takes about as long to do as a Statespace Add per threat. In other words, it is much more time consuming than a leg-by-leg lethality computation. However, a detailed route evaluation does not require a precomputed statespace and may be done at arbitrary leg AGL altitudes. That is, the leg altitudes do not have to match the pre-selected statespace altitudes.

Table 4.5.3-1 Statespace Add (4 Altitudes)

RADIUS	TOTAL CPU TIME (SEC)	I/O (WORDS)	TIME SPENT BY CPU IN I/O (SECONDS)	TIME SPENT BY CPU IN MATH (SECONDS)	# FPOs PERFORMED BY CPU (MFPO)	WALLCLOCK TIME (SECONDS)
10	.21	4,109	.11	.10	0.28	.477
20	.55	9,553	.25	.30	0.84	.922
40	1.76	28,669	.75	1.01	.281	2.977
60	3.51	56,821	1.50	2.01	.564	5.672
100	8.8	148,909	3.92	4.88	1.37	13.430

38,000 I/O WORDS/SEC I/O

The requirement to display danger contours at arbitrary altitudes within two minutes requires that a statespace add be done for each threat at the specified altitude. This is a significant amount of processing. If there are 100 threats of 20 nm each, then this function will require a CPU with a 15 MFLOPS performance, and a commensurate I/O speed. This requirement is much harder to satisfy than the 100 threats per hour requirement, which only requires 0.5 million floating point operations per second (MFLOPS).

The processing required to support automatic full 3-D route generation is not estimated in this report. Full 3-D optimization is not practical at this time. However, full 3-D route generation in a manual mode is possible and discrete 3-D route generation in an automatic mode is possible.

4.6 Route Generation

This function will generate routes for the missions tasked by the higher command levels, consistent with the threat, weather, and airspace coordination data, as shown in Figure 4.6-1. Missions may be generated using a route optimization procedure, by manual input of turn points, or a combination of these two.

4.6.1 Requirements

The requirements for route generation are summarized below. These requirements are taken from References (3) and (4).

(1) The route generation function must generate ingress and egress routes which optimize the survivability of the aircrew. These routes must meet fuel constraints.

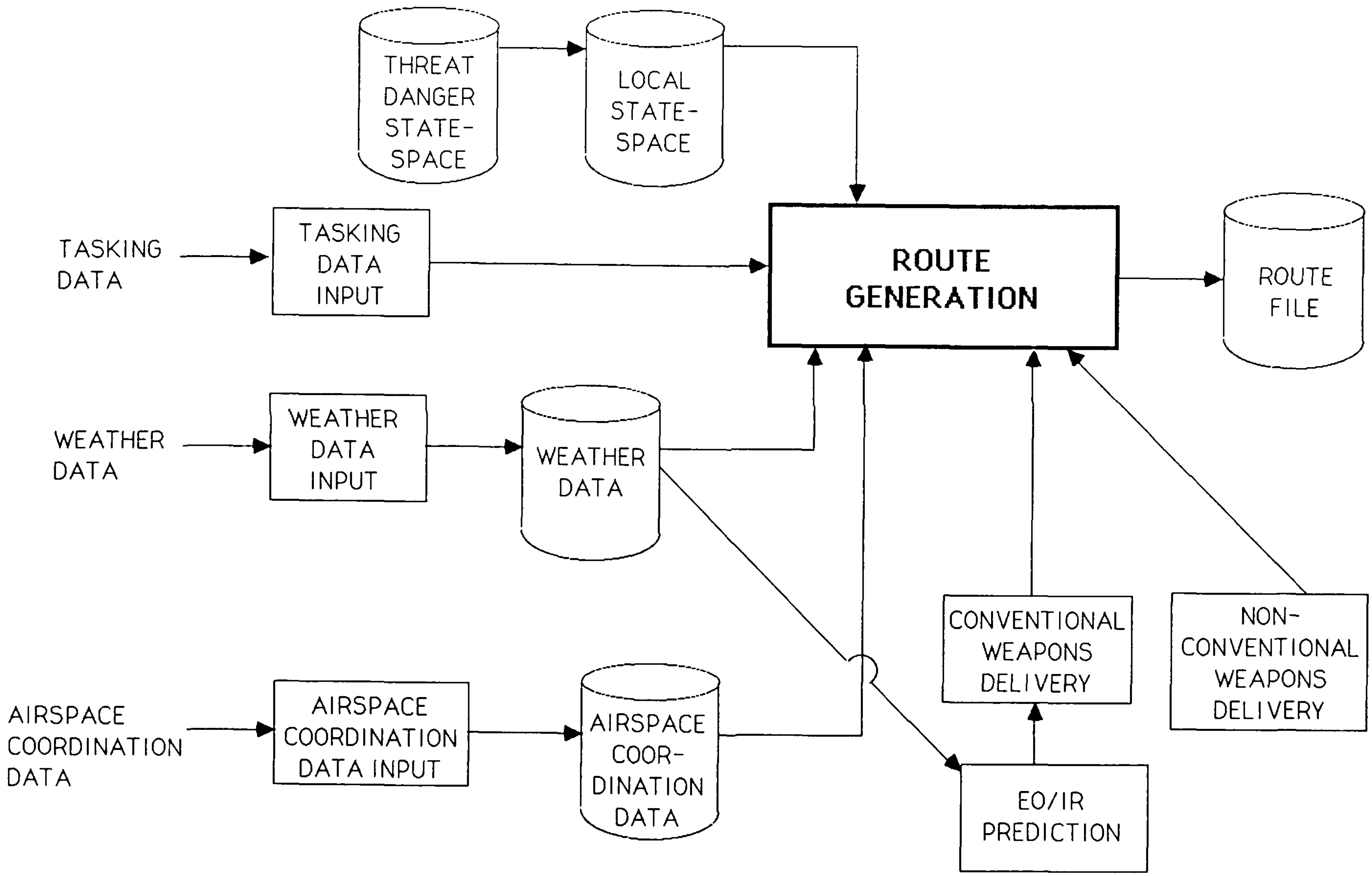

Figure 4.6-1 Route Generation

(2) The route generation algorithm must generate three-dimensional routes not limited to artificial altitude boundaries.

(3) The route generation function must allow user input of routes and modifications to the routes generated by the optimization algorithm. This includes the input of leg altitudes and turn points.

(4) Given a user-specified route or leg, the route generation function must recommend a penetration altitude which maximizes the probability of survival on that route or leg. Altitude selection may also be made based on a user designated maximum lethality value for the route or leg.

The processing required to meet the requirements stated above can best be described in three parts. The first part is route optimization performed on a statespace using dynamic programming. The second part is altitude selection along a route for which the turn points have already been specified. This does not require dynamic programming. The third is user input and modification of existing routes.

There is some overlap between route generation and the relative threat lethality modeling and route evaluation functions. The reader may have to refer to the descriptions of these two functions as this section is read.

There is often confusion between route optimization and route evaluation. The following definitions should eliminate some of this confusion. For this report, a route is a sequence of waypoints including latitude, longitude and altitude. Route optimization refers to the process of automatically generating a route between two points. Route evaluation refers to the process of taking an existing route and determining one or more measures of merit associated with it. These may include distance, fuel consumption, and relative threat lethality. Altitude selection along a route is the process of assigning an AGL altitude to a leg, where the coordinates of the end points of the leg are known.

Section 4.7 discusses, at some length, what one can reasonably expect a route optimization algorithm to do. Section 4.7 will conclude that a full three-dimensional route optimization algorithm that is not dependent on pre-assigned, or arbitrary, altitudes is not practical. However, it is very feasible to generate a route which selects the best penetration altitude for a leg from a fixed set of pre-assigned altitudes (best penetration altitude for a cell is computed prior to route generation). It is even feasible to run a three-dimensional dynamic programming algorithm based on pre-assigned altitudes.

4.6.2 Route Optimization Using Dynamic Programming

Dynamic programming is a mathematical algorithm which can be used to find the optimal trajectory for a process which evolves in time. In particular, it can be used to find the optimal path between two points in a grid, where the cost of going from one grid cell to the next is known. For this application, dynamic programming is similar to other procedures (such as shortest path algorithms). The processing requirements for route generation stated in this document will be based on dynamic programming. Other algorithms will require similar amounts of processing. It is also possible that other approaches will require significantly more processing, depending on how the approach is implemented. This possibility will not be considered in this report.

Dynamic programming can be used successfully if the routing problem can be confined to a two or three-dimensional grid. The cost in going from one grid cell to an adjacent cell must be known and must not depend on any previous (or future) parts of the route. The number of operations required to generate a route is proportional to the number of grid cells. If a grid is M cells by N cells, then the number of operations required to find an optimum path is proportional to M x N, and not M to the Nth!

For mission planning, the grid is the statespace. The cost of going from one cell to the next is the relative threat lethality of flying through a cell. This cost is calculated based on the number of, and types of threats that expose the cell. The dynamic programming algorithm (DPA) will optimize route survivability to the extent that the relative threat lethality statespace reflects survivability. The optimized route will have the lowest possible total threat lethality, based on the sum of the lethalities of the cells that were flown through.

It has been suggested that probability of survival cannot be calculated in the manner described above. For the purposes of this report, the minimum relative threat lethality route will be defined as the route which maximizes aircrew survivability.

The statespace grid must exist prior to running the route optimization algorithm. In particular, the threat lethalities for each cell must be known prior to running the algorithm. Because threat lethality is altitude dependent, an altitude must be assigned (implicitly or explicitly) to each cell. The most common method is to build the statespace at an arbitrarily assigned penetration altitude. In FLAPS and the current MSS, a three-dimensional statespace is built at several preassigned penetration altitudes. Prior to computing the optimum route, the planner specifies the altitude to be used. The two-dimensional statespace associated with this single altitude is used for route generation. This approach is useful for generating a "rough" or "first cut" route. The user may make changes to this route (including changes to leg altitudes) and reevaluate these changes using the route evaluation function. The requirements for route generation specify that route generation should be three-dimensional and should not be tied to preassigned, or arbitrary altitudes. There are two practical alternatives to meet this requirement.

The first alternative is to determine the best clearance altitude for each statespace cell prior to running the DPA. This approach will be referred to as a "decoupled 3-D" algorithm. The 3-D statespace is processed into a two-dimensional statespace, where each cell contains a lethality value and a clearance level. This clearance altitude is the altitude that results in the minimum threat danger, if the vehicle flies through this cell. The lethality value is that which corresponds to this best altitude level.

For example, suppose that the altitude levels are 200, 500, and 1000 feet. For a given cell, the threat danger is 0.1, 0.15 and 0.45 at 200, 500, and 1000 feet, respectively. The best altitude and danger for this cell would be 200 feet and 0.1 relative lethality. At another cell, the dangers are 0.25, 0.25, and 0.7 at 200, 500, and 1000 feet. The best altitude and lethality are 500 feet and 0.25. Flying at 200 feet does not reduce the survivability, so it is better to fly at 500 feet. This assumes that flying higher is better than flying lower, if the lethality is constant. The standard two-dimensional dynamic programming algorithm is run to determine the lateral portion of the route. The leg altitudes are retrieved by reviewing the cells that are flown through. The leg altitude is the lowest of those associated with the cells flown through on the leg. The problem with this approach is that the route will tend to fly low all of the time. There may also be many short legs, each at a different altitude.

The second approach is to run a three-dimensional DPA on the three-dimensional statespace. Recall that the 3-D statespace is constructed at preassigned altitude levels. This approach will require more processing time than the decoupled 3-D algorithm. It will also produce exactly the same result for the problem formulated as above. A 3-D algorithm will produce different (and better) results than the decoupled 3-D algorithm only if transition costs between altitude levels are assigned. This algorithm will also tend to fly low most of the time.

A full 3-D algorithm that is independent of preassigned altitude levels cannot be based on a statespace or DPA approach, because it is impossible to establish a statespace grid. The processing required for such an algorithm will not be estimated in this report.

4.6.3 Altitude Selection Along a Lateral Route

The altitude selection function will assign AGL clearance altitudes to each leg along a route. The route turn points, or the lateral part of the route, have already been input. The lateral route may be input manually by the user, or it may be the output of the route optimization algorithm.

The process of determining the best altitude to fly on a leg is very similar to route evaluation. For each threat which contains all or part of the route or leg within its maximum radius, the Minimum Observable Altitude (MOA) and associated threat template are read from disk. The route or leg is traced through the MOA data. The route or leg clearance altitude is set just below the lowest MOA for a threat along the leg, if this clearance is lower than the current clearance value. This masking process is repeated for the remaining threats. The result is a leg clearance setting that is just below the lowest MOA along the leg. A threat that is not effective at this MOA is ignored. The leg clearance may also be set so as to not exceed a specified leg lethality threshold.

This approach can result in very low clearance settings. The altitude selection may include a clearance of zero feet AGL. To avoid this, a minimum clearance must be used (for example, 100 or 200 feet). The computations to perform this altitude selection function are essentially equivalent to those required for detailed route threat evaluation.

4.6.4 User Input of Waypoints

The user must be able to create routes entirely by inputing waypoints and by modifying routes generated by the route optimization function. This route creation process is accomplished by inserting and/or deleting waypoints from a route via text or graphics inputs. The process of modifying or creating a route manually requires minimum computer resources. However, evaluating the route after it has been created does require significant processing. The route evaluation process is covered in Section 4.7.

4.6.5 Processing Requirements

There are three major steps in executing the dynamic programming algorithm. First, the statespace must be read in from disk. This is mainly an I/O operation. Second, the DPA must be run on the statespace. This produces an optimal transition at each statespace cell. Third, the route must be "retrieved" or constructed from the DPA result. This involves tracing through the statespace and retrieving the overall route from the optimal decisions at each cell. This last step also converts the turn decisions from grid cell coordinates to geographic coordinates.

The following analysis and example is based on a FLAPS scenario.

The first step only requires that a statespace be read into main memory. This requires that a block of data equal to the size of the statespace (or a windowed area of the statespace) be read in. A timing study was made using FLAPS based on a 4.5 nm by 4.5 nm statespace containing 68 by 69 cells. FLAPS uses eight threat lethality or danger values per cell, one for each of the eight cardinal directions. The two dimensional statespace (for 4.5 nm X 4.5 nm) is therefore 68 x 69 x 8 words, or 37,536 words (150,144 bytes).

The second step is to run the DPA. Technically, FLAPS uses a multi-pass implicit stage dynamic programming algorithm. The dynamic programming algorithm usually takes about 3 passes across the statespace to find an optimal solution. The current algorithm evaluates the three transitions directed towards the target (or target point) at each cell, at each pass. Each evaluation requires about three floating point operations (FPOs). For a 68 x 69 statespace and three passes (this is typical), the number of floating point operations is about 125,000. That is, the number of cells, times the number of transitions, times the number of operations, times the number of passes.

The third and final step is to retrieve the route. The number of operations here is difficult to estimate in terms of cells, but the time taken is significantly less than that needed to perform the DPA.

Route retrieval operations are performed by several software modules and are more difficult to benchmark than the DPA that is performed by a single "kernel".

FLAPS was run on the 68 cell x 69 cell statespace. It produces an ingress and egress route, each about 150 nm long. The generation of two routes required about two seconds on a VAX-11/785. The time required is as follows (see Table 4.6.5-1). Reading the statespace requires about one second (37,536 words at about 38,000 words per second). The I/O time was measured separately from the computation time. The DPA and retrieval required about one half second per route, for a total of about one second.

The DPA required about 125,000 floating point operations per route at 280,000 FPOS, or about 0.45 seconds. The remainder, roughly 0.05 seconds was taken by route retrieval. In this example, about 15,000 floating point operations. So a rough estimate of route retrieval operations is three times the number of statespace cells (15,000 = 3 x 68 x 69). Route generation for this typical route takes about 140,000 floating point operations (125,000 + 15,000 or 140,000 floating point operations per route).

4.7 Route Evaluation and Threat Analysis

This function allows the planner to evaluate a route against the threat statespace to determine the threat exposure level. The planner may also determine which threats are encountered along the route. The calculations are based upon the threat lethality statespace and the threat exposure file.

Table 4.6.5-1 Route Generation Requirements

	STATESPACE READ	RUN DPA (2 ROUTES)	ROUTE RETRIEVAL (2 ROUTES)
WORDS	37,536		
FPOs	-	125,000	15,000
VAX-11/785 CPU TIME (SEC)	1	0.9	0.10
VAX-11/785 WALLCLOCK TIME (SEC)	1	1.4	0.15

4.7.1 Requirements

The requirements for route evaluation and threat analysis are as follows:

(1) Determine the relative threat lethality for a route. The route may include arbitrary turnpoints and altitudes.

(2) Determine the amount of danger each threat contributes to the total relative threat lethality for the route.

Route evaluation relies heavily on the data produced by the relative threat lethality processing function. The multi-altitude statespace and threat observability file are used to evaluate the routes for threat danger. Figure 4.7.1-1 shows the interaction between the files.

There are two major components within the route evaluation function. One is route evaluation on the statespace, and the other is detailed route evaluation. Route evaluation involves tracing a route through the statespace and summing up the relative threat lethalities for each cell that the route passes through. The result is the total threat lethality for the route. If done on a leg by leg basis, the result is a leg by leg breakdown of the threat lethality. This process can only be used when AGL altitude for the route or leg is consistent with one of the altitudes the statespace was built at. This route evaluation process can be executed very quickly. Execution time depends on the length of the route and the quantization level of the statespace (the cell size). It does not depend on the number of threats.

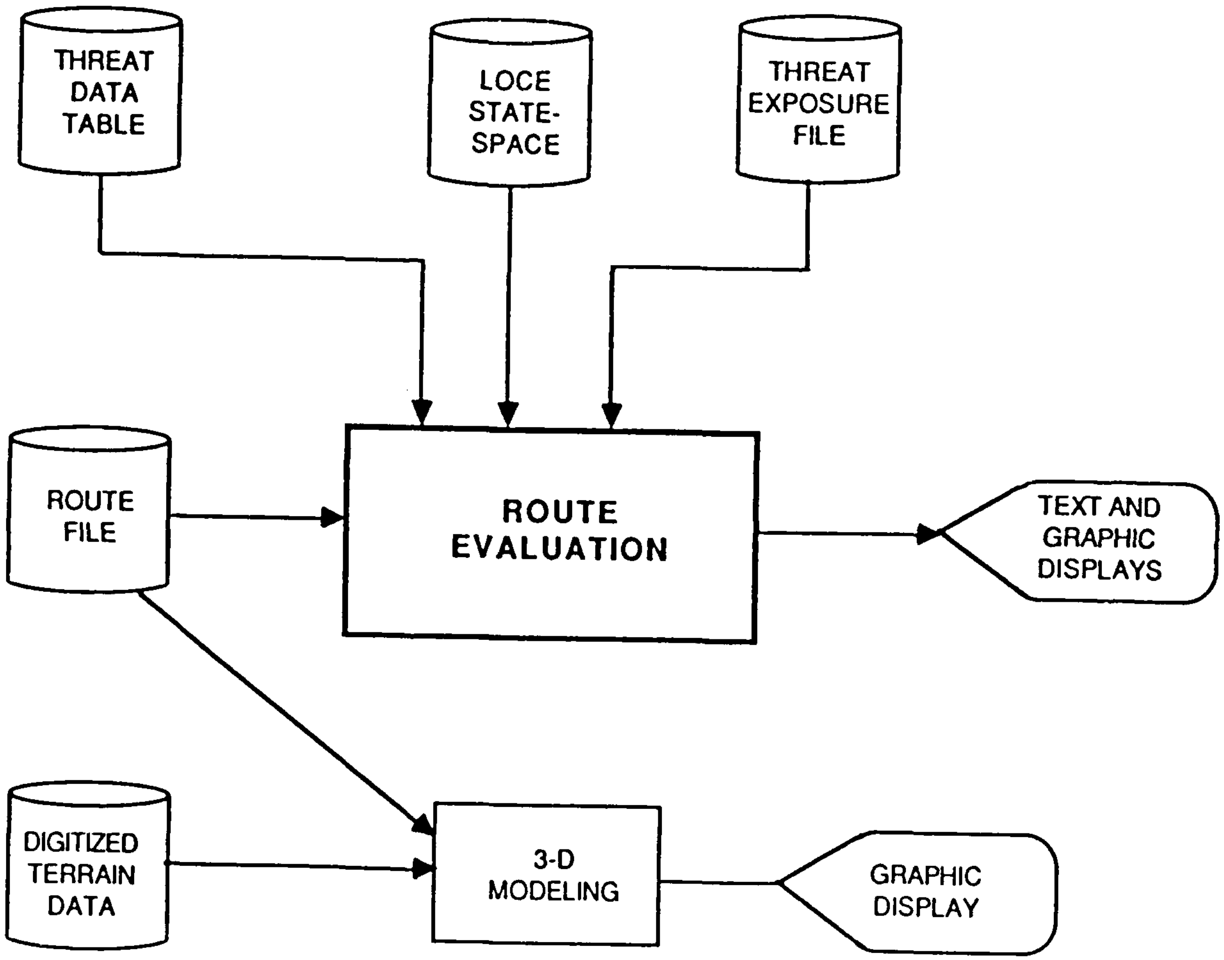

Figure 4.7.1-1 Route Evaluation

The second major component is detailed route evaluation. This process requires that the route be passed across the threat observability data and threat template for each threat. Threats that the route does not encounter are ignored. Threats which the route flies through, but which contribute no danger because of the contents of the threat template, are also ignored. For a threat that the route is exposed to and which will contribute danger, the relative threat lethality contribution is computed based upon the threat template, the MOA data for the threat, and the route. This process can be performed for any route, and for any route or leg altitudes. It is independent of the statespace and the altitudes the statespace was built at. Detailed route evaluation is slow to run. Details are described below. Run time is dependent on the number of threats encountered by the route.

4.7.2 Processing Requirements

To evaluate a route for relative threat lethality on a statespace requires an I/O step and a mathematical processing step. The I/O step involves reading in the statespace and the route. The mathematical processing involves tracing the route through the statespace and summing up the dangers.

If the statespace is M by N cells, then the number of words that must be read is M x N x k, where k is the number of danger values per cell. For FLAPS k is eight. Reading the route from disk requires much less time because this file is much smaller (several orders of magnitude smaller than the statespace). It is reasonable to ignore the I/O for the route in estimating the volume of I/O.

The processing required depends on the number of cells that the route passes through. The number of cells crossed is dependent on the length of the route. Route evaluation requires approximately twenty floating point operations per cell (based on an analysis of FLAPS software).

Consider the routing example used in the route generation subsection. The statespace is 68 x 69 cells. A round trip route was generated that was approximately 300 nm long. The cell size was 4.5 nm x 4.5 nm. Reading the statespace requires reading 68 x 69 x 8 words, or 37,536 words. A 300 nm route will cross about 67 cells (300 nm/4.5nm/cell). This relative threat lethality evaluation will require about 67 x 20 (1,340) floating point operations (see Table 4.7.2-1). This is small compared to the requirements for route generation.

Detailed route evaluation requires much more I/O and mathematical processing. Evaluating a route through the individual threat is very similar to a "statespace add" discussed in Section 4.5.3. The danger must be computed for each threat and for each cell that the route passes through. The processing is dependent on the number of threats the route passed through, and the radius of the threats. To estimate the I/O and processing required, refer to Table 4.5.3-1. The timing required will be approximately equal to the number of threats encountered by the route multiplied by the amount of time required to do a "statespace add" for a threat. This can be a time consuming process, however, note that the threats do not have to be masked all over again. The masking data is used directly from the threat observability data file.

Table 4.7.2-1 Route Evaluation Requirements

	STATESPACE READ	ROUTE EVALUATION PROCESSING (2 ROUTES)
WORDS	37,536	-
FPOs	-	1,340
CPU TIME (SEC)	1	0.01
WALL CLOCK TIME (SEC)	1	0.01

4.8 Flight Plan Generation

This function operates on the routes created during route generation. Detailed fuel leg timing computations are performed on the route. This includes the effects of enroute winds. This data is sufficient to produce a Form 691. This flight plan data is appended to the route data file and stored by the data base management system.

Detailed fuel computations will most likely be computed using some form of the TAF Flight Planning Software. This software computes fuel flow rates based on polynomials. These polynomials were constructed based on fuel consumption curves from the aircraft Dash-1 documents. This software is written in BASIC.

It is very difficult to estimate the processing required to produce the flight plan. The polynomial coefficients are stored on disk and must be read in prior to evaluating each leg. The I/O time required is insignificant. The polynomials are then processed to compute fuel flow and fuel consumption for the leg. The time required to process a leg is slightly less than one second on a VAX 11/785. For a 20 leg route, an estimate of the wall clock time would be 20 seconds. Estimates for other computers should be made by comparing the speed of that computer to the VAX. This should include I/O and floating point computation speed.

4.9 Combat Mission Folder (CMF) Generation

This function produces the CMF. This includes a Form 691 and color strip maps of each leg. The color strip charts will include a display of the leg on a standard navigation chart. The map images will be stored on optical disks (see Figure 4.9-1). The leg will be annotated using standard Course Arrow

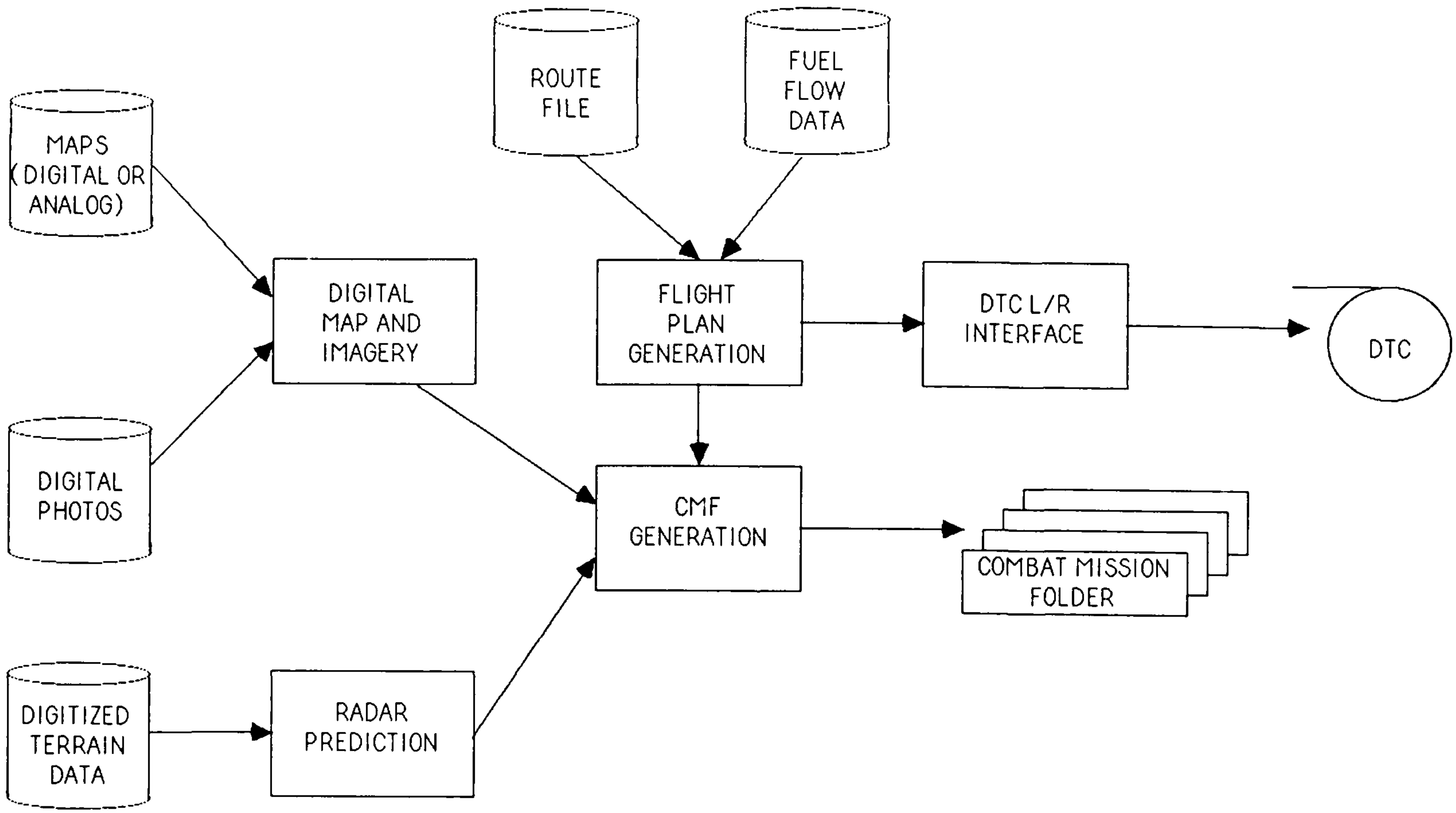

Figure 4.9-1 Combat Mission Folder Generation

Box (CAB) and Navigation Information Box (NIB) notation. This strip map will be produced entirely by the MPS Follow-On. A printer capable of producing high quality color strip maps is required to perform this function.

The data for the CMF is computed during the flight planning process. The CMF generation function is to produce hardcopy output of the annotated strip maps for each leg. The processor must first determine the location, on the map disk, of the images to be used as a background for the flight path. The field of view (FOV) of each individual map image stored on the disk is not large enough to contain a typical flight leg. A "mosaic" of several map images must be made by abutting several images end to end with the proper overlap so that the map features are aligned between map images. If video images (from Laserdisc based maps) are used, the image processor must be capable of digitizing an NTSC format video image in 1/30th of a second, since this is the time taken for display of an NTSC video image. The image processor should contain time base correction circuitry so that the images are digitized consistently. The "location" processing requirements are insignificant in terms of the host processor, but efficient operation upon the large arrays involved in image processing requires a special purpose processor that can perform block image transfer (BLIT) operations at high speed.

A typical high resolution displayed image may contain 1280 x 1024 pixels and will be obtained by selecting a portion of a larger image that is stored in the image processor's memory. In this manner, roaming and scaling operations can be done on the stored image. This stored image may typically contain 2k x 2k pixels. If each pixel can be one of 256 possible colors, the image memory must contain 4MB of storage. The images must be operated upon rapidly. An image transfer may involve the movement of a large percentage of the 4MB of storage, so to do the transfer in less than 1 second a BLIT speed of 4M pixels/second is required. For the 200% growth requirement, this becomes 12M pixels per second.

The processor must be capable of rotating the background map image at an arbitrary angle to align the image with a leg of the flight path.

Hardcopy of a displayed image can either be made by transferring the image from the image processor's memory to the host processor's memory, formatting the data and transferring the data along a digital link to the printer, or by redigitizing the RGB signals that drive the display monitor and sending these digitized signals to the printer. Each method has its tradeoffs.

The first method requires that the host processor format the data for transmission to the printer. There are about 1.3 million 1 byte pixels in the image, so if several instructions are needed to format each byte, this part of the transfer could take several seconds.

The printer typically has a Centronics parallel interface operating at an effective rate of about 50 Kbytes/second, thus it would take an additional 25 seconds to transfer the image. The printers do not typically have image storage, so multiple copies involve this 30 or 40 second transfer time.

A printer interface that captures the image and stores it for making multiple copies can be used. Picture quality may be slightly degraded from the original image. This type of interface has the advantage of being easily connected to various different image sources for printing.

The current state of the art in color graphics printers allows an 8.5 x 11" print to be made in about 60 seconds, or an 11 x 17" print in about 80 seconds. Assuming that two CMF pages of a typical 10 page CMF can be printed simultaneously on one 8.5 x 11" sheet, each CMF would take about 5 minutes to print. This is marginally acceptable.

4.10 Radar Predictions

This function will produce a prediction of the aircraft's ground mapping radar screen at critical points along the flight path. This may include critical navigation points, the initial point (IP), and the target. The radar prediction may be printed for insertion into the CMF.

This process is based upon the terrain masking algorithm and the graphics display. Producing the data for the radar prediction is very similar to producing a terrain mask for a threat. The only additional computation is for the intensity of the reflection of the radar when it strikes the ground. The reflection calculation is based on the slope of the terrain when the radar "ray" hits the ground. This reflection calculation is done inside the innermost loop of the line of sight calculation. The processing requires approximately double the number of floating point operations required to do the terrain masking. However, for terrain masking, a full three hundred sixty degree field of view is computed. For radar prediction, only a fraction, or cone is computed. This cone is typically 70 to 110 degrees.

To estimate the number of operations required to do radar prediction, refer to Table 4.5.2-1, which shows the processing required to do terrain masking. To do a radar prediction over a 90 degree field of view at R nm range requires about half the number of operations to mask a threat of radius R. The time required to do a radar prediction depends on the desired degree of resolution as well as the scope range (radius).

The actual radar prediction is produced on the display device by drawing filled polygons on the screen. The intensity of each polygon is proportional to the estimated intensity of the radar return.

In practice, much of the time required for radar prediction is spent in drawing in filled polygons. Drawing filled polygons is time consuming because of the large volume of data that must be passed between the host computer and the graphics display device, and because drawing a filled polygon is taxing for the graphics processor.

The FLAPS program can do radar prediction on several different display devices. One configuration is a VAX 11/785 connected to a Tektronix 4125 display device. The VAX is connected to the Tek 4125 over a 9600 baud line. In this configuration, it requires about four minutes to do a radar prediction at a twelve mile range. Most of this time is consumed passing data over the relatively slow 9600 baud line. Another configuration is a MicroVAX II with a graphics processor (Parallax 1280 board set) directly on the system bus. In this case graphics data is passed to the graphics processor at a much higher rate. In this configuration it takes about thirty seconds to produce a radar prediction at a twelve mile scope range. This is a speed up of about a factor of eight.

4.11 Electro-Optical/Infrared (EO/IR) Predictions

This function will produce a prediction of the aircraft's EO/IR sensors at critical points along the flight path selected by the planner. The predictions will show what these critical points will look like in the EO or IR spectrums. The EO/IR prediction may be printed for insertion into the CMF. This data will be used in determining terminal area tactics.

The Tactical Decision Aid (TDA) is a computer program developed by the Air Weather Service. It predicts acquisition, lock on, and designation ranges for EO/IR weapons based on target area weather conditions, time of day, and target area tactics. It does not produce a perspective view of the EO/IR sensors. As stated earlier, there is currently no automated data source to feed the TDA program. All data must be entered manually. While the program itself runs fairly quickly, the manual inputting of data is very time consuming. The time required to compute the range data will be around one to three seconds (on a VAX 11/785 or MicroVAX II). However, it will require several minutes to input the weather, weapon, and terminal area tactics data.

For the purposes of this report, an automatic feed of weather data to TDA (or software incorporating it) will be assumed. Processing time to compute acquisition and lock-on ranges is negligible, given that the data is available.

Processing required to produce perspective views in the EO or IR spectrum will not be closely estimated in this report. The processing required to produce 3-D perspective views is probably within the range of two to ten minutes for a 1280 x 1024 pixel display, several times greater than for any other function discussed in this report. A great deal of data, beyond digitized terrain data, will be required to produce perspective views in the EO and IR spectrums.

4.12 Electronic Combat (EC) Asset Modeling

This section will describe the requirements for standoff EC. The next section will discuss the requirements for onboard EC modeling.

This process computes the optimum placement of stand-off EC assets and calculates the effects of EC on the relative threat lethality statespace as shown in Figure 4.12-1. The EC effectiveness may be displayed and used by the planners during route evaluation.

4.12.1 Requirements

The requirements for EC asset modeling are as follows:

(1) Determine the optimal orbit placement to maximize the effectiveness of standoff jamming platforms.

(2) Given the location of jamming platforms, display the effects on the enemy electronic order of battle.

(3) Recalculate the relative threat lethality for a route so that the effects of standoff jamming platforms are included. EC jamming will be included in the leg-by-leg and total route lethality.

Optimizing the locations of EC assets is an extremely difficult problem. No existing programs optimize EC asset location, in the mathematical sense. This includes FLAPS, IMOM, and C3CM BMDA. It is possible to evaluate the effects of standoff jamming both graphically and numerically (in the statespace). There are several ways of interpreting what is meant by "the best EC asset location". One way is to evaluate EC effectiveness with a route for a penetrating aircraft. Then, the most effective EC asset location is the one which minimizes the relative threat lethality for that specific route. This is the interpretation used in this report. The planner can position the EC in several different locations, and evaluate the effectiveness at each one.

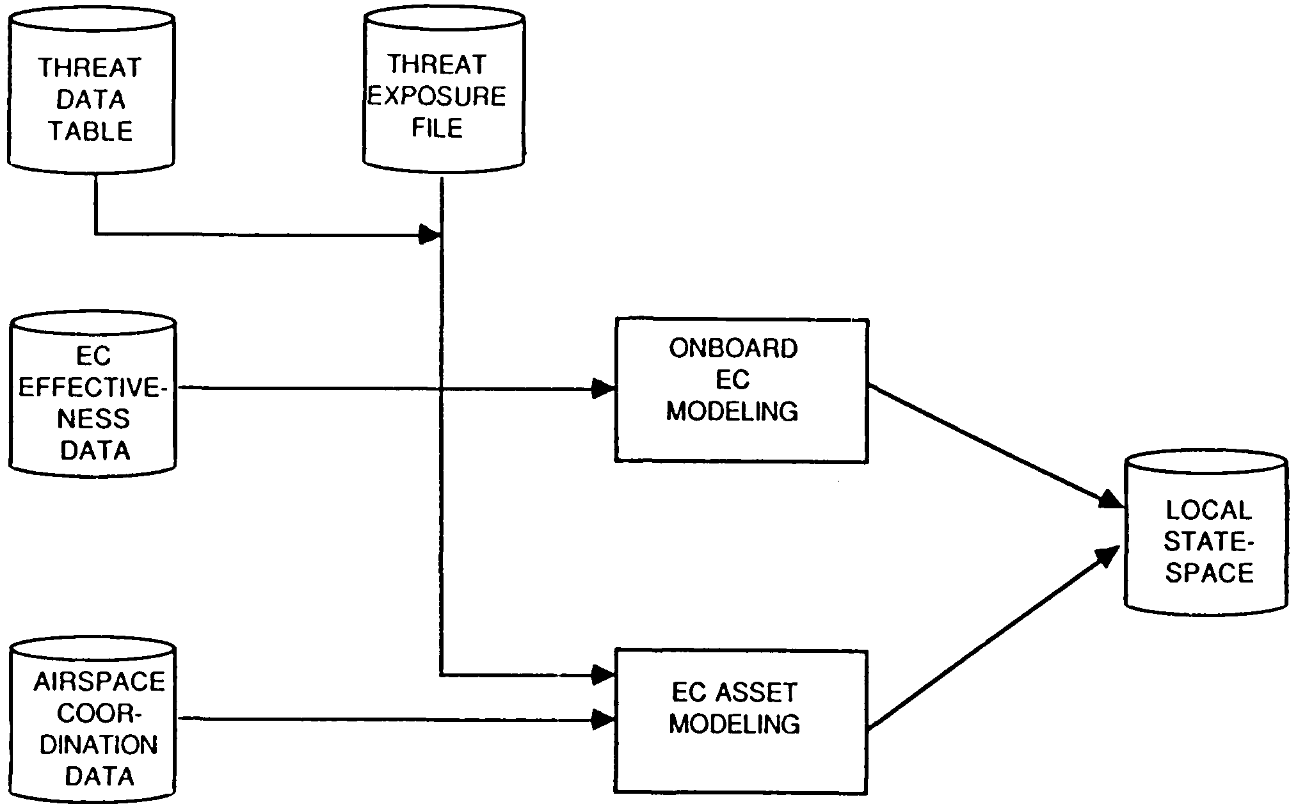

Figure 4.12-1 EC Modeling

The EC can then be assigned to the location that is best (among the alternatives that were evaluated). However, this is very different from determining mathematically where the best location is among all possible locations. This problem is resolvable. It is particularly solvable if the number of possible EC locations is small (for example 2, 3, or 4). However, even for a small problem, the processing required is enormous if the threat array is realistically large. In addition, the problem has to be resolved each time the route for the penetrating aircraft is modified.

For these reasons, the requirement for optimum placement of EC assets will not be considered in this report.

The requirement for evaluation can be met. However, this assumes that the effects of EC can be reflected in the threat models that were used to produce the threat lethality statespace. For jamming, this can be done using a standard signal to noise ratio jamming model. The approach used to reflect the effects of EC in the threat lethality statespace is as follows. The locations of the threat and the jammer are known. For each statespace cell that has a line of sight to the threat, compute whether or not a target in that cell can be seen by the threat's radar, based on a threshold value on the signal to noise ratio. This requires an assumption of the radar cross section of the target in the cell, the jamming power, the radar effective radiated power, and other parameters. This can be done for each cell. For cells that are now "masked" because of the jammer, the threat danger for that threat is subtracted out of the statespace.

4.12.2 Processing Required

Standoff EC effectiveness modeling is computationally and I/O intensive. Again, the amount of processing required for a threat is similar to that required for a "statespace add". In addition, the radar jamming model must be executed for each statespace cell within the coverage of the threat. The number of additional computations required to perform the jamming is approximately 100 floating point operations per statespace cell, based on an analysis of FLAPS software. Remember that many cells may be processed more than once. If a cell is exposed to several radars then the radar jamming formula will be applied to that cell once for each of the threats.

4.13 Onboard EC Modeling

This function will calculate the effects of onboard EC jamming pods on the relative lethality statespace.

4.13.1 Requirements

The requirements for onboard EC modeling are as follows:

(1) Capability to display the effects of onboard fighter jamming pods against the threats. This must be a user selectable option.

(2) Re-calculate the relative threat lethality for a route so that the effects of onboard jamming pods are included. Onboard jamming will be included in the leg-by-leg and total route lethality.

Onboard jamming effects will be processed in two ways. First, they will be processed into the statespace for display. The result will be a suppressed relative threat lethality statespace which may be used to plot the effects of onboard jamming. Second, onboard jamming will be included in the detailed route evaluation. Recall that for detailed route evaluation, the contribution of individual threats to the total threat lethality is computed. If the user instructs the program to use the onboard jamming model, then the danger from the individual threats will be computed including the effects of onboard jamming.

Onboard jamming effectiveness may be computed in two ways. The first and simplest way is to apply a percentage degrade to the threat's relative threat lethality model. For example, a one hundred percent degrade means that the threat is neutralized by onboard jamming. A fifty percent degrade means that the threat is only half as effective in the presence of onboard jamming (in the relative lethality sense). The degrades may be threat system dependent. That is, different types of threats may be affected differently by different types of jamming pods. This can be reflected in a simple matrix.

The second and more complicated model would be to make onboard jamming effectiveness range dependent. That is, the effectiveness of the onboard jamming is dependent on the range from the aircraft to the threat. The relative threat lethality will be degraded, but in a range dependent manner.

The onboard jamming effectiveness model will be used to re-display the danger contours for the statespace and for route evaluation. To display the effects of onboard jamming, the jamming effectiveness model must be applied to the statespace. If the route is known, then the effects can be computed for the threats which expose the route. These threats will be found by performing a detailed route evaluation. The onboard jamming effectiveness will be computed for those threats and they will be degraded in the statespace. The resulting statespace can then be displayed. Either the simple, or range dependent model may be used.

It is possible to compute a jammed statespace without a route. Here, all threats which can be suppressed using an onboard jammer would be degraded in the statespace. The resulting statespace could be displayed. The suppressed statespace could also be used for route optimization. These will be the optimum routes considering onboard jamming. While this is not a requirement at this time, it is a straightforward extension of the simple jamming model.

For route evaluation, the route is known. This means that the threats which expose the route can be determined. For each threat that exposes the route, the effects of onboard jamming can be computed. Either the simple, or range dependent model may be used. The result will be a threat by threat breakdown of the total relative threat lethality for the route, including onboard jamming.

4.13.2 Processing Required

Computing the effects of onboard jamming on the statespace is very similar to performing a "statespace add". The threat observability must be read in from disk, and a portion of the threat's lethality must be subtracted out of the statespace. This process is almost identical to the one that put the threat in the statespace in the first place. If the jamming model is range dependent, then more processing will be required. Table 4.5.3-1 contains the processing requirements for adding different sized threats to the statespace. If the simple onboard jamming model is used, then this table shows the approximate amount of processing required to compute onboard jamming effectiveness for a single threat. Of course, the processing required to evaluate a route depends on the number and size of the threats that the route flies through.

If the range dependent model is used, then the number of arithmetic operations, in Table 4.5.3-1 should be doubled. This is a rough approximation. The I/O required will not change.

4.14 Three-Dimensional Modeling

This function will produce three-dimensional perspective views, or "out the window" views at critical points along the flight path.

4.14.1 Requirements

The requirements for this function are as follows:

(1) Produce a three-dimensional perspective view based on DMA terrain data and/or overhead photography.

(2) The altitude and axis of the view may be user specified.

(3) The user shall be able to "fly" selected portions of the route using the visual display.

The requirements for this function will not be estimated in this report. The processing required will be substantial. Three dimensional perspectives of DMA terrain data alone is probably manageable on a small computer. However, digitizing and combining digital maps and the terrain data is likely to be very processor intensive. Producing three-dimensional perspective views in a reasonably short amount of time will probably require more processing capacity than any of the other functions discussed in this report.

Special purpose graphics systems do exist that will drive three-dimensional perspective views. Requirement 3 suggests that animation is also required. In this case, special graphics processors will be required.

4.15 Conventional Weapons Delivery

This function will generate ballistic and weapons delivery information required for all conventional free-fall and cluster bomb unit (CBU) ordnances.

Weapons delivery software has been developed by the TAF. The requirements to run this software are unknown at this time.

Weapons delivery planning probably requires much less processor resource and storage than threat analysis, route generation, and route analysis.

4.16 Nonconventional Weapons Delivery

This function will generate weapons delivery information required for all nonconventional ordnances.

It is not known if nonconventional weapons delivery software is available at this time. The requirements for this software are also not known.

As above, weapons delivery planning probably requires much less processor resource and storage than threat analysis, route generation, and route analysis.

4.17 Digital Map and Imagery Display

This function will produce displays of navigation charts and digital photographic images.

4.17.1 Requirements

The requirements for map and imagery display are:

(1) Navigation chart displays must be available during the planning process. Navigation charts must be available in scales from 1:50,000 to 1:1,000,000.

(2) The system must be capable of displaying routes, threats, restricted operating zones, and other data on the map display.

(3) The system must have the ability to electronically update map data (CHUM).

(4) The system must be capable of displaying digitized photographic images.

(5) The system must be capable of receiving digital photo images over a local area network.

These requirements are similar to those for combat mission folder generation. A video disk capability and indexing software is required. CHUM data will be stored in a digital data base. It will be displayed automatically for the appropriate maps.

Displaying digitized photographic images is not taxing on the graphics device or the processor. However, these images can consume enormous amounts of disk space. A typical high resolution display device has a resolution of 1024 x 1280, with 8 bit planes to provide 256 colors. To store a digital image requires 1024 x 1280 x 8 bits (10,485,760 bits), or 1.3 million bytes. Clearly, if a large number of digital images are required, then this will drive the disk storage requirement.

A low resolution black and white image could require as little as 640 x 480 x 4 bits (1,228,800 bits) or 153,600 bytes.

The requirement to receive digital photo images over a local area network is very significant. In order to receive 1.3 million bytes in a reasonable amount of time, a high-speed transmission capability is required. This volume of data is much higher than any other of the system inputs. If a high resolution color digital image was transmitted over a 9600 bit per second line, then it would require eighteen minutes to receive it. Recall that most other inputs can be received in a few seconds.

Requirements for the LAN are out of the scope of this report. In addition, special high-speed communications capability will not be assumed in this report because such capability at the wing and squadron appears to be unlikely in the near future. The requirements for transmitting digital images will therefore not be included in this report.

4.18 Data Transfer Cartridge Loader/Reader (DTC L/R) Interface

This function will transfer all data relevant to mission computer initialization and initialization of programmable munitions from the MPS Follow-On to the DTC L/R. The DTC L/R will then write a data transfer cartridge which may be taken to the aircraft. This function requires that the data be reformatted to match the data structures within the DTC L/R. In addition, an electronic interface between the MPS Follow-On and the DTC L/R will be established. The DTC L/R is a peripheral which will not be discussed in this report. The volume of data that is required to be passed to the DTC L/R is described as follows.

The volume of data that will need to be transmitted to the DTC L/R is not very large. In FLAPS, a route requires about 7200 bytes of storage. Over a 9600 bit per second line, this will require about six seconds. The amount of data required to initialize the weapons systems is not known at this time, however, it is probably reasonably small also. This suggests that a standard serial interface is sufficient to transmit data to the DTC L/R. This, together with special interface software, will meet the requirements for the DTC L/R interface.

4.19 User Interface

This function includes all processes and data which are used to produce menus and graphic displays for the user, and to interpret inputs from the user. This function may require substantial storage resources, especially if on-line help is provided. Table 5.2.3-1 indicates a requirement of 0.6MB for the user interface.

4.20 Data Base Management

This function includes all processing and data required to manage the MPS Follow-On data base. This includes processes to maintain data input by the user or from other systems, and data produced internally by the system. The data base manager may require substantial storage resources.

4.21 Mass Storage Requirements

The storage requirements for each data file shown in Figure 3.2-1 are discussed below. Table 5.2.3-1 shows that the data base management system will require 0.4MB.

4.21.1 The Threat Data File

The threat data file is a record (table) oriented data structure which contains data on threat locations, observation times, and uncertainty. The storage required is shown in Table 4.21.17-1. This figure of 15.2KB assumes that there are one hundred threats in the scenario. It is likely that there will be more than 100 threats in the scenario. Even so, the threat data file will be very small relative to the larger data files, like the statespace.

4.21.2 The Generic Threat Data File

The generic threat data file is a record (table) oriented data structure which contains threat model data for each generic type of threat. This includes the altitude dependent down-range and cross-range relative danger indices. The storage required is shown in Table 4.21.17-1. This 45KB assumes that there are ten different generic threat types in the scenario.

4.21.3 Scenario Data

A number of data items are required to describe the scenario. This includes the scenario boundaries, the quantization level, and other factors. In FLAPS, this data is spread among six different tables (or record oriented data structures). Each of these tables contains two records; the first containing header information and the second containing data. The storage required is shown in Table 4.21.17-1.

4.21.4 The Byte Packed Terrain Data File

The terrain data file is a special array oriented data structure which contains byte packed terrain elevation data. A byte packed terrain file is produced using special software and

DMA DTED data files. The data is subsampled and stored in a compressed (byte-packed) form in order to reduce storage requirements. The degree of subsampling is determined when the byte packed terrain file is created. The degree of subsampling is also affected by the area on the earth that the data is for. The number of DTED samples per degree longitude that DMA uses changes at 50 and 70 degrees north latitude.

The following data is based on a byte packed terrain file containing 200 samples per degree longitude and 400 samples per degree latitude. The data was subsampled at a rate of 3 to 1 in latitude. Because the file straddles 50 degrees north latitude, the degree of subsampling in longitude varies.

NUMBER OF SAMPLES PER SQUARE DEGREE = 80000

NUMBER OF BYTES PER SAMPLE = 2

NUMBER OF BYTES OF DISK STORAGE REQUIRED PER SQUARE DEGREE = 160K

DISK STORAGE REQUIRED FOR A FILE COVERING:

7-17 DEGREES EAST LONGITUDE (10 DEGREES)

48-54 DEGREES NORTH LATITUDE (6 DEGREES) OR 60 SQUARE DEGREES = 9.6 MEGABYTES.

4.21.5 The Threat Danger Statespace File

The threat danger statespace file is an array oriented data file which contains statespace data at multiple altitudes. The statespace contains the direction dependent relative danger values for each cell in the statespace. FLAPS uses eight directions. The size of the statespace array depends on the planning area being covered and the quantization level (or statespace cell size) being used. There is one floating point word for each direction at each cell at each altitude.

Based on a 4.5 nm x 4.5 nm cell size, near 50 degrees north latitude, (13.33 cells / degree latitude, 8.48 cells / degree longitude):

NUMBER OF BYTES OF DISK STORAGE REQUIRED PER

SQUARE DEGREE = 3.617K

For a multiple altitude statespace covering

48-53 degrees latitude (5 degrees)

8-16 degrees longitude (8 degrees) at 4 altitudes,

NUMBER OF BYTES OF DISK STORAGE REQUIRED = .579 Megabytes

4.21.6 <u>The Threat Exposure Data File</u>

The FLAPS terrain masking algorithm computes the Minimum Observable Altitude (MOA) at points along rays about a threat. This data is stored in radial coordinates in the local masking array file. After the terrain masking has been completed, the data is converted from radial coordinates to statespace coordinates and is stored in the threat exposure data file. The terrain masked data is maintained for each threat in order to facilitate electronic combat modeling.

The threat exposure data file consists of a large header, followed by the actual MOA data. MOA data is stored as 16 bit integers. The actual amount of storage required for the threat exposure data will depend on the number of threats, the size (radius) of the threats, and the quantization level. Currently, four MOA samples are stored for each statespace cell. The data is interpolated when computing danger within the cell.

BYTES REQUIRED FOR TOBS HEADER = 40K

BYTES REQUIRED FOR ONE 20 NM RADIUS THREAT AT 4.5 NM

CELL SIZE = 1.5K

BYTES REQUIRED FOR 100 20 NM RADIUS THREATS AT 4.5 NM

CELL SIZE (INCLUDED FILE HEADER) = 290K

4.21.7 The Local Masking Array File

The FLAPS terrain masking algorithm computes the Minimum Observable Altitude (MOA) at points along rays about a threat. This data is stored in radial coordinates in the local masking array file. After the terrain masking has been completed, the data is converted from radial coordinates to statespace coordinates and is stored in the threat exposure data file. The local masking array is only used as a temporary buffer.

BYTES OF DISK STORAGE REQUIRED = .33 Megabytes

4.21.8 The Local Statespace

This is an array oriented data file similar to the multi-altitude Threat Danger Statespace. The local statespace contains only one altitude dimension. It contains the relative lethality values for each cell, at the optimum altitude for each cell. It is produced during the altitude optimization process which takes place during route optimization.

4.21.9 EC Effectiveness Data

This is a record oriented data file. It contains effectiveness parameters for stand-off and onboard EC systems. This includes the effectiveness of each EC system against each type of threat in the generic threat data base. The storage requirements for this file are shown in Table 4.21.17-1.

4.21.10 Tasking Data

Tasking data may be stored as a record oriented data file after it has been processed by the Tasking Data Input task. This file contains the mission number, target (or objective), recommended weapons load, time on target, and other data. The storage requirements for this file are shown in Table 4.21.17-1.

4.21.11 Weather Data

This is a record oriented data file. It contains the coordinates of weather areas and other data. The storage requirements for this file are shown in Table 4.21.17-1.

4.21.12 Airspace Coordination Data

This is a record oriented data file. It contains the coordinates of restricted airspace areas and other data. The storage requirements for this file are shown in Table 4.21.17-1.

4.21.13 Route Data File

This is a record oriented data file. It contains the coordinates of the route turn points and summary route information. It is possible to store routes for reference in the future. These routes are managed by the data base management system, just like the other record oriented data files. The storage requirements for this file are shown in Table 4.21.17-1. A detailed description of the contents of this file can be found in the FLAPS User's Manual Volume II Data Base Specification.

4.21.14 Fuel Flow Data

This is an array oriented data file which contains coefficients for the fuel flow polynomials. These polynomials are the central part of the detailed fuel calculation routes which are used to produce the detailed flight plan. About 18,000 bytes of disk storage are required for each aircraft type.

4.21.15 Map Display Data

Map data is used by the digital map display function. Maps may be stored in two formats. They may be stored in analog form on optical disks (Laserdiscs) or they may be stored in digital form. For more information in this area, the reader is referred to Reference [1]. (See Section 2.0)

An optical disk will not compete with the other data base files for space on the system disk drives. About 150 24"x30" paper maps may be stored on each Laserdisc. The maps that will be displayed on the MSS will be made up of a mosaic of smaller images, or frames (This is because each frame does not have a large enough field of view to display an entire leg of a typical flight path.). About 54,000 frames may be stored on a single Laserdisc.

Maps stored in digital form require considerable disk storage area. Often these maps are stored on high density WORM (write-once, read-many) disks. In this case, digital maps will not compete for storage on the system disk drives. Digital maps are typically stored at a resolution of 150 to 300 points per inch. Each point typically requires a byte of storage. Therefore, a square inch of digital map data digitized at 150 points per inch will require 22,500 bytes of disk storage.

4.21.16 Digital Photographic Data

Digital photographs are discussed in Section 4.17. In that section is a discussion of the storage requirements for digital photographic data. A single photograph may require from 154 kilobytes to 1.3 megabytes of storage, depending on the number of colors and the resolution. Photographic data may be stored on the same video disk, or WORM disk that the map data is stored on. In this case, the photographic data will not compete with other data files for space on the system disk drives. If a WORM drive is used, new photographs may be added only to unused areas of the disk, because it is impossible to erase these disks. Alternatively, the digital photographs may be stored on the system disk drives. In this case, new photos may be added, and old photos deleted. However, a large amount of storage will be required to support these digital images.

The storage requirements for digital photographic data will not be estimated in this report, beyond what has already been stated. The requirements for the number of photographs, color, and resolution are not known at this time. If high resolution photographs are to be stored on the system disk drives, then this will be the major factor establishing the required disk capacity.

4.21.17 Record Oriented Data Storage Requirements

The data storage requirements for record oriented data are summarized in Table 4.21.17-1.

Table 4.21.17-1 Mass Storage Requirements for Record Oriented Data

DATA FILE	WORDS / RECORD (WORDS)	BYTES / RECORD (BYTES)	APPROX. NO. OF RECORDS (RECORDS)	APPROX. STORAGE REQUIRED (KILOBYTES)
GENERIC THREAT DATA	1126	4504	10	45
THREAT DATA	38	152	100	15.2
SCENARIO DATA	303	1212	2	2.4
EC EFFECTIVENESS DATA	429	1716	6	10.3
TASKING DATA	76	304	2	0.6
WEATHER DATA	41	164	10	1.6
AIRSPACE COORDINATION DATA	41	164	10	1.6
ROUTES	442	1768	10	17.7
TOTAL				94.4
TOTAL WITH 200% GROWTH				285

THE ASSUMPTIONS USED TO PRODUCE THIS ESTIMATE ARE DESCRIBED IN SECTION 4.20.

4.21.18 Non-Record Oriented Data Storage Requirements

The requirements for non record oriented data storage are summarized in Table 4.21.18-1. This includes data stored in array format (for example, the statespace), and the terrain data. This data is scenario specific. The reader should refer to the descriptions of the individual data files for more information.

Table 4.21.18-1 Non-Record Oriented Mass Storage Requirements

DATA FILE	STORAGE REQUIRED (KILOBYTES)
DIGITIZED TERRAIN DATA	9,600
THREAT DANGER STATESPACE	579
THREAT EXPOSURE DATA	290
LOCAL STATESPACE	165
FUEL FLOW DATA	18
DIGITAL MAPS	-
DIGITAL PHOTOGRAPHS	-
LOCAL MASKING ARRAY	330
USER INTERFACE	773
TOTAL	11,755
TOTAL WITH 200% GROWTH	24MB

PLEASE REFER TO SECTION 4.20 FOR A DETAILED DESCRIPTION OF THESE DATA FILES AND THE ASSUMPTIONS USED TO PRODUCE THE ESTIMATE ABOVE. APPROXIMATELY 36 MEGABYTES OF DISK STORAGE WILL BE REQUIRED TO PROVIDE A 200% GROWTH CAPACITY FOR THIS DATA.

5.0 MPS FOLLOW-ON HARDWARE REQUIREMENTS

The system block diagram shown in Figure 5-1 illustrates the MPS Follow-On hardware suite. The computer may consist of one or more separate processors connected together to allocate tasks between them and preserve existing hardware if desired. Processing, storage, peripheral and other requirements are described below.

The squadron ICS provides threat, tasking, and other environmental information. Map images stored on optical disks are used to provide a background for the routes generated on the computer by the mission planning software. Map indexing and manipulation software residing in the computer is used to bring the appropriate map images from the disk into the image processor so that generated routes and annotations can be overlaid on the maps displayed on the graphics monitor. Under control of the computer map indexing and manipulation software, the image processor abuts several map images together to form a mosaic with a larger field of view than shown on each image from the optical disk. The image processor also rotates and scales the background and route overlays and displays the resultant maps on the graphics monitor. Combat Mission Folder hardcopy output is either "captured" from the video signals driving the screen by an RGB interface and copied on the printer (while the computer is freed to perform more mission planning tasks) or the CMF image is sent to the printer via a digital interface. The intelligence data and DTC L/R interfaces are mounted within the computer (or in interconnected computers) designated here as the computer system.

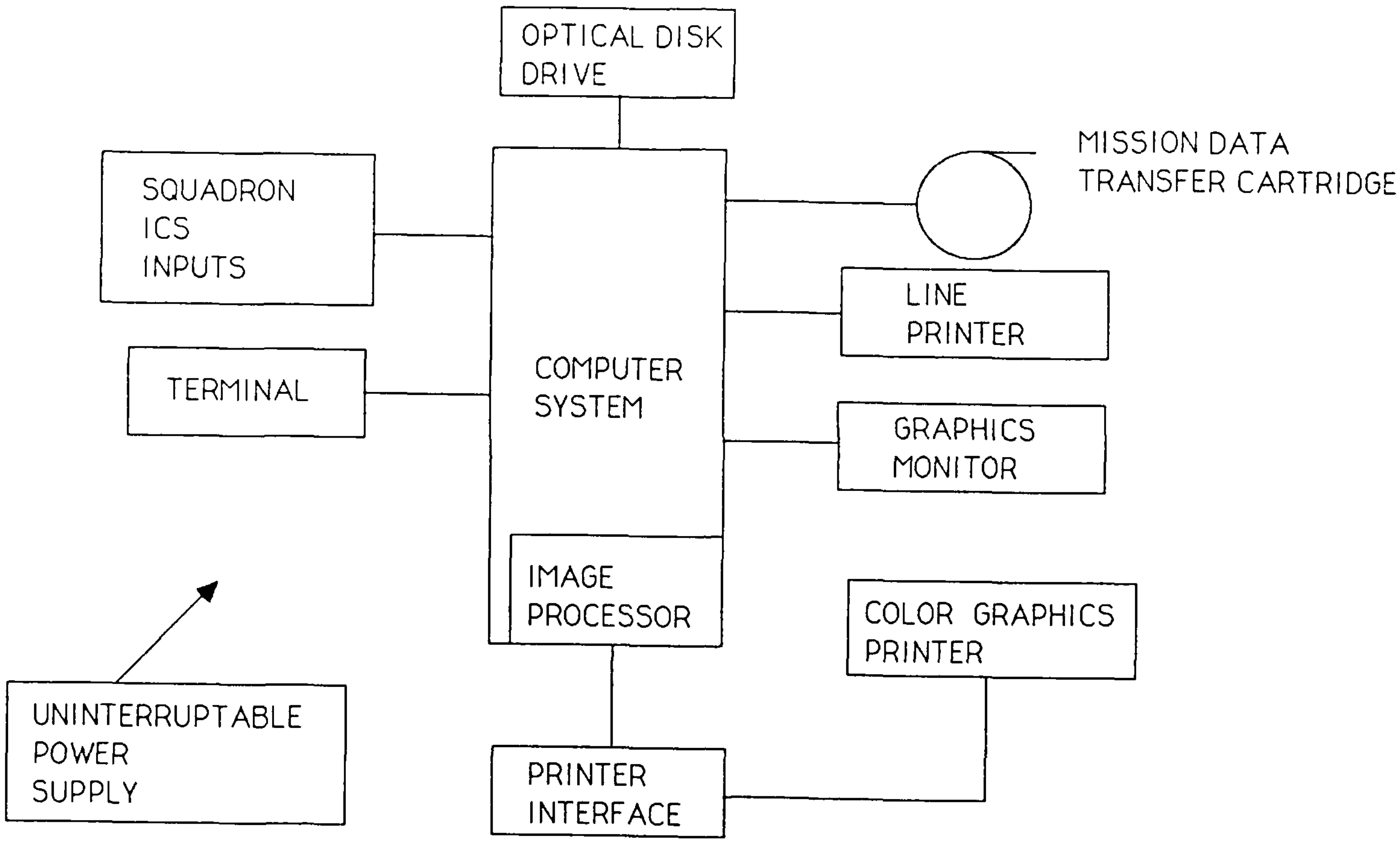

Figure 5-1 Major MPS Follow-On Hardware Components

5.1 MPS Follow-On Processing and Disk Speed Requirements

This section uses a typical scenario to evaluate the processing and storage requirements. It is assumed that intelligence data input, adding 100 threats to the statespace, and threat masking are to be done in one hour of wall clock time. Thirty additional minutes are then allotted for the remaining tasks outlined below.

Thus, the initial conditions necessary for route generation are set up in one hour, and then route generation, modification, evaluation, and CMF generation, etc. are done. The scenario assumes 100 20nm threats, a cell size of 4.5nm x 4.5nm and a statespace covering 48°N to 53°N and 8°E to 10°E. This statespace contains 68 x 69 cells. Computations are done at four altitudes; 200', 1,000', 5,000' and 10,000'. The scenario performance calculations derived from the estimates shown in Section 4 are then compared with a processing of the scenario made on a VAX-11/785.

For the purposes of this report the mass storage requirements will be assumed as cumulative. No new data will be assumed to overwrite old data to conserve storage resources. This will provide a storage requirement that will ensure the growth potential of the system.

5.1.1 Statespace Generation

Generation of the statespace is a highly time consuming operation, so it has been separated from the other tasks, such as route generation and evaluation, that will use the completed statespace. Much of the time taken to generate the statespace will be taken in disk I/O so the disk speed is an important determinant of overall system speed. A tradeoff will be made between CPU and disk speed to meet overall requirements of the system.

Generating the statespace will take one hour and is comprised mainly of statespace adds and threat processing operations. Automated input of intelligence data is assumed here, so the time taken for data input is negligible.

5.1.1.1 Average I/O Block Size

To evaluate the disk speed requirements we have to determine if access time or transfer rate is most important in a typical case. Based on FLAPS, the measured wall clock time taken for I/O of the threat data is much larger than would have been seen if the accesses were made in very large blocks of data. The wall clock times are largely determined by the disk access time rather than by the transfer rate. As an example, consider the masking of a 20nm threat. The wall clock time is 2.477 seconds (Table 4.5-1) for 28,269 words, so the overall, or effective, transfer rate is:

$$\frac{28{,}269 \text{ word} \times 4\text{B/word}}{2.477 \text{ sec} - 1.056 \text{ sec}} = 80\text{KB/sec}$$

This is too slow to be accounted for by a disk that should have an effective transfer rate of around 1MB/sec.

If we assume that the data is brought in via small numbers of blocks at a time, the disk access time becomes the major contributor to wall clock time. Assuming an average access time of 40ms we have at a block size of 512 bytes:

$$\frac{28{,}269 \text{ word/threat} \times 4\text{B/word} \times 40\text{ms/access}}{512\text{Bytes/block } (2.477-1.056) \text{ sec/threat}} = 6 \text{ blocks/access}$$

So the average access for these small threats is only 6 blocks long.

While it may seem as though the transfer rate of the disk is unimportant here, this is just one of many complex disk-intensive operations that may be done as the software evolves. The effective transfer rate requirement should be kept at .5MB/sec to keep this from becoming a limiting factor in the system speed.

The threat data (for an arbitrary threat size at an arbitrary location) is spread around many different parts of memory. The I/O speed is highly dependent upon the efficiency of the computer's operating system and the programming methods that were used, as well as the access time and transfer rates of the disk. The system could be "tuned" to a particular scenario, but as the conditions and software change the tuning may prove to be a disadvantage. It is preferable to specify a computer that has the generalized capabilities necessary for these types of tasks. One of these requirements is fast disk access time. There is a tradeoff between disk and CPU speeds for these types of operations, since each component of the process is large enough to determine the overall system speed. Faster disks are generally less expensive and more readily available than faster CPUs. Current technology will allow the specification of a 20ms disk access time, and this will meet the system requirements, so a 20ms access time should be specified.

5.1.1.2 Disk and CPU Speeds

For a scenario of 100 20nm threats, Tables 4.5.2-1 and 4.5.3-1 indicate that 14MB of data will be transferred and 110 MFPOs will be performed.

Even in this simple case the overall speed can be achieved by different ratios between I/O and processing speed. If we assume a balance between the two speeds, then 110 MFPOs must be done in 1/2 hour. For a 200% growth potential, this amounts to a CPU requirement of about 0.9 MFLOPS. The access time for a 512B block size and 200% growth must be:

$$\frac{\text{6 blocks/access} \times 512 \times 1800 \text{ sec}}{3 \times 14\text{M}} = 133\text{ms}$$

These requirements do not specify a fast computer by today's standards. The requirements placed upon the computer go up roughly as the square of the threat radius, so for 100 60nm threats, the processor would have to run at roughly 8 MFLOPS and the disk access time would have to be about 15ms.

As a check, Tables 4.5.2-1 and 4.5.3-1 show the wall clock time taken for the VAX 11/785 to perform these operations to total 940 seconds, about 1/4 hour for this scenario. The VAX 11/785 is not taxed very heavily by the scenario, and the VAX disk access time and CPU floating point speed agree with the estimates.

No clear worst case scenario exists, and the figures indicate "ballpark" estimates of processing speeds that are within the range that is achievable by computers today. As stated above, the average disk access time should be 20ms, although 40ms would be acceptable for most scenarios. The processing speed should be a minimum of 1.0 MFLOPS, but a faster CPU would be preferable.

5.1.2 Routing Operations

The second part of the processing is the route generation, evaluation, and output of the CMF, etc. This part will take 30 minutes, and is largely dependent upon how much time the operator takes to visually evaluate the route and enter changes. As long as the processing requirements for statespace generation are met, the CPU speed will be sufficient for routing. The largest part of the machine time is expected to be taken in printing the CMF (about 5 minutes for the maps and about 2 minutes for printing two radar predictions). An additional minute or two will be taken to generate the radar predictions on a 0.9 MFLOPS processor. The other times are small, as shown by the route generation example in Section 4.6.5.

5.1.3 Video/Graphics Processor Requirements

Graphics and video processing are discussed in Section 4.9. The requirements for the processor are as follows:

NTSC video input capture capability
- 1/30th of a second capture time per frame
- time base correction circuitry.

Block image transfer at 12M pixels/second.

2K x 2K x 8 bit image memory

8 MIPS processing speed

5.2 MPS Storage Requirements

Disk and main memory requirements for the general purpose processor are estimated in this section. Optical disks and memory used in a graphics processor are shown elsewhere.

5.2.1 Mass Storage Requirements for the Program

The approximate mass storage required to store the program executable, the object library, and the source code is shown below. This estimate is based on MPS Follow-On functions that are similar to those found in the FLAPS software.

FILE	STORAGE REQUIRED (megabytes)
Executable	6.1
Object Library	3.2
Source Code	2.8

The only file that is absolutely required to execute the program is the executable. The object library is required to recreate the executable due to a change in the operating system or the graphics libraries. In an operational system it should not be necessary to maintain the object library on the system. The source code is also not necessary to operate an operational system.

5.2.2 Mass Storage Requirements for the Data, and Total Required Disk Space

Sections 4.21.17 and 4.21.18 show that the overall requirements for data storage on disk are approximately 12MB. The requirement is driven largely by the DTED.

Adding the storage requirements for the executable and a source or object file brings the total to 20MB exclusive of the operating system. The minimum disk space would therefore by 60MB for a 200% growth capability, but it should be far higher in specifying a system that will be useful for future development. Disk capacity should be much larger than the estimated requirements of FLAPS for several reasons. The operating system and other programs and data will take considerable space. Disk space is often partitioned into many sections or disks to avoid writing over important information and to decrease access time by searching over smaller areas of a disk. As a disk or partition fills up, files become increasingly fragmented as the operating system searches for small areas to store or retrieve data from. This fragmentation can cause the computer speed to slow to a crawl. Disk capacity is relatively compact and inexpensive, so extra capacity is highly recommended. A minimum of 200MB is easily obtained using current technology and more is recommended. The operational system should have a minimum of 200MB of disk space available.

5.2.3 Main Memory Requirements

The approximate main requirements for each of the twenty MPS Follow-On tasks are listed in Table 5.2.3-1.

These numbers are, in most cases, based on functions similar to those found in FLAPS. For conventional and nonconventional weapons delivery, EO/IR predictions, flight plan generation, CMF preparation and DTC L/R interface the main memory requirements are estimates and it is assumed most of these functions will not require large amounts of main memory.

Table 5.2.3-1 Main Memory Requirements

FUNCTION	APPROXIMATE MAIN MEMORY REQUIREMENT (Kilobytes)
1. Threat Data Input	52
2. Mission Tasking Input	2
3. Airspace Coordination Data Input	2
4. Weather Data Input	2
5. Relative Threat Lethality Processing	1624
6. Route Generation	480
7. Route Evaluation and Threat Analysis	10
8. Flight Plan Generation	50
9. Combat Mission Folder Generation	50
10 Radar Prediction	1950
11. EO/IR Predictions	50
12. EC Asset Modeling	10
13. Onboard EC Modeling	10
14. Three-Dimensional Modeling	NOT SIZED
15. Conventional Weapons Delivery	20
16. Nonconventional Weapons Delivery	20
17. Digital Map and Imagery Display	20
18. DTC L/R Interface	20
19. User Interface	612
20. Data Base Management	440
Total	5400

Caution must be taken when considering the graphics intensive functions. These include CMF generation, digital map and imagery data, and 3-D modeling. In order for these functions to be practical, special graphics hardware must be available. This special graphics hardware may have a large amount of memory and processing power available internally. In other words, a graphics board (or boards) may contain processors and memory for manipulating graphics. This memory is not available to the general purpose processor. The estimates above are for the general purpose processor only and do not include any requirements for the graphics processor. With a graphics processor, these functions will not place a large load on the general purpose processor.

Three dimensional modeling is not estimated. However, the main memory requirements for three-dimensional modeling may be fairly small for the general purpose processor.

The 612 kbytes of main memory required for user interface includes 239 kbytes required for a Tektronix GKS graphics library. While GKS may not be required for the MPS Follow-On, it is assumed that some graphics software will be used. It is further assumed that this graphics software will require about as much main memory as this GKS library. For a 200% growth capability, the minimum main memory required is 16MB.

5.3 MPS Peripheral Requirements

Peripherals include a line printer for Form 691 output, a video terminal for system operation, a color monitor for viewing and manipulating the mission plan's graphics, a color graphics printer for the CMF hardcopy output and the DTC L/R interface which has been specified and built previously.

5.3.1 Line Printer Requirements

A typical Form 691 contains approximately 2-3 textual pages, and should be printed in less than 1 minute. The largest form would contain about 10 pages. The Form 691 lines are approximately 5 inches wide and the font should print 12 characters per inch. At 6 lines per inch, each page contains about 48 printed lines.

For a Form 691 with 2.5 pages of printing, the nominal printing speed is:

12 characters/inch x 5 inches/line x 48 lines/page x 2.5 pages/60 seconds = 120 characters/second. For 200% growth, the printing speed should be at least 360 characters per second.

If ASCII code is used to transmit the characters to the printer, each character to be printed will take two ASCII 8-bit characters with a start and a stop bit. This ASCII bit stream is at a rate of 360 printed characters/second x 2 ASCII characters/printed character x 10 bits/character = 7800 baud. An RS-232 link of at least 9600 baud will be sufficient although a Centronics parallel interface (100K baud burst rate) would work much better.

5.3.2 Optical Disk Requirements

The optical disk drive must store map images of the area of interest. The drive must be capable of rapidly accessing a desired image under control of the computer. The images will be used to form the CMF strip chart background maps, so the images must be clear and undistorted enough to use as a substitute for paper maps.

Reference 1 discusses analog and digital map disks and drives. The conclusions of that reference are valid for this application, i.e., analog map disks are available, are of acceptable quality, and are relatively inexpensive. They should be used until digital map disks are available for the areas of interest.

The requirements of the disk drive are as follows:

Average access time $\leq$ 1.5 seconds

RS-232C control of operation

12" diameter Laserdisc format

5.3.3 Color Printer Requirements

This printer will produce the hardcopy CMF output. As discussed in Section 4.9, present printers are slower than would be ideal. As the technology improves this would be a recommended area for improvement in system speed. It is desirable to maintain a flexible interface to the printer so that the printer may be upgraded as faster printers are developed. For the present, the following requirements are applicable.

Color Printer Interface

If an RGB interface as described in Section 4.9 is to be used, the following requirements apply, otherwise, the printer connection is made directly to the computer via a Centronics parallel interface.

<15 second image capture time

256 colors ($\geq$ 4096 colors are recommended for best quality)

RS-232C control of all operations is desirable, as opposed to front panel control

Storage for one 1280 x 1024 pixel image minimum

X1, X2 and X3 magnification

Centronics parallel output to printer

<u>Color Printer</u>

Centronics parallel data and control input, or direct RGB input

Minimum print size 8.5 x 11"; and 11 x 17" print size availability is desirable

Print time $\leq$ 60 seconds for 8.5 x 11" print
$\leq$ 90 seconds for 11 x 17" print

256 colors minimum, 4096 colors desirable

$\geq$ 300 dots per inch resolution

X1, X2 and X3 magnification, if not featured in the interface

5.4 MPS Follow-On Communications Requirements

As stated in Sections 4.1, 4.2, 4.3, and 4.4, 9600 bps serial lines are sufficient for receiving threat, tasking, airspace coordination, and weather data from the squadron ICS. An additional serial port may be needed for a line printer, another may be required for either a color printer or a color printer interface, a third one will be needed for the optical disk drive, and lastly, one will be needed for the DTC L/R. Therefore, allowing for a 200% growth margin, the MPS Follow-On must provide eight, 28.8K bps, asynchronous serial ports. The MPS Follow-On should also provide two parallel ports.

5.5 MPS Follow-On Security Requirements

The MPS system must use TEMPEST certified hardware. This is not expected to be a difficult requirement to meet. Many of the components are presently available in TEMPEST certified form.

The system must be capable of processing at the TOP SECRET level. This requires that the computer and its associated communication links be protected via physical security methods such as guarded and restricted entry areas.

5.6 MPS Follow-On Environmental Requirements

The system must be easily transportable and use power sources that are available in various parts of the world. Reference 3 states 2-man portability as a preference. Reference 4 gives 2-man portability as a definite requirement. Reference 3 states that uninterruptable power supplies (UPS) and spiker boxes will be unit supplied items. Reference 4 shows these items as part of the MSS with the additional requirements that the power must be uninterruptable for at least ten minutes, and that the system must be capable of withstanding voltage fluctuations of plus or minus 10 percent of the assigned voltage without requiring reinitialization or losing data.

The ten minute UPS requirement is the most stringent. Without special purpose power-down circuitry built into the MPS Follow-On, the system will require an estimated 2KVA of power. The UPS will consist of a controller and battery pack, each may be in a separate cabinet to distribute the weight. Most of the weight of the UPS is normally in the battery pack, which is expected to weigh approximately 200 lbs. Different UPS controllers may be required for operation from 50Hz versus 60Hz sources.

5.7 MPS Follow-On Reliability and Maintainability Requirements

The following reliability requirements are stated in Reference 4, the TAF MPS Statement of Operational Need.

(a)	Mission Reliability (24 hours/day for 30 days)	90.0 Percent
(b)	Uptime Ratio	99.9 Percent
(c)	Mean Time Between Critical Failures	6834 Hours
(d)	Mean Downtime	2.0 Hours
(e)	Mean Time Between Maintenance (preventive)	1000 Hours
(f)	Combined Fault Diagnostics (built in test, manual test, technical test)	100 Percent

Maintenance shall be accomplished by removal and replacement of line replaceable units (LRUs). Line replaceable units will include the computer; disk drives, such as the optical disk drive; terminal; video monitor; DTC L/R and the color graphics and line printers.