Report No. Volpe Center-DTS56-TMS-002

ENHANCED TRAFFIC MANAGEMENT SYSTEM (ETMS)

Functional Description

Version 5.0

June 30, 1995

Volpe National Transportation Systems Center
U.S. Department of Transportation
Kendall Square
Cambridge, Massachusetts 02142

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

AGENCY USE ONLY (Leave Blank)	REPORT DATE June 30, 1995	REPORT TYPE AND DATES COVERED Manual

TITLE AND SUBTITLE Enhanced Traffic Management System (ETMS) Functional Description, Version 5.0	FUNDING NUMBERS FA-5M6 A-5150
AUTHORS ATMS/ETMS Development and Documentation Teams	
PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Volpe National Transportation Systems Center, Automation Applications Division (DTS-56), Kendall Square, Cambridge, Mass. 02142	PERFORMING ORGANIZATION REPORT NUMBER Volpe Center-DTS56-TMS-002
SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Federal Aviation Administration, AUA-500 800 Independence Avenue SW Washington, D.C. 20591	SPONSORING/MONITORING AGENCY REPORT NUMBER
SUPPLEMENTARY NOTES	
DISTRIBUTION/AVAILABILITY STATEMENT	DISTRIBUTION CODE

ABSTRACT

This document is a functional description of Version 5.0 of the Enhanced Traffic Management System (ETMS). The ETMS is an automation system for supporting the strategic management of air traffic. The ETMS is being developed and maintained in conjunction with the Advanced Traffic Management System (ATMS), a research and development program to explore new concepts in traffic management systems.

SUBJECT TERMS Aircraft Situation Display, User Interface, Estimated Departure Clearance Time, Flight Table Manager, Alert Server Process, Flight Database, Traffic Demands Database, Schedule Database			NUMBER OF PAGES 148 PRICE CODE
SECURITY CLASSIFICATION OF REPORT Unclassified	SECURITY CLASSIFICATION OF THIS PAGE Unclassified	SECURITY CLASSIFICATION OF ABSTRACT Unclassified	LIMITATION OF ABSTRACT Same as report

NSN 7540-01-280-5500

Standard Form 298

Foreword

This document provides a functional description of Version 5.0 of the Enhanced Traffic Management System (ETMS). The ETMS is an automation system for supporting the strategic management of air traffic. The ETMS is being developed and maintained in conjunction with the Advanced Traffic Management System (ATMS), a research and development program to explore new concepts in traffic management systems.

The work presented here is sponsored by the Federal Aviation Administration (FAA) under the guidance of William L. Umbaugh, Delois Smith, and Tommie Tyson of AUA-500 as well as Tim Fleming and Gilbert Armbruster of ATM-500. The work is being performed by the Volpe National Transportation Systems Center (Volpe Center), Office of Operations Engineering, Automation Applications Division (DTS-56) under PPA FA-5M6, managed by Richard D. Wright, Division Chief. This document was written by Kenneth Howard of Arcon Corporation, a sub-contractor to Unisys Corporation, and revised and edited by Michael Potash and Mary Costello of Unisys Corporation.

Acronyms

ACARS	ARINC Communications Addressing and Reporting System
ACES	Adaptation Controlled Environment System
ADS	Automated Dependent Surveillance System
AF	Flight Plan Amendment message
ARINC	Aeronautical Radio Incorporated
ARRD	Arrival Delay prediction
ARTCC	Air Route Traffic Control Center
ARTS	Automated Radar Tracking System
ASD	Aircraft Situation Display
ASP	Alert Server Process
ATCSCC	Air Traffic Control System Command Center
ATMS	Advanced Traffic Management System
AZ	Arrival message
BFPSD	Batch Flight Plan Schedule Database
CDT	Controlled Departure Time
CMD	Command process
CNX	SI Cancellation
CONUS	Contiguous United States
COTS	Commercial Off The Shelf
COUNT	Flight Counts
CT	Control Time message
CTA	Calculated Time of Arrival
DOD	Department of Defense
DOTS	Dynamic Oceanic Tracking System

DSS	Data Systems Specialists
DVFR	Defense Visual Flight Rules
DZ	Departure message
EARTS	En route Automated Radar Tracking System
EDCT	Estimated Departure Clearance Time
ERL	Environmental Research Laboratories
ETA	Estimated Time of Arrival
ETE	Estimated Time En Route
ETMS	Enhanced Traffic Management System
FA	Fuel Advisory message
FAA	Federal Aviation Administration
FAATC	FAA Technical Center
FIR	Flight Information Region
FDB	Flight Database
FDBD	Flight Database Distributor
FIXL	Fix Loading
FL	Flight Level
FPA	Fix Posting Area
FPSD	Flight Plan Schedule Database
FS	Scheduled Flight message
FT	Terminal Forecast
FTM	Flight Table Manager
FTP	File Transfer Program
FZ	Flight Plan message
GA	General Aviation
GPS	Global Position System
IAS	Indicated Air Speed

ICAO	International Civil Aviation Organization
ID	Identification
IFCN	Inter–facility Flow Control Network
IFR	Instrument Flight Rules
INM	Integrated Noise Model
INTO/INTI	FAA computer communications protocol
KCW	SA and FT reports provided by Kansas City Weather process
LAN	Local Area Network
LIFP	List Flight Plan
LIST	List Server
LISTO	Flight List Original
LOM	Outer Marker Locations
MAPS	Mesoscale Analysis and Prediction System
MOA	Military Operations Area
NADIN	National Airspace Data Interchange Network
NAS	National Airspace System
NAVAID	Navigational Aid
NDA	NAS Data Assembler
NFDC	National Flight Data Center
NOAA	National Oceanographic and Atmospheric Administration
NOWRAD6	Now radar 6
NOS	National Oceanic Service
NWS	National Weather Service
OAG	Official Airline Guide
OCS	Offshore Computer System
PCU	Protocol Conversion Unit
RPL	SI Replacement

RS	Scheduled Flight Cancellation message
RZ	Flight Cancellation message
SA	Surface Observation
SCDT	Selected Controlled Departure Time
SCT	Southern California TRACON
SDB	Schedule Database
SI	Substitution Request
SID	Standard Instrument Departure route
SIO	Serial Input/Output
STAR	Standard Terminal Arrival Route
SUA	Special Use Airspace
SUB	SI Substitution
TA	Global Position Update message
TDB	Traffic Demands Database
TM	Traffic Management
TMU	Traffic Management Unit
TO	Oceanic Position Update message
TRACON	Terminal Radar Approach Control facility
TTM	Total Traffic Management
TZ	Position Update message
UTC	Coordinated Universal Time
UZ	ARTCC Boundary Crossing message
VT	Verify Time
WAN	Wide Area Network
WXSEND	Weather distribution software
XFER	Communications software

Contents

Figures

Tables

Section 1

Introduction

1.1 Background

The Enhanced Traffic Management System (ETMS) assists the Federal Aviation Administration (FAA) in the performance of air traffic management. Air traffic management is the strategic control of traffic flow; its purpose is to minimize delays and congestion, and to maximize the overall throughput of the National Airspace System (NAS).

Air traffic management is performed through a hierarchical organization. At the top of the hierarchy is the Air Traffic Control System Command Center (ATCSCC) which is concerned with the management of nationwide traffic problems and the coordination and approval of actions taken by the distributed traffic management facilities. The next level of traffic management consists of Traffic Management Units (TMUs) at the 20 contiguous U.S. (CONUS) Air Route Traffic Control Centers (ARTCCs). Each ARTCC TMU is responsible for the management of traffic problems that are within the scope of the ARTCC. The final level of the hierarchy consists of Traffic Management Units (TMUs) at the Terminal Radar Approach Control (TRACON) facilities. The TRACON TMUs manage problems specific to the terminal or terminals under their control.

The ETMS is associated with the Advanced Traffic Management System (ATMS) research and development effort. The goal of the ATMS project is to develop and evaluate new concepts for traffic management automation. The ATMS project calls for the phased development of new prototype functions that are used to establish their feasibility and usefulness. The development of working prototypes allows the FAA to test new functions in an operational context and to provide feedback to their design and implementation before they become operational. When an ATMS function is judged by the FAA to be appropriate and ready to be included into the operational system, it becomes part of the ETMS domain.

The ETMS implementation currently represents the first two of five ATMS development phases. The first phase was the development of the *Aircraft Situation Display* (*ASD*) for graphically displaying current aircraft positions on a national scale superimposed on maps of geographical boundaries and NAS facilities. The second phase was the development of the monitor and alert function. Monitor/Alert provides the means for projecting traffic demands for all airports, sectors, and fixes of interest within the limits of the grid (which extends from 0 to 90 degrees North Latitude, and from 10 to 180 degrees West Longitude) and automatically generating alerts when the projected demands exceed alert thresholds. The Monitor/Alert data is made available to traffic management specialists through the *ASD*.

Future phases of the ATMS development will handle the resolution of traffic problems as follows. Phase three will automatically provide alternative traffic problem resolution strategies that consider the full impact of the resolution on the NAS. Phase four will provide the means to automatically evaluate and select the best strategies. Phase five will handle the problem of automatic distribution of the directives to implement the chosen strategies.

1.2 About This Document

The *ETMS Functional Description* is intended to provide a logical description of the ETMS functions from the traffic management specialist's perspective without going into all the details of how the system is operated or implemented. The ETMS is essentially a data access system; therefore, this description focuses on the types of data available to the traffic management specialist and how those types of data are produced. The *ETMS Functional Description* is organized in 11 sections:

- Section 1 — introduction
- Section 2 — overview of data available to the specialist through the ETMS
- Section 3 — ETMS processing of geographical data
- Section 4 — ETMS processing of airline schedule data
- Section 5 — ETMS maintenance of its flight database
- Section 6 — ETMS interpretation of a flight path (field 10)
- Section 7 — ETMS modeling of flights
- Section 8 — ETMS projection of traffic demands and generation of alerts
- Section 9 — ETMS processing of traffic management data
- Section 10 — ETMS displays of data to the specialist (*ETMS User Functions*)
- Section 11 — overview of ETMS implementation

The *ETMS Functional Description* is written for the reader who is already familiar with FAA terminology, air traffic control, and traffic management procedures. It does not attempt to explain why each type of data was developed or how it is used for traffic management other than in a general way. The document will be best understood by readers who have a working familiarity with the operation of the *ASD* and other *ETMS User Functions*.

1.3 Related Documents

The technical documentation of the ETMS consists of this document and the *ETMS System Design Document*. The *ETMS System Design Document* is a detailed description of the design and implementation of the ETMS hardware and software. Readers who are unfamiliar with the ETMS are advised to read this document, the *ETMS Functional Description*, first.

Users gain access to the ETMS through the *ASD* and other *User Functions*. It is highly recommended that readers of this document familiarize themselves with the operation of the *User*

Functions and the data they present. The operation of the *User Functions* is described in a set of user documents. The *ETMS Tutorial* is a self-teaching document for new users of the system. The *ETMS Reference Manual* is a complete description of the *ETMS* commands and features in a dictionary format for users who are already familiar with the system. The *ETMS Quick Reference Guide* is a convenient reminder of the main commands.

Section 2

Functional Overview

2.1 Summary

The ETMS maintains and monitors national air traffic data and displays the data to traffic management specialists in a variety of forms on a graphics display. An overview of the ETMS system is shown in Figure 2-1.

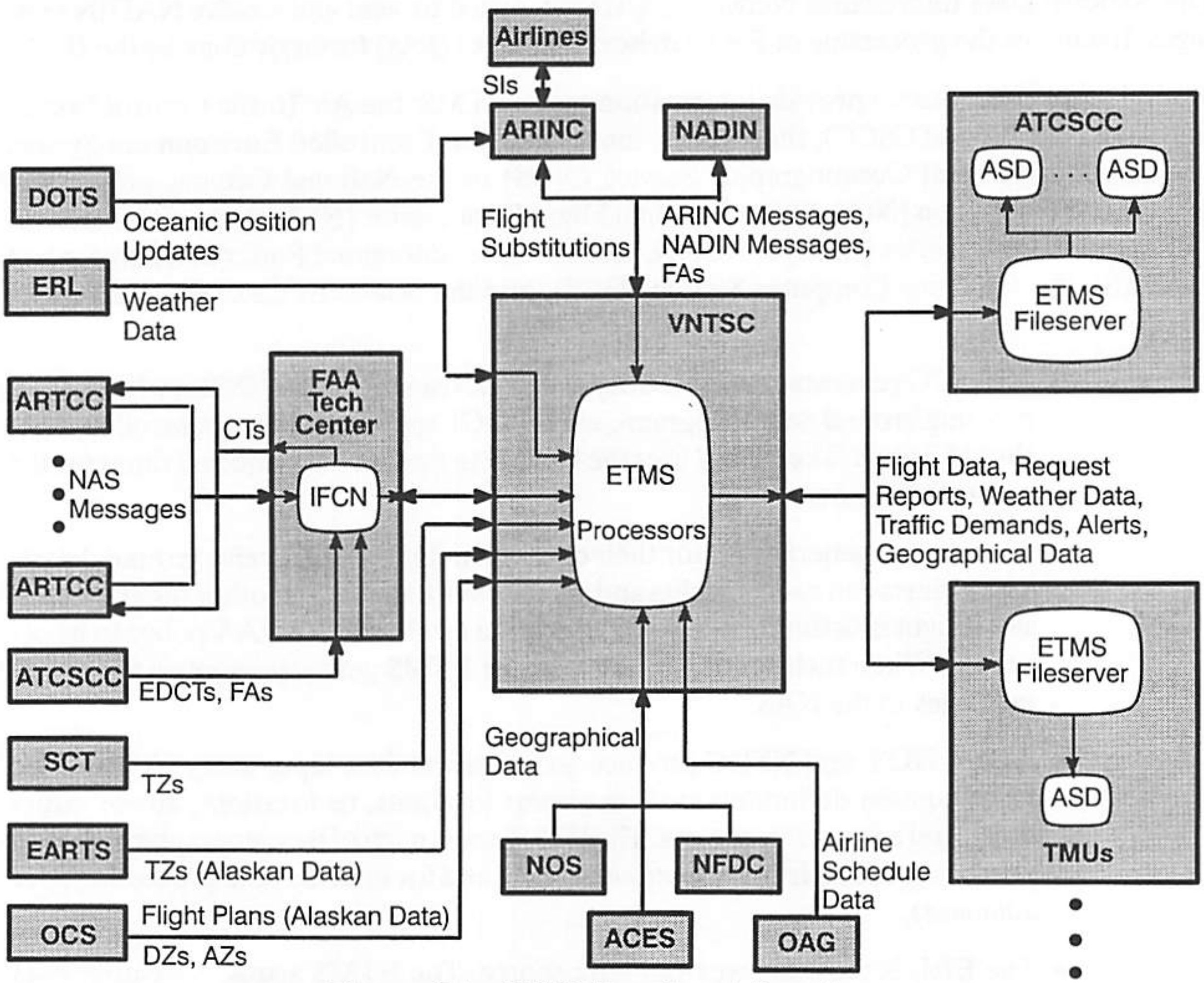

Figure 2-1. ETMS Functional Overview

The ETMS derives air traffic data from several primary types of input: airline schedules from the Official Airline Guide (OAG), messages generated by the NAS computers at the 20 ARTCCs which control the airspace over the contiguous United States (CONUS) and oceanic position messages from ARINC and *directly* from the Dynamic Oceanic Tracking System (DOTS; a portion of DOTS is integrated into the ETMS). The airline schedule data files arrive weekly by communication link from the OAG. The NAS messages are sent in real time to the FAA Tech Center (FAATC) through the Inter-facility Flow Control Network (IFCN), which relays the messages to the ETMS central processing site at the Volpe Center. The ETMS processes airline schedules and NAS messages at the central site and stores the combined data in a set of interrelated databases. The OAG file provides the ETMS with the planned schedules of all flights arriving in the United States or Canada, departing from the United States or Canada, or flying over the United States for the next month. The contents of the NAS messages provide the ETMS with the filed flight plans and the current state of all Instrument Flight Rules (IFR) air traffic in the CONUS. The DOTS messages provide position updates for oceanic flights.

ARINC sends and receives ARINC messages (including oceanic position messages and global postion messages). It handles the substitution requests (SIs) between the ETMS and the users. The National Data Interchange Network (NADIN) is used to send and receive NADIN messages. It handles the processing of Fuel Advisory messages (FAs) formerly done by the IFCN.

Several other data sources provide information to the ETMS: the Air Traffic Control System Command Center (ATCSCC), the airlines, the Adaptation Controlled Environment System (ACES), the National Oceanographic Service (NOS) of the National Oceanic and Atmospheric Administration (NOAA), the National Flight Data Center (NFDC), the Environmental Research Laboratories (ERL) of NOAA, the En Route Automated Radar Tracking System (EARTS), the Offshore Computer System (OCS), and the Southern California TRACON (SCT).

- ATCSCC generates estimated departure clearance times (EDCTs) when implementing ground delay programs; each EDCT specifies when a controlled flight should depart. The ETMS uses the EDCTs to update the projected times for the controlled flights.

- Airlines can generate SIs for their controlled flights that receive ground delays. The airlines can cancel flights and substitute a flight into another flight's arrival slot. Flight substitutions must meet criteria established by FAA policy to be accepted. When such conditions are met, the ETMS generates control time (CT) messages to the NAS.

- ACES, NOS, and NFDC produce geographical data tapes every 56 days. The tapes contain definitions such as airport locations, fix locations, airway structures, and airspace structures. The ETMS uses much of this geographical data to produce graphic displays (*maps database*) and for internal data processing (*grid database*).

- The ERL is the main weather data source. The ETMS acquires weather data from the ERL consisting of grid winds reports, terminal forecasts (FTs), current terminal surface observations (SAs), jet stream information, precipitation radar information, and lightning reports. The grid winds are used to compute flight times; the terminal weather reports, precipitation, jet streams, and lightning are used for display purposes.

- The OCS and the EARTS supply Alaskan data. The OCS supplies flight plan information (FZs), AZs, and TZs; and the EARTS supplies position reports (TZs).
- The SCT supplies position reports (TZs).

The ETMS processes all the available data to maintain the best possible picture of the current state of national air traffic. The ETMS also keeps a history of the flight data and, more significantly, uses the flight data to project the future traffic demands at airports, sectors, and fixes throughout the NAS. The traffic demand projections are continuously updated by the ETMS and compared to alert thresholds (pre-defined traffic levels). When the ETMS observes a projected demand exceeding an alert threshold, a traffic management alert is generated. The flight data, traffic demand projections, and alerts are distributed to sites where traffic management is performed, including ATCSCC, TMUs at the ARTCCs, and TMUs at selected TRACONs.

The ETMS makes several types of data available to the traffic management specialist: geographical data, aircraft situation data, monitor/alert data, request report data, weather data, and traffic management data. The traffic management specialist accesses these types of data through an integrated interface function — the *Aircraft Situation Display (ASD)*. The composition and use of the six data types are summarized in Table 2-1 and described in Sections 2.2 through 2.7.

2.2 Geographical Data

The ETMS uses the geographical data to generate a set of graphic overlays which can be displayed on the *ASD* (*maps database*). The traffic management specialist can turn the overlays on and off by the use of menu selections and keyboard commands. The traffic management specialist can view overlays with other overlays and with the aircraft situation and monitor/alert data. The ETMS also uses the geographical data to draw alerts on the display.

The following map overlays are available on the *ASD*:

(1) Boundaries — The boundaries represent the geographical boundaries for the United States and nearby countries. Within the United States, state boundaries are shown.

(2) Sectors — Sector boundaries can be displayed for low, high, superhigh (ultrahigh), and oceanic sectors. A sector is shown as an outline with the sector name in the middle (at zoom level 3 or more).

(3) ARTCCs — ARTCC boundaries can be drawn for the 20 ARTCCs covering the CONUS, and the Alaska ARTCC. An ARTCC is shown as an outline with its name in the middle. Only the high-altitude ARTCC boundaries are displayed.

(4) TRACONS — Boundaries can be displayed for the TRACONS covering the CONUS. A TRACON is shown as an outline with the TRACON name in the middle (at zoom level 3 or more).

(5) Airways — Jet, Victor, and oceanic airways can be displayed. Airways are shown as line segments with the airway numbers alongside.

(6) Fixes — Arrival and departure fixes can be displayed. A fix is shown by the display of its name at the fix location.

(7) NAVAIDs — All NAVAIDs can be displayed. A NAVAID is shown by the display of its name at the NAVAID location.

(8) Airports — Two sets of airports can be displayed: pacing airports and other airports (airports without numbers in the name, currently about 6000). An airport is shown by the display of its name at its location.

(9) Runways — Runway layouts can be observed for many airports in the United States. Runways may be drawn as a map overlay, or requested for a specific airport.

(10) SUA (Special Use Airspace or Military Area) — Five types of SUAs can be displayed: alert areas, Military Operation Areas (MOAs), prohibited areas, restricted areas, and warning areas. A SUA is shown as an outline with its name in the middle.

The ETMS also uses the geographical data internally to perform the flight path processing and demand projections (*grid database*). Whereas the map overlays contain only the data desired for display, the grid database contains the most complete data available to support the *look-ups* that must be performed to interpret a flight plan. For example, the airports map overlay shows about 6000 airports; however, the internal airports database contains approximately 19,000 domestic landing strips which could conceivably be found in a filed flight plan. In other cases, data types that are used for the flight path processing are not available for display. These are summarized as follows:

(1) Airspace fixes — In addition to the arrival and departure fixes described above, the ETMS keeps a file of all airspace fix names (five-letter) and locations.

(2) SIDs and STARs — The ETMS maintains a file of all Standard Instrument Departure routes (SIDs) and Standard Terminal Arrival Routes (STARs).

The ETMS derives the maps database and grid database from data received every 56 days on magnetic tape from the NFDC, NOS, and ACES, and from static data files provided electronically by the ARTCCs and by the ATCSCC. NFDC provides four tapes: the landing facilities tape, the navigational fixes tape, the airspace fixes tape, and the Special Use Airspace tape. NOS provides two tapes: the airways data tape, and the military training routes tape. ACES provides one data tape via each ARTCC. This data includes Fix Posting Area (FPA) boundaries, airport, and fix information; and other data that the ETMS will use in the future. ARTCCs provide electronic data files of sector capacities, sector types, and sector altitudes for ASD display. The ETMS reads the data files and tapes, extracts the needed data, checks for errors, and processes the data into the internal forms used to generate the map overlays, interpret flight paths, and monitor element demands.

Table 2-1. ETMS Data Type Summary

Data Type	Contents	Uses
Geographical Data	Geographical boundaries Sectors ARTCCs TRACONs Airways Arrival/departure fixes Navigational Aids (NAVAIDS) Airports Runways Special Use Airspaces (SUAs)	Map overlay displays Alert displays Flight path processing
Aircraft Situation Data	Flight ID Aircraft type Altitude Speed Time to arrival Position Origin Destination Flight path text Waypoints Flight events	Flight data display Flight selection Traffic demand displays Request Reports
Monitor/Alert Data	Proposed demand counts Active demand counts Alert thresholds (capacities) Flight lists Alerted airports, sectors, fixes Alert statuses	Traffic demand displays Alert displays
Request Report Data	Airport, fix, ARTCC, sector arrival/departure lists Airport, fix, ARTCC, sector arrival/departure counts Airport arrival delay predictions Airport/arrival fix loading Airport and ARTCC time verification Flight plans	Flight lists reports Flight counts reports Arrival delay prediction reports Fix loading reports Time verification reports List flight plan reports
Weather Data	Surface observations (SAs) Terminal forecasts (FTs) Grid Winds Radar maps Radar labels Jet Streams Lightning	Terminal weather display Flight Modeling Map overlay displays
Traffic Management Data	Flight ground control times Controlled airports Controlled departure times Controlled arrival times Substituted departure times Substituted arrival times	Ground delay program status reports Ground delay program histories

2.3 Aircraft Situation Data

The aircraft situation data describes the state of the airborne flights at a given moment. The normal use of the aircraft situation data is for displaying the *live* flight positions. When displaying live data, the *ASD* indicates the position of each airborne flight by an airplane icon or by a dot. The traffic management specialist can view other flight data by requesting *data blocks* and flight paths. Data blocks show flight data in text form, including the flight ID, aircraft type, flight level, ground speed, estimated time to arrival, and optionally the origin/destination or filed flight path (i.e., field 10). The *ASD* shows flight paths in text form or graphically, overlaid on the maps. The ETMS updates live flight positions approximately every minute.

The ETMS obtains most of the flight information from the NAS messages which consist of flight plans (FZs), flight plan amendments (AFs), departures (DZs), ARTCC boundary crossings (UZs), position updates (TZs), arrivals (AZs), and cancellations (RZs). The ETMS also obtains oceanic position updates for oceanic flights from DOTS. The ETMS extracts values such as flight ID, aircraft type, and ground speed directly from fields of the NAS, DOTS, and ADS messages; the values displayed are the latest ones received for that flight. The ETMS computes displayed flight positions from TZ data for flights within radar coverage of the CONUS ARTCCs. For flights outside of ARTCC coverage, the ETMS computes displayed flight positions from the DOTS position messages. The ETMS also computes displayed flight positions for flights outside of ARTCC and DOTS coverage.

The ETMS draws the flight path by interpreting the *field 10* (filed flight path text). The ETMS uses the grid database and built-in knowledge of the field 10 syntax to determine the latitudes and longitudes of the points at which the flight path bends. These points are then used to generate the flight path display. The ETMS also determines the origin and destination points as well as the airways, sectors, fixes, and ARTCCs that each flight is traversing.

The value of the aircraft situation data is greatly increased by the selectivity which can be imposed on the *ASD*. The traffic management specialist can instruct the *ASD* to display or highlight only flights with certain data values, i.e., certain flight IDs, aircraft types, flight levels, etc. The traffic management specialist can also select flights by geographical criteria, i.e., origin, destination, fix, airway, sector, or ARTCC.

The traffic management specialist can also use the *ASD* to perform certain historical or predictive functions using the aircraft situation data. The traffic management specialist can examine short-term history of flights by using the *history* feature, which causes the *ASD* to leave a trace of the previous flight position when it draws the new position. The traffic management specialist can perform historical analysis by using the *replay* feature. Replay causes the *ASD* to show aircraft situation data for a past time interval. During replay the *ASD* looks and acts just like it is displaying live data except that the display can be started, stopped, and speeded up. The ETMS saves only the most recent six hours of data for replay purposes.

2.4 Monitor/Alert Data

Monitor/alert data allows a traffic management specialist to examine the current or projected traffic demand values at airports, sectors, and fixes, and to be automatically alerted if the projected demands exceed the alert threshold. The demand counts for an airport are the numbers of arrivals and departures per 15-minute interval. Sector demand values represent the number of aircraft in the sector at the peak instant during each 15-minute interval. Fix demands are the number of flights crossing the fix in each 15-minute interval. Alert thresholds are specified in the same units as the demands.

The ETMS computes traffic demands from the airline schedule data as well as the NAS messages used for the aircraft situation data display. The ETMS adds scheduled flights to the traffic demands when the scheduled departure times are within 12 hours of the current time. The ETMS updates the traffic demands with the NAS flight plan data as it becomes available. The ETMS continues to update the demands as the following messages from the NAS are received: cancellations, amendment, departure, boundary crossing, position update, arrival.

The *ASD* displays alerts graphically on the screen. The *ASD* draws alerted elements (sectors, fixes, and airports) in red if the alert is caused by active flights alone, and in yellow if the alert is caused by a combination of active and proposed flights. The *ASD* symbolizes airport alerts by circles at the airport locations, fix alerts by triangles at the fix locations, and sector alerts by filling the sector boundaries. The traffic management specialist can select the types of elements (e.g., high sectors, airports, etc.) for which alerts should be displayed. The traffic management specialist also can enter a time period defining the *look-ahead* time for which the manager would like to receive alerts (e.g., show alerts for the next hour).

The traffic management specialist can view traffic demands for alerted and other elements on the *ASD* through charts, reports, and flight displays. Charts are bar graphs of demand counts vs. the alert thresholds for a specified time range. Airport charts show both arrival and departure values on one chart. Reports are textual listings of the flights which comprise the demand for each 15-minute interval along with the relevant event time(s). Airport reports show arrival and departure times, sector reports show the entry and exit times, and fix reports show the fix crossing times. The flight display shows the positions and flight paths of the aircraft for a specified time interval.

After an alerted element is examined, the traffic management specialist can change one or more 15-minute intervals of the time bar to green. This indicates that the specialist is resolving the alert for that period of time. Specialists at the ATCSCC are authorized to turn an interval green for any element. Specialists at TMUs are authorized to turn an interval green only for elements within their region. During the next monitor/alert update, those intervals that have been turned green are displayed as such on all *ASDs* at all sites.

The *ASD* does not explicitly distinguish between past and projected values when displaying demand values. However, whenever the *ASD* generates time related displays such as a chart of traffic demands vs. time, the *ASD* indicates the current time on the display. The traffic management specialist can therefore easily distinguish the projected values from the historical values.

2.5 Request Report Data

Request reports are textual reports of flights arriving at, departing from, or traversing an airport, fix, sector, or ARTCC; some reports provide data on each individual flight, while others show only the number of flights in each time interval. A traffic management specialist can request reports through the *ASD* or the *Traffic Management Shell (TM Shell)* for times up to 12 hours in the past and up to 28 days in the future for airports and ARTCCs (plus or minus 12 hours for sectors and fixes).

The ETMS allows the traffic management specialist to request six types of reports:

(1) Flight list — The specialist can request a list of flights and associated information and can specify the time interval, the flight data to appear in the report (e.g., flight ID, scheduled arrival time, aircraft type, etc.) and the order in which the flights should appear (e.g., sorted by airline, sorted by proposed departure time, etc.).

(2) Flight counts — The specialist can request a count of the number of flights departing from, arriving at, or traversing an airport, fix, sector, or ARTCC and can specify the time interval for the counts (e.g., number of arrivals per hour, per 15 minutes, etc.) and the categories for counting (flights by aircraft type, flights by airline, etc.).

(3) Arrival delay prediction (ARRD) — The specialist uses this report to assess potential traffic congestion.

(4) Fix loading (FIXL) — The specialist can request a count of the number of flights traversing a specified arrival fix or all arrival fixes for a specified airport.

(5) Verify time (VT) — The specialist can request time verification for flight arrivals and departures, that is, discrepancies between the *actual* arrival and departure times and a specified *time type* (scheduled, proposed, original, or controlled). The specialist can specify the flights, time type, and the report format.

(6) List flight plan (LIFP) — The specialist can request lists of flight plans for selected aircraft.

The ETMS produces the request report from the same data sources as the aircraft situation data and the monitor/alert data. When the requested flight data falls within 12 hours of the present time, the ETMS provides a combination of the schedule data, the NAS message data, and the EDCT data. When the request falls beyond 12 hours in the future, the ETMS provides only schedule data. Note that the scheduled, NAS, and EDCT data are modeled.

2.6 Weather Data

The Environmental Research Laboratories (ERL) supply all weather data to the ETMS.

The ETMS provides terminal forecasts (FTs) and surface observations (SAs) to the traffic management specialist. The specialist can request these reports for any airport through the *ASD.* The ETMS processes the raw FTs and SAs to provide a readable presentation using standard weather terminology. Scheduled updates of SAs arrive from the ERL hourly and include the current cloud cover, precipitation, ceiling, visibility, wind speed and direction, temperature, dewpoint, and barometric pressure. Scheduled updates of FTs arrive every 9 hours and include a detailed forecast for the next 18 hours plus a general forecast for an additional 6 hours. (In addition the ETMS checks for any FT or SA amendments or updates every 10 minutes.)

The ETMS also processes grid winds reports. The grid winds reports provide wind estimates for the airspace over the continental U.S. The grid winds reports divide the country into 5022 blocks and provide wind estimates for fifteen different altitude ranges within each block. Grid winds are received every three hours and project the winds for the next 12 hours. The ETMS uses the grid winds data to estimate the effect of high-level winds for each segment of a flight path (see Section 7.3.2). There is no direct display of the grid winds data on the *ASD*.

The ETMS provides the specialist with a Jet Stream overlay, which shows the perimeters of the jet stream winds and gives the direction, speed, and flight level (altitude in 100 feet levels) of the winds. The ETMS receives this information from the ERL via satellite link every three hours. The ERL generates the jet stream graphics by using the Mesoscale Analysis and Prediction System (MAPS) software with vertical profiler wind and temperature data as well as other wind source data from the National Weather Service (NWS) and the ARINC Communication Addressing and Reporting System (ACARS).

The ETMS provides two overlays that show precipitation information: Radar Map and Radar Labels. The radar summary displays the perimeters of precipitation areas. Radar labels identify the type of precipitation in each area and the vertical height, direction, and speed of the area. The NWS collects source data at sites all over the Unites States and sends the combined information via the ERL every hour to the ETMS and to all Weather Service Forecast Offices. NOWRAD6 is another weather feature provided by the ETMS.

The ETMS provides lightning information , that is, location of lightning strikes, to the specialist every five minutes. The ETMS receives the lightning data from the ERL every five minutes.

2.7 Traffic Management Data

The ETMS uses traffic management data to update projected times for controlled flights. When certain conditions are satisfied, the ETMS issues Control Time (CT) messages. The traffic management specialists at the ATCSCC can use the *ASD* or the *TM Shell* to view various types of traffic management reports.

Section 3
Geographical Data Processing

The ETMS derives the maps database and grid database primarily from geographical data files received every 56 days from three sources.

(1) Adaptation Controlled Environmental Systems (ACES) of the 20 ARTCCs

(2) National Oceanic Service (NOS) of the National Oceanographic and Atmospheric Administration (NOAA)

(3) National Flight Data Center (NFDC) of the FAA

These files are received on magnetic tapes and read onto the ETMS computers at the central site (Volpe Center) where the ETMS extracts and/or converts the required data. The ETMS combines the ACES, NOS, and NFDC data with manually maintained data and data extracted from the OAG airline schedule file to produce the databases and map overlays used by the ETMS. The ETMS displays geographical data as graphic overlays on the *ASD*. The ETMS also uses the geographical data to interpret flight paths to support the generation of the aircraft situation data and monitor/alert data. The remainder of this section describes the source geographical data and the processing performed on the geographical data.

3.1 Source Geographical Data

3.1.1 ACES Data

The following data is received from ACES:

(1) Fixes — reporting points in the U.S. including aliases and holding fixes

(2) Landing facilities — airports in the U.S.

(3) Airspaces — sectors in the U.S.

(4) Airways — U.S. airways

(5) Metering data — U.S. metered airports

3.1.2 NOS Data

The following data is received from the NOS:

(1) High altitude airways — the high altitude or *jet* airways for the continental United States. Each airway is described by the airway number followed by the

fixes composing the airway. The fixes are of several types: NAVAIDs, reporting points (adapted fixes), ARTCC boundary crossings, airway intersections, and waypoints (unnamed fixes). Each fix is specified by the latitude and longitude pair, the type of fix, and the facility name where appropriate.

(2) Low altitude airways — the low altitude or Victor airways for the continental United States. The file format is the same as for the high altitude airways.

(3) Alaskan high altitude airways — the high altitude or jet airways for Alaska. The file format is the same as for the high altitude airways.

(4) Alaskan low altitude airways — the low altitude or Victor airways for Alaska. The file format is the same as for the high altitude airways.

(5) Bahamian airways — the airways for the Bahamas. The file format is the same as for the high altitude airways.

(6) Hawaiian low altitude airways — the low altitude or Victor airways for Hawaii. The file format is the same as for the high altitude airways.

(7) Oceanic airways — selected off-shore airways. The data includes jet airways, color coded airways, and airways that are simply numbered. The file format is the same as for the high altitude airways.

(8) Puerto Rican airways — the airways for Puerto Rico. The file format is the same as for the continental high altitude airways.

NOTE: The ETMS does not use the sector boundaries provided by NOS (see Section 3.2.1).

(9) SIDs — the Standard Instrument Departure routes. Each SID is preceded by its full name followed by a name in parentheses. The parenthesized name is the SID name and fix name as they would appear in a flight plan (e.g., **ALCOS2.ALCOS**). The following lines then specify the route of the *root segment* of the SID in the same format as the other route files. The root segment (the first segment of a SID) is followed by any *transitions* for the SID. (Transitions are the possible choices for a second, continuing segment of a SID.) Again, each transition is preceded by the full name optionally followed by the SID name and transition fix name, in parentheses, as they would appear in a flight plan (e.g., **ALCOS2.CKW**). The lines following the transition title then specify the route of the transition in the usual route format. There is a variable number of transitions associated with a given SID. Generally, the root segment ends at the fix, which is the start of the transition.

NOTE: The SID definitions do not include any altitude or speed profile data.

(10) STARs — the Standard Terminal Arrival Routes. The file format is the same as for the SIDs except that the direction is reversed; i.e., the transition ends at the fix which is the start of the root segment.

NOTE: The STAR definitions do not include any altitude or speed profile data.

(11) Military training routes — military training routes identified by names such as IR-2, IR-11, etc. The format is somewhat similar to high altitude airways, except all points are fix-radial-distances, and altitude information is included.

3.1.3 NFDC Data

The following data files are received from the NFDC:

(1) Landing facilities — an entry for every landing facility in the continental United States. Each entry is an exhaustive listing of all known data for a facility, including location, type (airport, seaport, heliport), type of traffic, hours of operation, light and marker types and locations, runway layouts, number of operations last year, etc.

(2) Navigation aids — a description of each FAA defined NAVAID, including some foreign and all domestic NAVAIDs. File format includes name, type, location, magnetic variation, and usage.

(3) Airspace fixes — a description of each reporting point in the continental United States. File format includes name, type, location, usage, and fix-radials used to identify the fix.

(4) SUA (Special Use Airspace, formerly Military Areas) Boundaries — the boundaries for the SUAs grouped by the five types: alert areas, Military Operation Areas (MOAs), prohibited areas, restricted areas, and warning areas. Each SUA definition includes the name, the hours of operation, and the altitude limits, followed by a set of boundary points in latitude and longitude.

3.1.4 Manually Maintained Data

Sector types and sector capacities are generated by traffic management specialists at the ARTCCs and maintained by personnel at the ETMS central site. Also, some location identifiers that appear in flight plans are not defined in the NOS and NFDC data files. Some of the missing location identifiers have been defined by talking to FAA and military personnel; some are defined in published reference materials. All such data is maintained manually in ASCII files, which are grouped by data type and source and are defined as follows:

(1) Sector types, capacities, and ASD altitude slices — provided by the associated ARTCCs

(2) ARTCC boundaries — generated from ACES Fix Posting Area (FPA) boundaries; maintained and provided by specialists at the ARTCCs. ARTCC boundaries are defined by the ARTCC name followed by a series of latitudes and longitudes which, when connected, form the boundary of the ARTCC. No altitude limits are defined.

(3) TRACON boundaries — generated from ACES FPA boundaries; provided by the traffic management branch of the FAA. TRACON boundaries are defined

by the TRACON name followed by a series of latitudes and longitudes which, when connected, form the boundary of the TRACON. No altitude limits are defined.

(4) Pacing airports — appear on the *ASD* screen in response to the **pacing airports** command. Each pacing airport is defined by name only.

(5) Arrival fixes — selected definitions provided by the traffic management branch of the FAA. Arrival fixes are defined by name, by the airport they are associated with, and by the type of aircraft they are used for. Aircraft types are defined as jet, prop, or both.

(6) Departure fixes — determined by the ETMS central site. Departure fixes are defined by name.

(7) Military fixes — defined for local use by the military. A fix is defined by its name and its position in latitude and longitude.

(8) Oceanic Fixes — defined for oceanic use. A fix is defined by its name and its position in latitude and longitude.

(9) ICAO location identifiers — international airport codes defined by the International Civil Aviation Organization (ICAO). An airport is defined by its name and its position in latitude and longitude.

(10) Atlanta gates – the names of arrival and departure gates used by the Atlanta ARTCC. A gate is defined by its name and its position in latitude and longitude.

(11) Indianapolis gates – the names of arrival and departure gates used by the Indianapolis ARTCC. A gate is defined by its name and its position in latitude and longitude.

(12) Canadian location identifiers – numerous Canadian airports, adapted fixes, and NAVAIDs. Each is defined by its name and its position in latitude and longitude.

(13) Foreign location identifiers – numerous foreign airports, adapted fixes, and NAVAIDs. Each is defined by its name and its position in latitude and longitude.

(14) Flight information regions (FIRs) — named regions of airspace. A FIR is defined by its name and an approximate position in latitude and longitude.

(15) MOA guess data — Military Operations Areas (MOAs) that appear to be referenced in flight plans. *Guesses* are made by correlating the flight plan name with the names that appear in the NFDC SUA file. Each guess MOA is defined by its name and its position in latitude and longitude.

(16) Fix radials — routes observed in flight plans. A fix radial is defined by the fix name and radial, the fix position in latitude and longitude, and a position calculated to be 350 miles from the fix along the radial.

(17) Control airways — intersect control regions. Specified in same format as the high altitude airways.

(18) Airway aliases — a list of high altitude airways shared by Canada and the U.S. and referred to by different names. Each alias is defined by the FAA and Canadian names for the airway.

(19) World geographical boundaries — a manually digitized set of geographical boundaries that were obtained in house at the Volpe Center. They are sequences of connected latitude and longitude points.

3.2 Geographical Data Processing

The specific manner in which the ETMS processes the source data to produce each of the outputs is described in Sections 3.2.1 through 3.2.6. Most of the processing is straightforward and consists of extracting the needed information from the source files. The following sections describe the data that is saved for each data type and any exceptional processing that is performed or assumptions that are made.

The ETMS uses the geographical data mainly to generate two products. The maps database (graphics files) is generated for displaying data on the *ASD* (see Section 10 — *Displaying Data*). The grid database is generated for use in interpreting field 10s (filed flight paths; see Section 6 — *Field 10 Processing*) and for determining the *flight events* for modeling flights (see Section 7 — *Flight Modeling*).

3.2.1 Sector, ARTCC, and TRACON Boundaries

The ETMS uses the sector and ARTCC boundary data to display maps on the *ASD* and to model sector traffic demands. There are three main issues regarding the processing of the sector data: which sector and ARTCC definition data to use, how to resolve non-unique sector names, and how to handle inconsistent boundary data.

The ETMS uses ARTCC boundary data manually generated by traffic management. These boundaries are originated by personnel at the ARTCC TMUs. The boundaries are coordinated and verified by personnel at the ETMS central site, who also resolve discrepancies when necessary. The manually generated ARTCC boundaries consist of a single, two-dimensional boundary for each ARTCC name. The TRACON and ARTCC boundaries are defined as independent of altitude considerations. Sector boundaries are provided by ACES data and are complex shapes that may have more than one top and bottom altitude. Sector type and capacities are supplied by the associated ARTCC. Each sector is made up of smaller polygons with one top and bottom altitude.

The ETMS assumes that the boundaries are essentially consistent. Inconsistent boundaries can result in gaps (areas of airspace which do not belong to any sector) and overlaps (areas of airspace which belong to two or more sectors). If a gap exists, it is considered to be open air space (not within a sector or in control); that is, an aircraft in a gap is not part of any sector demand for that time. Overlaps and gaps are reported to the ARTCC and resolved by ETMS central site personnel at the discretion of the ARTCC.

3.2.2 Airways

The ETMS uses airways data to display maps on the *ASD* and to determine the flight paths as specified in the field 10s of flight plans. The data extracted from the ACES airways source files consists of the airway names, the sequence of points of latitude and longitude that constitute the airway, and the names of any NAVAIDs or adapted fixes used to define the airway. Two issues are elaborated upon here: naming the airways and using the airway intersection specification data. ACES supplies pre-named Jet routes, Victor routes, fix radials, and miscellaneous other routes. The processing of the NAVAID and adapted fix data contained in the airways file is described in Section 3.2.4 — *Fixes*.

For NOS data, airways are named automatically by the data conversion software. The text names in the data files are used to produce the names as they appear in flight plans. For example, Jet Airway No. 94 becomes **J94**, Amber Route No. 12 becomes **A12**, etc. This technique applies to the jet, Victor, and color coded airways for the continental United States and for the oceanic airways. Bahamian airways are in a separate file and are simply numbered; for these, the ETMS creates names prefixed by **BR**. Similarly, the Puerto Rican airway names are created by prefixing the airway number with **RTE**, and Atlantic airways by prefixing with **AR**.

NOS and/or NFDC supply Alaskan, Hawaiian, Puerto Rican, and Atlantic airways. (The ACES does not.) Alaskan and Hawaiian airways are identified as jet with a J prefix or Victor with a V prefix. However, the Victor numbers are not always unique with respect to the airways over the continental United States. Any Alaskan or Hawaiian airways that have duplicate names with the continental airways are handled by the ETMS; the route processor checks and corrects for such occurrences.

For NOS data, the airways definitions include points defined as airway intersections. For example, the **J66** definition includes the point where **J66** crosses **J151**. During the course of the ETMS development it was determined that the intersections defined in the file are a small subset of all the actual airway intersections. The processing therefore does not rely on the defined intersections, but determines the airway intersections dynamically during the field 10 processing (see Section 6 — *Field 10 Processing*).

3.2.3 SIDs and STARs

The ETMS uses the SID and STAR data from NOS to determine the flight paths as specified in the field 10s of flight plans. The SIDs and STARs are special cases of airways that present unique problems. It is assumed that an end point of the transition for a SID or STAR coincides with an end point of the root segment of the SID or STAR; if not, the SID or STAR is discarded. Otherwise the SID or STAR is stored, as given, in the flight path processing database with the given names. During field 10 processing when a SID or STAR is found in a flight plan, the name of the SID or STAR and the name of the adjacent fix are used to look up the appropriate portion of the SID or STAR to use. The information that is saved for each portion of a SID or STAR is the same as for the other airways.

3.2.4 Fixes

The ETMS uses the fix data to display maps on the *ASD* and to determine the flight paths as specified in the field 10s of flight plans. (A fix, in the ETMS context, is defined as any named point.) For the purposes of the field 10 processing, it is desired to build as complete a database of named fixes as possible. The information saved for each fix is the primary name, secondary names (aliases), type of fix (NAVAID or adapted), location in latitude and longitude, and, for NAVAIDs, the magnetic variation. The issues regarding the fix processing are the resolution of conflicting definitions, the aliasing of fix names, the processing of arrival fixes, and the manual definition of fixes.

There are many input files containing fix definitions, and a fix may be defined in more than one file. When the fix processing encounters a duplicate fix definition (that is, two fixes with exactly the same name), it checks the new position against the previous definition. If the new position is 20 miles or more from the old position, the processing assumes that there are two fixes of the same name and enters both into the ETMS databases. (The proper fix to use is determined in the context of the field 10 processing as described in Section 6.) If the new position is within 20 miles of the old, the processing assumes that they are the same fix. The processing selects one of the positions based on a pre-defined order of precedence. Positions from the ACES Airways files are given highest priority; this helps ensure that the ETMS fix and airway databases are consistent. Positions from the ACES NAVAID or airspace fix files are given second-highest priority. Other conflicts are resolved manually. All inconsistent fix definitions are written to a file for manual checking. Discrepancies are researched and resolved, and the file containing the error is corrected.

The ETMS creates fix name aliases when two fixes of the same type (e.g., two NAVAIDs) are defined to be at the same location (within a tolerance of .01 miles). This aliasing mainly handles two cases: NAVAIDs which have an FAA and ICAO identifier, and adapted fixes that refer to the same point by different names. In the former case, the FAA designator is made the primary name and the ICAO identifier the secondary name. In the case of adapted fixes, the assignment of primary and secondary names is random.

The ETMS processes arrival fixes differently from other fixes. The arrival fix definition files received from traffic management associate the arrival fixes with the airports for which they are used. The ETMS maintains this association in its databases in order to support the field 10 processing as described in Section 6. Arrivals fixes are associated only with specific airports. They are also distinguished as a jet arrival fix, prop arrival fix, or both.

The extensive manual definition of fixes is dictated by necessity. During the development and testing of the field 10 processing, it became apparent that many of the fixes referred to in the flight plans were not defined in the provided data files. Therefore, a cumbersome but necessary procedure was developed whereby the field 10 processing programs record every occurrence of a fix which is not found in the ETMS databases. The records are periodically analyzed to determine which unknown fixes appear with a relatively high frequency. The missing fixes are then manually researched through a variety of sources including publications from the FAA, NOS, Department of Defense (DOD), ICAO, and Transport Canada, as well as direct contact with air traffic personnel. Fixes that are defined in this manner are maintained in a set of ASCII files according to the type of fix and source of data (see Section 3.1.4 — *Manually Maintained Data*).

3.2.5 Airports

The ETMS uses the airport data to display maps on the *ASD* and to determine the flight paths as specified in the field 10s of flight plans. The ETMS airports database is drawn mainly from the ACES and NFDC landing facilities source files and includes all facilities (approximately 21,000) which accommodate conventional aircraft. The airports database is further enhanced by the addition of international airports from manually maintained files. Each airport is defined by its name and its position in latitude and longitude. The issues regarding airport processing are the resolution of inconsistent definitions and the aliasing of airport names.

The ETMS resolves inconsistent airport definitions (i.e., two airports with exactly the same names and different locations) similarly to fixes (see Section 3.2.4 — *Fixes*) except that duplicate airport definitions are not allowed. Definitions from the FAA landing facilities file are always used over other definitions. Conflicts between definitions in the manually maintained airports files are resolved manually (see Section 4 — *Schedule Data Processing*).

The aliasing of airport names is more involved than the aliasing of fix names described in Section 3.2.4. Airport names are aliased to equate FAA (three-letter) airport designators with ICAO (four-letter) airport designators and to equate incorrectly used ICAO designators with correctly used ICAO designators.

The ETMS correlates three-letter designators with ICAO four-letter designators using knowledge of how the ICAO designators are created. ICAO designators for countries such as the U.S. or Canada are created by adding a region code (K or C, respectively) to the locally used three-letter code. Therefore, when the ETMS is processing a three-letter code, it checks whether the same code is already defined with one of the country codes added. For example, when processing JFK, the ETMS checks whether KJFK (or CJFK, MJFK, etc.) is already defined. Conversely, when a four-letter code is processed, the ETMS checks whether the three-letter version exists (e.g., when processing KJFK it looks for JFK). In either case, if the name is found and the location definitions are the same (within a tolerance of .75 degree), an alias is created. For airports in the U.S., the three-letter code is the primary name; for foreign airports, the four-letter code is the primary name. U.S. airports are compared to all other U.S. airports and use a .005 degree tolerance for aliases.

In other parts of the world, the ICAO region codes are used for regions which include many countries. In these cases, the second letter is a country code, and the last two letters are an abbreviation of the locally used, three-letter code. Therefore, the aliases cannot be found by a simple character string match. When processing an airport code outside of the U.S., the ETMS looks for any airport definition that is at the same location (with a tolerance of .03 degree). If it finds two airports defined at the same location, one having a name with four letters and the other having a name with three letters, then it creates an alias (i.e., three-letter codes are never aliased). The four-letter name is used as the primary name.

After processing the airport definitions, the ETMS checks airport codes found on the OAG for existence in the map databases. Unknown airport codes are flagged, researched, and defined in the manually maintained airports file.

3.2.6 Special Use Airspace

The ETMS uses the Special Use Airspace (SUA) data to display maps on the *ASD* and to determine the flight paths as specified in the field 10s of flight plans. Each SUA is defined by its name and a sequence of boundary points in latitude and longitude. The only significant issue regarding SUAs is the manner by which they are used in the field 10 processing.

A field 10 is defined by the FAA as consisting of a sequence of fixes and routes. Fixes are intended to correspond to specific points in space. During the development of the field 10 processing, it was observed that SUAs are used as fixes in field 10s. To expedite the implementation of the field 10 processing, the SUAs are built into the ETMS databases to look like fixes (although they are still displayed by the *ASD* as boundaries). Therefore, the *SUA-fix name* refers to the SUA in the field 10s, and *SUA-fix location* is defined as the approximate center point of the SUA. The SUA-fix names are determined as follows.

When displayed as a map overlay, an SUA name is shown as it appears in the source data file. However, when it appears in a field 10, the SUA name is condensed. For example, a restricted area may be named in the source file as **.R-131** but it will appear in a field 10 as **R131**. Furthermore, it has been observed that SUAs which are defined as several subsections may be referred to as a whole in a field 10. For example, areas defined only as **R131A** and **R131B** may be referred to as **R131**. Therefore, all SUAs whose names are suffixed by letters are also built into the field 10 processing database using the name without the suffix. The format of the restricted, alert, prohibited, and warning area names are the same and can be built into the field 10 processing database automatically.

MOA names are textual in format (e.g., **MOA-TOMBSTONE, MOA-OWYHEE**) and cannot be processed automatically. MOAs are given field 10 names only in selected cases when they resolve a case of a *fix-not-found* in a field 10. When the *fixes not found* in field 10s are being researched (see Section 3.2.4 — *Fixes*), they occasionally refer to an MOA using an abbreviated name. These MOAs are manually maintained in an MOA guess file (see Section 3.1.4 — *Manually Maintained Data*). Each is defined by the field 10 name and the approximate center point of the MOA.

Section 4

Schedule Data Processing

The airline schedule data file arrives weekly by communication link from the Official Airline Guide (OAG). The airline schedule data contains the flight schedules for one month. The ETMS at the central site reads the OAG file and processes the airline schedule data with historical flight path data to produce a new ETMS Schedule Database (SDB). The ETMS uses the SDB to provide schedule information in request reports to the traffic management specialist on a request/reply basis and to provide scheduled flight data for the monitor/alert traffic demand predictions.

The ARTCCs can use a Batch Flight Plan Schedule Database (BFPSD) process to add one or more scheduled flights that are not in the OAG to the ETMS. Every week the Volpe Center retrieves FPSD batch files, processes the data, and loads it into the SDB. As a result, when a traffic management specialist requests arrival or departure data for an airport, the valid flights entered through the BFPSD process are included in the list. This process improves the accuracy of the ETMS data. Described below is source OAG data and the OAG data processing.

4.1 Source Schedule Data

The files received every week from the OAG contain the following data for each flight entry:

(1) Departure country code

(2) Departure airport

(3) Departure time (Coordinated Universal Time, UTC)

(4) Arrival country code

(5) Arrival airport

(6) Arrival time (UTC)

(7) Flag code

(a) **0** — any carrier; departing and arriving outside the U.S.

(b) **1** — domestic carrier; departing and arriving in the U.S.

(c) **2** — domestic carrier; departing or arriving but not both in U.S.

(d) **3** — international carrier; departing and arriving in the U.S.

(e) **4** — international carrier; departing or arriving but not both in the U.S.

(8) Aircraft type code

(9) Airline code

(10) Flight number

(11) Days of service — Sunday through Saturday, 1 if scheduled, 0 if not

(12) Taxi or intrastate flight flag

(13) Effective date — date flight begins within period of this file

(14) Discontinue date — date flight ends within period of this file

4.2 Schedule Data Processing

The processing of the schedule data is generally straightforward. The schedule data is extracted from the OAG file, and a schedule database is built to support the ETMS processing. The ETMS processing requires time access and airport/ARTCC access to schedule data. Airport/ARTCC access is required to respond to traffic management specialist requests for arrival and/or departure data for an airport and/or an ARTCC. Time access is required to find the scheduled flights within the monitor/alert prediction time range. The ETMS schedule database is therefore structured to provide fast look-up of schedule data by both airport and time using index files. All times are stored in UTC; all dates are stored in Julian date (numbers of days since a reference point). All pertinent data provided on the OAG data files is stored in the ETMS schedule database.

There are several significant aspects to the schedule data processing which are not completely straightforward: the handling of airport, ARTCC, and airline codes; the dual-designated carrier processing; the assignment of flight plan data to scheduled flights; and the processing of the schedule database update commands. These issues are described in the following sections.

4.2.1 Airport and ARTCC Codes

The OAG data includes the arrival and departure airport codes for the scheduled flights. During the schedule data processing, the ETMS generates a list of all airport codes referenced on the OAG. During the geographical data processing, the ETMS checks that all the OAG airports are defined in the map databases. Missing airport codes are resolved through manual intervention.

The ETMS geographical data processing produces three products used during schedule data processing:

(1) Airports associated with ARTCCs — used to generate flight lists in response to requests (e.g., list flights departing ZDC for ZNY)

(2) ARTCC for each airport — used to fill departure ARTCC and arrival ARTCC fields in flight lists

(3) Airport aliases — used to correlate different names for the same airport (e.g., JFK and KJFK) and to define primary airport names

During the schedule database building, the ETMS looks up the OAG provided airport names in the airport alias file to find the primary airport names. The ETMS then adds the flight to the schedule database, using the primary arrival and departure airport names. Similarly, when the specialist makes a flight list request for an airport, the ETMS finds the primary names and uses them to generate the response. In this manner, the ETMS ensures, for example, that all flights bound for JFK or KJFK will appear in an arrival list for JFK or KJFK.

4.2.2 Airline Codes

The OAG file contains two types of airline codes: three-letter FAA airline codes and two-letter OAG airline codes. The ETMS replaces the two-letter OAG airline codes with valid FAA three-letter airline codes for airlines that are listed in a manually updated file (see Section 4.2.3). The ETMS ignores all schedule information for airlines that have no valid FAA airline code.

4.2.3 Dual Designated Carriers

Flights may be listed in the OAG under one or more airlines but are actually flown by another (typically a commuter carrier). Such flights are known as *subcarriers* or *dual designated carriers*. As the OAG is used for marketing, there is normally no problem listing a flight under more than one airline or under a carrier different from the one that operates the flight, as long as the reservation system knows that there is really only one flight. The dual designated carriers are a problem for the ETMS, because it may look like more than one flight is scheduled to fly, or it may be difficult to correlate a flight plan filed by a subcarrier with a scheduled flight listed by a major carrier. Therefore, a file is manually maintained which contains dual designated carrier information. The ETMS uses this file to translate the OAG flight IDs into the flight IDs that should appear in the flight plans. The file contents are as follows:

(1) Carrier code substitutions — for each set of substitutions, the file contains

(a) carrier code under which flights are listed in the OAG

(b) range of flight numbers to be substituted

(c) carrier code to be substituted for the OAG code

(d) offset to be applied to the flight numbers

(e) flag for aircraft types to which the offset should not be applied

(2) Flight ID adjustments — contains flights which should have a letter suffixed to the flight ID (e.g., PAA79 may appear as PAA79A). Each is specified by the following items:

  (a) original carrier code and flight number

  (b) departure airport

  (c) arrival airport

  (d) letter to be appended to the flight ID

The ETMS applies the carrier code substitutions in a straightforward manner. As each flight is processed, it is checked against the specified criteria. If the criteria apply, the adjustment is made. For example, a carrier code substitution might be specified as follows: for **AAL** flights **1000-1499** substitute **NBE** and apply an offset of **1000** for all aircraft types except **jets**. If the flight **AAL1234** is encountered and it is serviced by a **DC3**, the flight number would be changed to **NBE234**. All schedule information for that flight would be stored under **NBE234**.

Flight ID adjustments are also straightforward to apply. As each flight is processed, it is checked against the specified criteria. If the departure and arrival airports match, the specified letter is appended to the flight ID. If the flight ID or the departure or arrival airports do not match, the adjustment is not applied. At present, this file is empty, because only Pan American Airways employed these flight ID adjustments.

4.2.4 Flight Plan Data

The ETMS requires flight paths, ground speeds, and altitudes for scheduled flights to estimate the impact that the flight will have on the NAS long before a flight plan is received. For this purpose, the ETMS maintains a database of flight paths, cruising speeds, and cruising altitudes observed in recent flight plans. As the ETMS adds a scheduled flight into the live database, it uses the most commonly filed flight path, speed, and altitude for the flight's city pair, airline, and/or aircraft type. Appendix A contains a full description of how the ETMS determines flight plan data for scheduled flights.

It is possible that a flight in the OAG data will not correspond to any recently received flight plan (i.e., for a new flight). When this occurs, the ETMS uses a direct flight path between the departure airport and arrival airport for the flight. The ETMS constructs the flight path to look like a field 10. For example, if the airport pair **JFK** to **LAX** is not found in any flight plans, the flight path **JFK..LAX** is used. The ETMS also assigns an average cruising speed and cruising altitude for the scheduled aircraft type.

4.2.5 Update Commands

The traffic management specialist at the ATCSCC can use four types of transactions to update the schedule database: inhibit flight (**INHB**), activate flight (**ACTV**), add/edit flight (**FPSD**), and cancel flight (**CXSD**). These commands are entered through the *ASD*, and they are fully described in the ASD user documentation. The specialist at the ATCSCC can allow or restrict access of these commands to the field sites. The following sections summarize what each command does and how the ETMS processes it. One of the main processing issues is the effect of a command when a schedule database update is performed.

(1) Inhibit flight — The **INHB** command causes a flight to be suppressed with respect to any ETMS data access. An inhibited flight will not appear in a request report nor will it appear in a monitor/alert prediction. An inhibit is carried over from one ETMS schedule database to the next; that is, once a flight is inhibited it will remain inhibited for that inhibit period until it is explicitly activated for the same period or until it no longer appears in the OAG data. If an inhibited flight disappears and then reappears in the OAG data updates, it will no longer be inhibited.

(2) Activate flight — The **ACTV** command reverses the effect of an inhibit. Once a flight is activated for all inhibit periods, it is indistinguishable from flights that had never been inhibited. An activate has no carry-over effect on an ETMS schedule database update.

(3) Add/edit flight — The **FPSD** command adds a new flight to the ETMS schedule database or edits an existing flight in the schedule database. At present, there is no way to delete an added flight from the ETMS schedule database. However, that is not a problem, since the added flight will not be carried over from one ETMS schedule database to the next; and, if desired, it may be inhibited indefinitely, to cause it to "disappear" from the current database.

(4) Cancel flight — The **CXSD** command causes a flight to be suppressed for the current day (from 12 hours before to 12 hours after the time of the CXSD command). The flight will not appear in a request report or in the monitor/alert predictions for the current day. The flight will reappear in the data on the next day. The **CXSD** command has the same effect as an inhibit for one day followed by an automatic activation. The effect of a cancellation is carried over from one ETMS schedule database to the next; that is, if a flight happens to be cancelled when an OAG data update is processed, it will remain cancelled (until the next day).

Section 5

Flight Database Maintenance

The ETMS maintains a Flight Database containing all flight information for the twelve hours in the past and 12 hours in the future; this database is sometimes referred to as the *live* flight database. The ETMS derives flight information from messages generated by several sources: the airline schedule database, the National Airspace System (NAS), the Dynamic Oceanic Tracking System (DOTS), the Aircraft Traffic Control System Command Center (ATCSCC), the Offshore Computer System (OCS), the En route Automated Radar Tracking System (EARTS), the Southern California TRACON (SCT), and the airlines. The airline schedule database processing generates messages to add scheduled flights to the live flight database and to cancel scheduled flights from the live flight database. The NAS host computers generate messages based on air traffic control information; these messages include flight plans, flight plan amendments, cancellations, departures, arrivals, ARTCC boundary crossings, and track position updates. DOTS generates oceanic position updates for flights in oceanic airspace. ATCSCC generates Estimated Departure Clearance Time messages (EDCTs) to implement ground delay programs. The OCS supplies Alaskan flight plans. The EARTS supplies Alaskan position reports. The SCT supplies position reports. The airlines generate flight cancellation and arrival time substitution messages for controlled flights. The ETMS extracts all available data from the incoming messages and combines it to maintain a complete and current set of data for every known flight.

The ETMS flight database supports many ETMS functions. The ETMS uses flight data to generate the aircraft situation data displays showing airborne flights. The ETMS also uses flight data to generate request reports for the past or future twelve hours. Finally, the ETMS monitor/alert processing uses the flight data to predict traffic demands and consequently to generate alerts.

The remainder of this section describes the flight database maintenance. Section 5.1 describes the contents of the incoming flight data messages. Section 5.2 summarizes the contents of the flight database. Section 5.3 describes how the data from the messages is used to update the flight database entries. Section 5.4 describes how the flight messages are matched to flight database entries. Section 5.5 describes how flight data is removed from the flight database.

Two other sections of the document describe processing closely related to the flight database maintenance. Section 6 describes how the field 10 (flight path field) of a flight message is interpreted. Section 8 describes how the flight data is used to predict the effect of a flight on the traffic demands at airports, sectors, and fixes. However, the maintenance of the flight data which results from the field 10 interpretation and the flight modeling is described in this section.

5.1 Flight Data Messages

5.1.1 Scheduled Flight Messages

The ETMS uses data from the airline schedule database to update the live flight database. The ETMS schedule database sends updates to the flight database using two types of messages: a scheduled flight plan message and a scheduled flight cancellation message.

A *scheduled flight plan* message (FS) is sent to the live flight database whenever a flight is scheduled to depart 12 hours in the future. The FS is constructed to look like a NAS flight plan message (FZ). The computer ID, which is normally generated by the host NAS computer, is instead generated by the ETMS; a letter is used as the first character of the computer ID to distinguish it from a NAS computer ID (which starts with a number). The ETMS computer ID is generated randomly to ensure that the same flights have different IDs on different days (this is important for post-analysis and will be more important if the time range of the flight database is expanded). The fields of the FS are filled with data from the ETMS schedule database (see Section 4 — *Schedule Data Processing*) with the exception of the cruising speed and altitude; these fields are filled with standard values for the given aircraft type.

A *scheduled flight cancellation* message (RS) is sent to the live flight database whenever a scheduled flight, which has already been added to the flight database using an FS, is inhibited or cancelled. The RS looks like a NAS cancellation message (RZ). The computer ID generated for the FS is saved and used on the corresponding RS to ensure that the correct flight is cancelled. The other fields of the RS are filled with data from the schedule database.Summaries of the scheduled flight messages follow.

5.1.1.1 Scheduled Flight Plan (FS)

Purpose: Feeds a scheduled flight into the ETMS live flight database.

Contents: Flight ID
Computer ID
Aircraft type
Cruising speed
Departure airport
Scheduled departure time
Cruising altitude
Flight path
Scheduled time en route
Julian departure date

5.1.1.2 Scheduled Flight Cancellation (RS)

Purpose: Cancels a scheduled flight previously fed into the ETMS live flight database.

Contents: Flight ID
Computer ID

Departure airport
Destination airport
Julian departure date
Scheduled departure time

5.1.2 NAS Messages

The NAS messages received by the ETMS are fully described with respect to form, generation, and content in the NAS Configuration Management Document NAS-MD-315, *Remote Outputs*. Sections 5.1.2.1 through 5.1.2.7 summarize the message descriptions in this document.

NAS messages may refer to flights that are *proposed* (not yet departed) or *active* (in the air). Some messages, such as flight plans, may refer to either active or proposed flights. A message for a proposed flight is referred to as a *proposed message*; a message received for an active flight is referred to as an *active message*. Proposed messages are distinguished by the presence of a computer ID. NAS messages are received only for flights that are flying under instrument flight rules (IFR). Summaries of the NAS messages follow.

5.1.2.1 Flight Plan (FZ)

Purpose: Transmits the intentions of a flight as filed with the NAS.

Contents: Flight ID
Computer ID (proposed flights only)
Aircraft type
Speed
Coordination fix
Coordination time
Cruising altitude
Flight path
Estimated time en route (ETE) (proposed flights only)
Estimated time of arrival (ETA) (active flights only)

Remarks: May be filed for proposed or active flights. Multiple flight plans may be filed for the same flight.

5.1.2.2 Amendment (AF)

Purpose: Amends a flight's intentions that were previously filed with the NAS.

Contents: Flight ID
Computer ID (proposed flights only)
Departure point
Destination
Which field(s) to amend
New contents of field(s)

Remarks: May be filed for proposed or active flights.

5.1.2.3 Cancellation (RZ)

Purpose: Cancels a flight plan previously filed with the NAS.

Contents: Flight ID
Computer ID (proposed flights only)
Departure point
Destination

Remarks: May be filed for proposed or active flights.

5.1.2.4 Departure (DZ)

Purpose: Signifies the activation of a proposed flight.

Contents: Flight ID
Computer ID (proposed flights only)
Aircraft type
Departure point
Activation time
Destination
ETA

Remarks: None

5.1.2.5 ARTCC Boundary Crossing (UZ)

Purpose: Transmits the current flight plan data as sent from the ARTCC from which a flight is leaving to the ARTCC into which the flight is entering.

Contents: Flight ID
Aircraft type
Speed
Boundary crossing point inbound
Calculated inbound boundary crossing time
Altitude
Flight path

Remarks: Flight path field usually specifies only the remainder of the flight's path.

5.1.2.6 Position Update (TZ)

Purpose: Transmits the current position, altitude, and speed of a flight as tracked by the NAS.

Contents: Flight ID
Computer ID
Speed
Altitude
Position

Remarks: A position update is generated for each flight at least once every 5 minutes. A single TZ may contain position updates for multiple flights.

5.1.2.7 Arrival (AZ)

Purpose: Signifies the termination of an active flight.

Contents: Flight ID
Aircraft type
Departure point
Destination
Deactivation time

Remarks: None

5.1.3 DOTS Messages — Oceanic Position Updates (TO)

The ETMS obtains position updates for oceanic flights outside of CONUS radar coverage from the DOTS. International flights are required to report their positions each time they cross ten degrees of longitude. These position updates are sent to ARINC. The ETMS reads the ARINC line and sends the data to a DOTS subsystem that produces TO messages. The TO messages are buffered and sent to the Parser. A TO is summarized as follows:

Purpose: Transmits speed, altitude, and position of oceanic flights as tracked by DOTS.

Contents: Flight ID
Speed
Time of current report
Altitude
Position
Time of next report
Altitude
Position
Time of next report (usually not given even though position is)
Altitude
Position

5.1.4 EDCT Messages

The ETMS manages airport arrival delay problems by applying the Selected Controlled Departure Time (SCDT) program. The output of SCDT is a list of controlled departure times (at other airports) which will resolve the predicted arrival problem. The controlled departure times are distributed to the airlines, the TMUs at the ARTCCs, and the ETMS through the EDCT messages. The EDCT messages are summarized as follows:

Purpose: Transmits a controlled departure time assigned by the ETMS.

Contents: Flight ID
Departure airport
Arrival airport
Original departure time
Controlled departure time (CDT)
Controlled time of arrival (CTA)
Departure ARTCC

5.1.5 Flight Substitution Messages

When the ETMS issues a ground delay program, the EDCTs for each airline's flights are sent to that airline. Individual airlines can then adjust their controlled flights in two ways: by cancelling them and by substituting one flight's arrival time for another.

NOTE: Substitutions are allowed only if they meet specific criteria set by the FAA. Substitution checking is described in Section 9, *Traffic Management Data Processing.*

The airlines generate four types of messages which are used in combination with each other to specify substitution requests (SIs). An airline uses an SI cancellation message (CNX) to cancel a flight which has received a controlled departure time, an SI substitution message (SUB) to substitute one flight's controlled arrival time for another's, an SI replacement message (RPL) to indicate a flight whose arrival time is available for use by another flight (via a SUB message), and an SI exchange message (EXC) to exchange the controlled arrival times between two flights. The ETMS uses the RPL message only for error checking; i.e., only the CNX, EXC, and SUB messages cause an actual change in the flight database. Note that the flight substitution messages are used in pairs: a CNX with a SUB, an RPL with a SUB, or an EXC with an EXC. Summaries of the flight substitution messages follow.

5.1.5.1 SI Cancellation (CNX)

Purpose: Cancels a flight which has received a controlled departure time. Used in combination with a SUB.

Contents: Flight ID
Departure airport
Original departure time
Current controlled departure time
Original arrival time

Current delay factor
Available controlled arrival time

5.1.5.2 SI Substitution (SUB)

Purpose: Substitutes one flight's controlled arrival time for another's. Used in combination with a CNX or with an RPL.

Contents: Flight ID
Departure airport
Original departure time
New controlled departure time
Original arrival time
New delay factor
New controlled arrival time
Original delay factor

5.1.5.3 SI Replacement (RPL)

Purpose: Indicates a flight whose arrival time is available for use by another flight (via a SUB message). Used in combination with a SUB.

Contents: Flight ID
Departure airport
Original departure time
Previous controlled departure time
Original arrival time
Previous delay factor
Available controlled arrival time

5.1.5.4 SI Exchange (EXC)

Purpose: Exchanges the controlled arrival times of two flights. Used in combination with another EXC.

Contents: Flight ID
Departure airport
Original departure time
New controlled departure time
Original arrival time
New delay factor
New controlled arrival time
Original delay factor

5.2 Flight Database Contents

The ETMS contains an entry for each known flight in the flight database (FDB). A flight is considered to have only one leg. For example, an aircraft may go from BOS to LAX stopping at

ORD under one flight ID. This will cause two entries to be created in the ETMS flight database. Both will have the same flight ID, but one will be going from BOS to ORD and the other from ORD to LAX. The flight database contains the following information extracted from the incoming flight data messages for each flight:

(1) Flight ID
(2) Computer ID
(3) Aircraft type
(4) Cruising altitude
(5) Reported altitude
(6) Cruising speed
(7) Reported speed
(8) Reported heading
(9) Last reported position
(10) Last position update time
(11) Filed flight path (Field 10)
(12) Departure ARTCC
(13) Departure airport
(14) Departure date
(15) Original departure time
(16) Scheduled departure time
(17) Filed departure time
(18) Controlled departure time
(19) Actual departure time
(20) Arrival ARTCC
(21) Arrival airport
(22) Scheduled arrival time
(23) Original arrival time
(24) Filed arrival time

(25) Initial arrival time

(26) Current arrival time

(27) Message history (i.e., what types of messages have been received)

(28) Current ARTCC

The flight database also contains the following data which is derived indirectly from the message data:

(1) Flight status (scheduled/filed/active/completed/cancelled)

(2) User category (commercial/military/general aviation)

(3) Aircraft type class (single piston/multi-piston/ etc.)

(4) Weight class (small/large/heavy)

The flight database also contains data that results from the interpretation of the field 10 and from the modeling of the flight (see Sections 6 and 7 respectively for descriptions of this processing). Of the flight modeling data, the *event list* is of particular significance. The event list is the internal representation of a flight used to predict its effect on airport, sector, and fix demands. The event list contains each airport event (departure and arrival), sector event (entry and exit), and fix event (crossing) for the flight. Each event is described by the airport, sector, or fix at which the event takes place; the event type (i.e., arrival, departure, etc.); the event location in latitude and longitude; the altitude and speed of the flight at the event; the event time; and the event status. The event status can be predicted (for future events) or actual (for past events). More details on the event list are presented in Section 7 — *Flight Modeling*. The flight modeling data contained in the flight database is as follows:

(1) Predicted departure time

(2) Predicted arrival time

(3) Flight event list

(4) Total distance

(5) Ground time

The flight database also contains status data that results from the processing of the EDCT and SI messages (see Section 9 — Traffic Management Data Processing for a description of this processing):

(1) Substitution status

(2) Control issued status

5.3 Flight Database Updating

This section describes how the ETMS updates the flight database when flight data messages are received. There are 15 different types of flight data messages which, in practice, can be received in almost any combination and sequence. The processing of each message is somewhat dependent on what messages have already been received. Section 5.3.1 describes the processing of messages when they are received in the normally expected combinations and sequences. Sections 5.3.2 and 5.3.3 present the processing of duplicate flight plans and flights which depart late (two situations which are not fully normal, but which are not considered errors). Finally, Section 5.3.4 presents the processing of messages that are received in an erroneous (or abnormal) fashion.

There are many references throughout Section 5.3 to the process of matching flight messages to flight entries. The flight message matching criteria are complex and are described in Section 5.4. For the purposes of understanding Section 5.3, it is sufficient to assume that the matching criteria successfully determine which entry or entries, if any, in the flight database contain data about the flight described in the message.

5.3.1 Normal Message Processing

The first step in understanding the updating of the flight database is to gain an overview of the most complete, normal life cycle of messages. In other words, how does the ETMS update the flight database in the expected case? The normal case is a flight that begins as a scheduled flight, is updated by a proposed NAS flight plan, departs, crosses ARTCC boundaries, generates track position updates, and arrives. The flight database processing for each message in the normal case is as follows:

(1) Scheduled flight plan (FS) — The ETMS creates an entry in the flight database for the flight, extracts all information from the FS, and stores it. Then the field 10 is interpreted, the flight is modeled, and the results are stored. The flight status is marked *scheduled.*

(2) Proposed flight plan (FZ) — The ETMS searches the flight database and finds either a matching flight entry with a status of *scheduled* or *controlled*, or no matching entry. If no matching entry is found, an entry is created, all data from the FZ is saved, and the status is marked *filed*. If a *scheduled* matching entry is found, the proposed departure and arrival times in the FZ message (in addition to the scheduled departure and arrival times) are saved. The cruising speed, cruising altitude, and field 10 from the FZ replace the values received in the FS. The field 10 is interpreted, the flight is modeled, and the results replace those already in the flight database. The flight status is marked *filed.* If a *controlled* matching entry is found, the data is again saved and the flight is modeled. However, the flight is modeled from the controlled departure time rather than the proposed departure time, and the status is left *controlled.*

(3) Departure (DZ) — The ETMS searches the flight database and finds a matching flight entry with a status of *filed* or *controlled*. The activation time is saved as the actual departure time. The actual departure time is used to update the flight event times, and the results are stored. The flight status is marked *active*.

(4) Position update (TZ) — The ETMS searches the flight database and finds a matching flight entry with a status of *active*. The flight's position, speed, and altitude are stored along with the time that the message was received. The TZ data is used to update the flight event times, and the results are stored. The flight status remains *active*. At present the position update is being enhanced by two additional sources of information: 1) DOTS messages or oceanic position updates (TOs) and 2) GPS messages.

The oceanic position update (TO) provides data similar to the TZ, but may be normally received for flights not yet active (the NAS does not generate DZ messages for flights departing foreign airports). The TO message processing is as follows:

(5) Oceanic Position Update (TO) — The ETMS searches the flight database and finds a matching entry with a status of *scheduled*, *filed*, or *active*. If the flight status is *scheduled* or *filed*, the flight status is changed to *active*. The flight's position, speed, and altitude are stored along with the time that the message was received. The TO data is used to update the flight event times, and the results are stored.

NOTE: TZs and TOs are related by the following flight data update precedence rules: TZ data is always used to update the flight data. TO data is used when TZ data has not been received for a specified period of time.

(6) ARTCC boundary crossing (UZ) — The ETMS searches the flight database and finds a matching flight entry with a status of *active*. The cruising speed, cruising altitude, and field 10 replace the previously stored values. The UZ data is used to update the flight events, and the results are stored. The flight status remains *active*.

(7) Arrival (AZ) — The ETMS searches the flight database and finds a matching flight entry with a status of *active*. The deactivation time is saved as the actual arrival time and used to update the flight event times. The flight status is marked *completed*.

The message types RS, RZ, and AF are used to cancel scheduled flights, cancel flights in the NAS, and amend NAS flight data, respectively. The processing of these messages, when they are received in their expected context, is as follows:

(8) Scheduled flight cancellation (RS) — The ETMS searches the flight database and finds a matching flight entry with a status of *scheduled*. The flight is marked as *cancelled*.

NOTE: A *cancelled* flight will not appear in request reports or monitor/alert predictions.

(9) Cancellation (RZ) — The ETMS searches the flight database and finds a matching flight entry with a status of *filed*, *controlled*, or *active*. If the RZ is proposed and the flight status is *filed* or *controlled*, the flight status is changed to *cancelled*. If the RZ is active and the flight status is *active*, the flight status is changed to *cancelled*.

(10) Amendment (AF) — The ETMS searches the flight database and finds a matching flight entry with a status of *filed*, *controlled*, or *active*. The data from the AF is extracted and replaces any previous data in the flight database. If the data amended by the AF includes the aircraft type, cruising speed, cruising altitude, and/or flight path, the flight's events are re-computed and stored. The flight status is not changed.

The EDCT assigns a controlled departure and arrival time to a flight. The EDCT can be received before or after a flight plan is received; however, the EDCT and FZ data are processed such that the order does not affect the contents of the flight database. Flight substitution messages can be received following an EDCT. The flight substitution messages are used in pairs: a CNX with a SUB, an RPL with a SUB, or an EXC with an EXC. The processing of the EDCT, CNX/SUB, RPL/SUB, and EXC/EXC data is as follows:

NOTE: Processing of EDCT and flight substitution messages not related to updating the flight database is described in Section 9, *Traffic Management Data Processing*.

(11) Estimated departure clearance time (EDCT) — The ETMS searches the flight database and finds a matching flight entry whose status is *scheduled* or *filed*. The controlled departure and arrival times are stored and used to update the flight event times. The flight status is marked *controlled*.

(12) SI cancellation/SI substitution (CNX/SUB) — The ETMS searches the flight database and finds the two flights specified by the CNX and SUB messages, and it makes sure that both flights are *controlled*. The ETMS changes the status of the CNX flight to *cancelled*, and it assigns the controlled arrival time to the SUB flight. The system then remodels the departure times.

(13) SI replacement/SI substitution (RPL/SUB) — The ETMS searches the flight database and finds the two flights specified by the RPL and SUB messages. The ETMS makes sure that the RPL flight was previously SUB'd, and it assigns the RPL flight arrival time to the SUB flight. The system then remodels the departure times.

(14) SI Exchanges (EXC/EXC) — The ETMS searches the flight database and finds the two flights specified by the EXC/EXC. The ETMS makes sure the departure times follow the current rules, assigns the new departure times to the flights, and then remodels the flight times.

The final, normal message processing is for active flight plans (FZs). An active FZ is a flight plan which is filed while the flight is already in the air. An active FZ is identified by the absence of a computer ID and is processed as follows:

(15) Active flight plan (FZ) — The ETMS searches the flight database for a matching flight entry. If a matching entry is found with a *scheduled* status, that entry is updated; otherwise, a new entry is created. In either case, the information from the FZ is extracted and stored, and the flight is marked *active*. The field 10 is interpreted, the flight is modeled, and the results are stored, replacing the previous results, if any.

5.3.2 Duplicate Flight Plans

It is a regular occurrence that multiple flight plans are filed for a flight before it departs. This allows a user, for example, to propose two different flight paths from which to select at the time of departure. Multiple flight plans are distinguished in the NAS host computers by different computer IDs. The NAS departure message (DZ) uses the computer ID to identify which flight plan is being activated.

There are two issues regarding duplicate flight plans. The first issue is how other data are correlated with the flight plan data. When an FZ is received for a scheduled flight (i.e., an FS was first received), the ETMS updates the existing flight entry with the data from the FZ; that is, the entry contains a complete set of schedule and filed data. When the duplicate FZ is received, the ETMS creates a second flight entry to store the duplicate FZ data; however, the second entry should contain the schedule data also (the first FZ has no more right to be associated with the FS than the second). Similarly, if an EDCT is received for a flight with duplicate flight plans, the EDCT data should be applied to both flight entries. The ETMS processes the messages such that the appropriate data is shared between duplicate flight entries.

The second issue is how the flight appears in the ETMS output data, that is, how the duplicate flight entries appear in the request reports and monitor/alert predictions. It would be misleading for a flight to appear twice in the output data. Therefore, the ETMS includes only the flight with the flight plan received last in the request reports and the monitor/alert predictions. Subsequently, when one of the flight entries is activated by a DZ, it will be the only one to appear in the output data.

Following are the ETMS processing steps for multiple flight plans:

(1) First flight plan (FZ) — The ETMS searches the flight database for a matching flight entry. If a matching entry is found with a *scheduled* status, that entry is updated; otherwise a new entry is created. In either case, the information from the FZ is extracted and stored, and the flight is marked *filed*. The field 10 is interpreted, the flight is modeled, and the results are stored, replacing the previous results, if any.

(2) Second (or subsequent) flight plan (FZ) — The ETMS searches the flight database for a matching flight entry. One (or more) matching entry is found with a *filed* status and a different computer ID. A new entry is created, the information from the FZ is extracted and stored, and the flight is marked *filed*. If the matching entry had information from an FS, the information is copied to the new entry. If the matching flight entry had control information from an EDCT and/or an SI SUB, the control data is applied to the new flight entry as well. The new field 10 is interpreted, the flight is modeled, and the results are stored. The previous matching entry is flagged so that it will not appear in request reports or monitor/alert predictions.

(3) Departure (DZ) — The ETMS searches the flight database and finds the matching flight entry with the same computer ID. The activation time is saved as the actual departure time. The actual departure time is used to update the flight event times, and the results are stored. The flight status is marked *active*. The other matching flights that have different computer IDs are flagged so that they will not appear in any request reports or monitor/alert predictions.

The remaining messages are processed as in the normal case.

5.3.3 Flights Not Activated

Another regular occurrence is that a flight described in an FS or a proposed FZ never departs or departs very late. The ETMS monitors all flight entries whose status is *scheduled*, *proposed*, and *controlled* to check whether they do, indeed, depart near their predicted departure time. If a *proposed* or *controlled* flight is more than five minutes late departing, the ETMS adds five minutes to the predicted departure time. The ETMS continues checking and delaying the flight every five minutes until the flight departs or is delayed beyond one hour after its original departure time. Once the flight has been delayed more than one hour, the ETMS flags the flight so that it does not appear in any request reports or monitor/alert predictions. If the flagged flight does eventually depart, the DZ is processed as in the normal case, and the flight will once again appear in the ETMS output data. If the flight never departs, it never reappears in the ETMS output data and is eventually removed from the flight database. A *scheduled* flight (no FZ received) from CONUS airports is removed ten minutes after its scheduled departure time. A *scheduled* flight from non-CONUS airports is removed after its scheduled arrival time.

5.3.4 Abnormal Message Processing

Occasionally a flight message is received and matched to an entry in the flight database whose status does not match the status indicated by the message. For example, a TZ or TO may be received for a flight entry which has a *filed* status, a DZ may be received for a flight entry which is *scheduled*, or an EDCT may be received for a flight entry which is already *active*. Also, a message may be received which should match an existing entry in the flight database but which does not. These situations may occur because a message was lost, a message was delayed, or a message was never generated (e.g., a TZ may be received for a pop-up flight, that is, a flight with no flight plan but still sending TZs, UZs, or an AZ).

The ETMS attempts to handle all abnormal occurrences in a graceful manner by making the most probable assumptions. The abnormal message processing is performed as follows:

(1) Scheduled flight plan (FS) — If an FS matches a flight entry with a status of *scheduled*, *filed*, *controlled*, *active*, or *completed*, the FS is considered to be in error; the error is logged, and the message is ignored.

(2) Proposed flight plan (FZ) — If a proposed FZ matches an existing entry with a status of *filed* and the computer IDs match, it is not considered to be a duplicate (see Section 5.3.2) but, rather, an amendment to the previous flight plan. The only field which can be amended is the filed departure time. The new filed departure time is stored replacing the previous value. The flight event times are updated using the new value. The flight status remains *filed*. If an FZ matches an existing entry with a status of *active*, the FZ is assumed to be an error. The error is logged, and the FZ is ignored.

(3) Active flight plan (FZ) — If an active FZ matches an existing entry with a status of *scheduled* or *controlled*, the data from the FZ is used to update the flight entry in the normal manner. If the active FZ matches an entry which is *filed*, it is treated as a duplicate flight plan (see Section 5.3.2). If the active FZ matches an entry which is *active*, it is used to update the entry as if an amendment. In all cases, the entry is left with a status of *active*.

(4) Departure (DZ) — If a DZ does not match an existing entry with the same computer ID (the expected case), the database is searched for other matching entries. If the DZ matches an entry with a status of *scheduled*, *filed*, or *controlled* and with a different computer ID, it is assumed that a flight plan message was lost. The data from the DZ is used to update the existing flight entry as it normally would be for a *filed* or *controlled* entry with a matching computer ID. If the DZ matches an entry that is *active*, the DZ is assumed to be an error; the error is logged and the message is ignored. If the DZ matches no existing entries, it is

also assumed that the flight plan message was lost. An entry is created, and the data from the DZ is stored. An airport departure event is created and stored; therefore, the flight will appear in request reports for the departure airport even though the flight path is not known.

(5) Position update (TZ, TO) — If a TZ or TO does not match an existing entry with a status of *active*, the database is searched for other matching entries. If the position update matches an entry that is *filed* or *controlled*, the position update data is stored and is used to update the flight event times. The flight status is changed to *active*. If the position update does not match any entry (there is no way to tell whether a TZ matches a scheduled entry), an entry is created, the position update data is stored, and the flight is marked *active*. In this case, there are no flight events, nor will the flight appear in request reports or monitor/alert predictions. However, the flight will appear in the aircraft situation data at its current position.

(6) ARTCC boundary crossing (UZ) — If a UZ does not match an existing entry with a status of *active*, the database is searched for other matching entries. If the UZ matches an existing entry with a status of *scheduled*, *filed*, or *controlled*, it is assumed that a DZ (and possibly an FZ) was lost. The data from the UZ is used to update the flight entry as it would be for an *active* flight, and the flight status is changed to *active*. If the UZ matches an entry that was created from only a DZ as described in (4), the event list computed from the UZ data is updated with the departure time previously stored from the DZ, and the results are stored. If the UZ does not match any existing flight entry, an entry is created, and the data from the UZ is stored. The coordination fix and time from the UZ are used to provide a reference point for computing the flight event times (no other reference time, such as a departure time or track position report, is available).

(7) Arrival (AZ) — If an AZ does not match an existing entry with a status of *active*, the database is searched for other matching entries. If the AZ matches an entry with a status of *scheduled*, *filed*, or *controlled*, it is assumed that some number of messages for the flight were lost. The data from the AZ is used to update the flight entry as it would be in the normal case, and the flight status is set to *completed*. If the AZ matches an entry that is *completed*, the AZ is assumed to be an error; the error is logged, and the message is ignored. If the AZ matches no flight entries, an entry is created, and the data from the AZ is stored. An airport arrival event is created and stored; therefore, the flight will appear in request reports for the arrival airport even though the flight path is not known. The flight is marked *completed*.

(8) Scheduled flight cancellation (RS) — If an RS does not match an existing entry with a status of *scheduled*, the RS is assumed to be an error; the error is logged, and the message is ignored. In particular, this means that if a scheduled flight is added to the database and the traffic management specialist tries to inhibit or cancel that flight, it will not be inhibited or cancelled if a flight plan has been received.

(9) Proposed cancellation (RZ) — If a proposed RZ does not match an existing entry with a status of *filed* or *controlled* and with the same computer ID, the database is searched for other matching entries. If the proposed RZ matches only an existing entry with a status of *scheduled*, it is assumed that a flight plan was lost; the flight entry is therefore marked *cancelled*. If the proposed RZ matches an entry with a status of *active*, or if it matches an entry with a status of *filed* or *controlled* but a different computer ID, the RZ is assumed to be an error; the error is logged, and the message is ignored.

(10) Active cancellation (RZ) — If an active RZ does not match an existing entry with a status of *active*, the database is searched for other matching entries. If the active RZ matches a flight that is *scheduled*, *filed*, or *controlled*, the message is ignored.

(11) Proposed amendment (AF) — If a proposed AF does not match an existing entry with a status of *filed* or *controlled*, the database is searched for other matching entries. If the proposed AF matches an existing entry with a status of *scheduled* or *active*, it is assumed that the data contained is valid. The AF data is stored (possibly replacing previous data), but the flight status is unchanged.

(12) Active amendment (AF) — If an active AF does not match an existing entry with a status of *active*, the database is searched for other matching entries. If the active AF matches a flight that is *scheduled*, *filed*, or *controlled*, it is assumed that a DZ (and possibly an FZ) was lost. The data from the AF is stored (possibly replacing previous values from a flight plan), and the flight status is changed to *active*. If an AF matches no existing entries, the message is ignored.

(13) Estimated departure clearance time (EDCT) — If an EDCT matches an existing entry with a status of *active* or *complete*, it is assumed that the EDCT is an error; the error is logged, and the message is ignored. If the EDCT matches an existing entry that is *controlled*, it is assumed that the EDCT is an update of a previous EDCT; therefore, the EDCT is processed as in the normal case. If the EDCT does not match any flight entry, the EDCT is processed as described in Section 5.3.1.

(14) SI cancellation (CNX) — If the ETMS does not find an existing flight entry with a status of *controlled*, an error is logged, and the message is ignored.

(15) SI substitution (SUB) — If the ETMS does not find an existing flight entry with a status of *controlled*, an error is logged, and the message is ignored.

The final, abnormal situation which must be processed is common to many of the situations described above in (1) through (15). When a flight entry is activated (i.e., the status is set to *active*), it is normal to assume that there are no other active entries with the same flight ID. However, if an AZ or RZ message were lost for a previous, active flight leg, the subsequent flight leg could be activated causing two *active* flight entries with the same ID. This would be a violation of the NAS processing requirements and would create havoc for the flight message matching criteria described in Section 5.4. Therefore, whenever a flight status is set to *active*, the ETMS searches the flight database for other active entries with the same flight ID. If any are found, it is assumed that the termination message (AZ or RZ) for that flight was lost; the status for that entry is therefore set to *completed*.

5.4 Flight Message Matching Criteria

The flight message matching criteria determine how the ETMS matches an incoming flight data message to an entry (or entries) in the flight database. It is important for the ETMS to find all entries in the database which refer to a given flight, not just the entry to which the message specifically relates. For example, when a DZ is received, the specific flight entry to which the DZ refers is identified by the computer ID. However, there may have been duplicate flight plans for that flight, one of which is currently being used in the monitor/alert predictions. The ETMS must find all the flight entries, regardless of computer ID, so that it can remove the other flight entry from the predictions as well as activate the flight entry referred to by the DZ. The message matching problem is made more difficult by the many exception cases that the ETMS must process (see Section 5.3.4). For example, when a DZ is received, the ETMS cannot simply look for flight entries with a *filed* status, but must look for *scheduled* entries as well. The schedule status complicates the job even further, because the flight database may contain the same flight scheduled for two or more consecutive days.

The ETMS flight message matching criteria are summarized in Table 5-1. The table identifies the fields that are compared between the message and the flight entry depending on the message type, message status, and the flight entry status. For example, the first line of Table 5-1 indicates that a proposed FZ is considered to match a flight entry with a status of *scheduled* if the flight ID, departure airport, arrival airport, and departure time all match. The table presents all combinations of message and flight entry statuses checked by the ETMS.

NOTE: The presence of matching criteria in Table 5-1 does not imply that the given combination of message status and entry status is considered normal. Some matches are made simply to log errors. See Section 5.3 for a description of the message processing.

The ETMS compares flight IDs, computer IDs, and airport names in a literal fashion; that is, they must match exactly. The ETMS compares departure or arrival times with a large tolerance (6 hours). The ETMS performs time checks in case entries exist for the same flight for more than one day. The time checks are therefore only performed for *scheduled* entries.

5.5 Flight Data Deletion

The ETMS maintains flight data for the past 12 hours on all flights that achieved *active* status (became airborne). These flights are deleted from the database when their arrival times are more than 12 hours prior to the current time. Flights that do not achieve *active* status are deleted based on the following criteria:

- Scheduled flights (no FZ received) from CONUS airports are removed ten minutes after their scheduled departure times.
- Scheduled flights from non-CONUS airports are removed after their scheduled arrival times.
- Filed flights are removed two hours after their proposed departure times.

Once a flight is deleted, its data cannot be viewed in any ETMS outputs.

The ETMS uses the best time available for performing the deletion check for a flight. If the flight was activated, the ETMS uses the arrival time from the AZ message. If the flight was never activated, the ETMS uses the last predicted departure or arrival time.

Table 5-1. Flight Message Matching Criteria

Message Type	Flight Entry Status	Flight ID	Comp. ID	Dep. Apt.	Arr. Apt.	Dep. or Arr. Apt	Dep. Time	Arr. Time
Proposed FZ	Scheduled	X		X	X		X	
	Filed	X		X				
	Controlled	X		X				
	Active	X		X				
Active FZ	Scheduled	X		X	X		X	
	Filed	X				X		
	Controlled	X		X	X			
	Active	X		X				
DZ	Scheduled	X		X	X		X	
	Filed	X		X				
	Controlled	X		X	X			
	Active	X		X				
FS	Scheduled	X		X	X		X	
	Filed	X		X	X		X	
	Active	X		X	X		X	
UZ	Scheduled	X		X	X		X	
	Filed	X				X		
	Controlled	X		X	X			
	Active	X		X	X			
TZ	Filed	X	X					
	Controlled	X	X					
	Active	X						
AZ	Scheduled	X		X	X			X
	Filed	X			X			
	Controlled	X			X			
	Active	X			X			

Table 5-1. Flight Message Matching Criteria (cont.)

Message Type	Flight Entry Status	Flight ID	Comp. ID	Dep. Apt.	Arr. Apt.	Dep. or Arr. Apt	Dep. Time	Arr. Time
	Filed	X			X			
	Controlled	X			X			
	Active	X			X			
Proposed AF	Scheduled	X		X	X		X	
	Filed	X	X					
	Controlled	X		X				
	Active	X		X	X			
Active AF	Scheduled	X		X			X	
	Filed	X		X				
	Controlled	X		X				
	Active	X		X				
RS	Scheduled	X		X	X			
	Filed	X		X	X			
	Active	X		X	X			
Proposed RZ	Scheduled	X		X	X		X	
	Filed	X	X					
	Controlled	X		X	X			
Active RZ	Active	X		X				
EDCT	Scheduled	X		X	X		X	
	Filed	X		X	X			
	Controlled	X		X	X			
	Active	X		X	X			
SI CNX	Scheduled	X		X	X		X	
	Filed	X		X	X		X	
	Controlled	X		X	X		X	
	Active	X		X	X		X	

Table 5-1. Flight Message Matching Criteria (cont.)

Message Type	Flight Entry Status	Flight ID	Comp. ID	Dep. Apt.	Arr. Apt.	Dep. or Arr. Apt	Dep. Time	Arr. Time
SI SUB	Scheduled	X		X	X		X	
	Filed	X		X	X		X	
	Controlled	X		X	X		X	
	Active	X		X	X		X	
TO	Scheduled	X					X	
	Filed	X					X	
	Controlled	X					X	
	Active	X					X	

Section 6

Field 10 Processing

One of the most significant parts of the ETMS is the processing that interprets the field 10s. A field 10 is the field of a NAS message that is used to specify an intended flight path. A field 10 specifies a flight path as a sequence of *fixes* and *routes*, which can each be specified in a great variety of ways. The ETMS parses the field 10 text and translates it into a sequence of *waypoints* defined in latitude and longitude. The ETMS uses the waypoints of the flight path to draw flight paths on the *ASD*, as described in Section 10 — *Displaying Data*, and to model the effects of the flight on airports, sectors, and fixes, as described in Section 7 — *Flight Modeling*.

The main source of data required to process the field 10s is the geographical data. The ETMS uses the geographical data to look up the positions of airports, fixes, and airways referred to by name in the field 10. The ETMS builds a special *grid database* from the geographical data which is used to determine the interrelationship of the geographical data elements. The source geographical data is described in Section 3 — *Geographical Data Processing*.

The remainder of this section describes the field 10 processing in detail. Section 6.1 describes the contents of the field 10. Section 6.2 describes the construction of the grid database. Section 6.3 then describes the interpretation of the field 10.

6.1 Field 10 — Filed Flight Path

The field 10 contents and format are completely described in the NAS Configuration Management Document NAS-MD-312, *Route Conversion and Posting*. An informal description is provided here.

A field 10 describes a flight path as a sequence of fixes and routes separated by periods; that is, *fix.route.fix.route.fix.*, etc. A field 10 must start with a fix, and may end with a fix, or a route field that contains **VFR** or **XXX**. A fix field can contain one of the following fix types:

(1) Airport

(2) Named fix (NAVAID or adapted)

(3) Fix-radial-distance

(4) Special Use Airspace (SUA) name

(5) Latitude/longitude.

(6) Flight Information Region (FIR) name

The fix types other than fix-radial-distance have been discussed as part of the geographical data in Section 3.1. A fix-radial-distance is defined by a NAVAID name, a bearing, and a distance from the NAVAID. A fix may have an airborne delay appended to it of the form *Dh+mm* where *h* is a number of hours and *mm* a number of minutes. The delay specifies airborne holding time to be taken at that fix.

A route field may contain one of the following route types:

(1) Airway

(2) Fix-radial

(3) SID

(4) STAR.

The route types other than fix-radial have also been discussed as part of the geographical data in Section 3.1. Named airways consist of jet, Victor, oceanic, Bahaman, Puerto Rican, military training, and control airways. A fix-radial consists of a NAVAID name and a bearing from the NAVAID.

In addition to the above types, a fix or route can be implied; that is, the field can be empty. The only constraint on implied fields is that two implied fields may not appear in a row (that is, an implied fix may not be followed by an implied route, nor an implied route by an implied fix), nor may an implied fix appear in the first or last field. An implied fix is defined by the intersection of the previous and following route; an implied route is defined as a straight line (actually a *great circle*) between the previous and following fixes.

Several special designators can also appear in a route field of a field 10. The processing of these designators is described in Section 6.3.4.

(1) / — a *tailoring indicator*; a field 10 containing a tailoring indicator is called a *tailored route*. A tailoring indicator in the first route field means that the flight path up to the fix following the tailoring indicator is not specified. A tailoring indicator frequently appears in the route fields for boundary crossings (UZs), amendments (AFs), and in active flight plans (FZs)

(2) **AFIL, AFILE** — indicates that the flight plan was *air filed*

(3) **DCT, DIRCT, DRCT** — indicates the flight is flying *direct*

(4) **DR** — indicates the flight is *dead reckoning*

(5) **RNAV** — indicates the flight is using *random area navigation*

(6) **RV** — indicates that a *radar vector* is being used

(7) **TLTP** — indicates to a controller that the flight plan was *too long; see teletype*

(8) **VFR, AVFR** — indicates that the flight is intending to fly *visual flight rules* for some portion of its flight path

(9) **XXX** — an *incomplete route indicator*; indicates part of the field 10 was in error

(10) **ZZZZ** — indicates that the flight is part of a *search and rescue* operation

The manner in which error checking is performed on the field 10s by the NAS has a significant impact on the ETMS field 10 processor. When a flight plan is filed with an ARTCC, it is checked completely for syntax errors (errors of form). However, semantic errors (errors in meaning) are detected only for the portion of the flight which is internal to the ARTCC. When the flight is handed over to the next ARTCC, the field 10 is checked within its boundaries, and so on. Therefore, the ETMS may receive several erroneous versions of a field 10, and the flight may be well along its flight path before an error is corrected.

6.2 Grid Database

The geographical data files provide the ETMS with the basic information necessary to look up the field 10 elements (fixes and routes) by name and discover their locations. To perform the complete field 10 processing, however, the field 10 processor must also know where routes intersect, what fixes a route goes through, etc. To perform the flight modeling, the ETMS must know where a flight crosses a fix, a sector boundary, etc. (The use of the grid database for flight modeling is described in Section 7.) To support the field 10 processing and flight modeling, the ETMS builds a *grid database* to represent all the geographical interrelationships between the data. A logical representation of the grid database is shown in Figure 6-1.

The central aspect of the grid database is a spherical grid which covers the CONUS with cells of the dimension five minutes latitude by five minutes longitude. Each grid cell is used to provide a cross-reference between the various map elements which are located in or intersect the cell. The cross-referencing is accomplished by recording, as part of each map element definition, the grid cells contained in a sector, the cells intersected by a route, or the cell in which a fix or airport resides. Conversely, for each grid cell the ETMS records which sectors the cell resides in, which routes the cell falls on, and which fixes and/or airports the cell contains. Figure 6-1 shows, for example, the shaded grid cell containing (i.e., pointing to) airport **ABC** and NAVAID **DEF**, residing in sector **ZYX01**, and falling on route **J123** (a jet airway). Conversely, the **J123** jet airway definition includes a list of the grid cells that it intersects (i.e., points to), the **ABC** airport and **DEF** NAVAID definitions indicate the grid cell in which they reside, and the **ZYX01** sector definition includes the grid cells that it contains.

The field 10 processor uses the grid database to process the map data elements as in the following example. Consider a flight plan which starts **ABC..J123..J456.GHI.**, etc. (**J456** is not shown). The field 10 processor looks up airport **ABC**, finds the grid cell it resides in, and checks that route **J123** is also in that cell. The field 10 processor thus knows exactly where the flight picks up **J123**. **J123** is then followed through its list of grid cells. At each cell, the field 10 processor checks to see if **J456** is there; once it finds **J456**, it knows exactly where to start following it. Similarly, the field 10 processor looks along **J456** for the cell containing **GHI**.

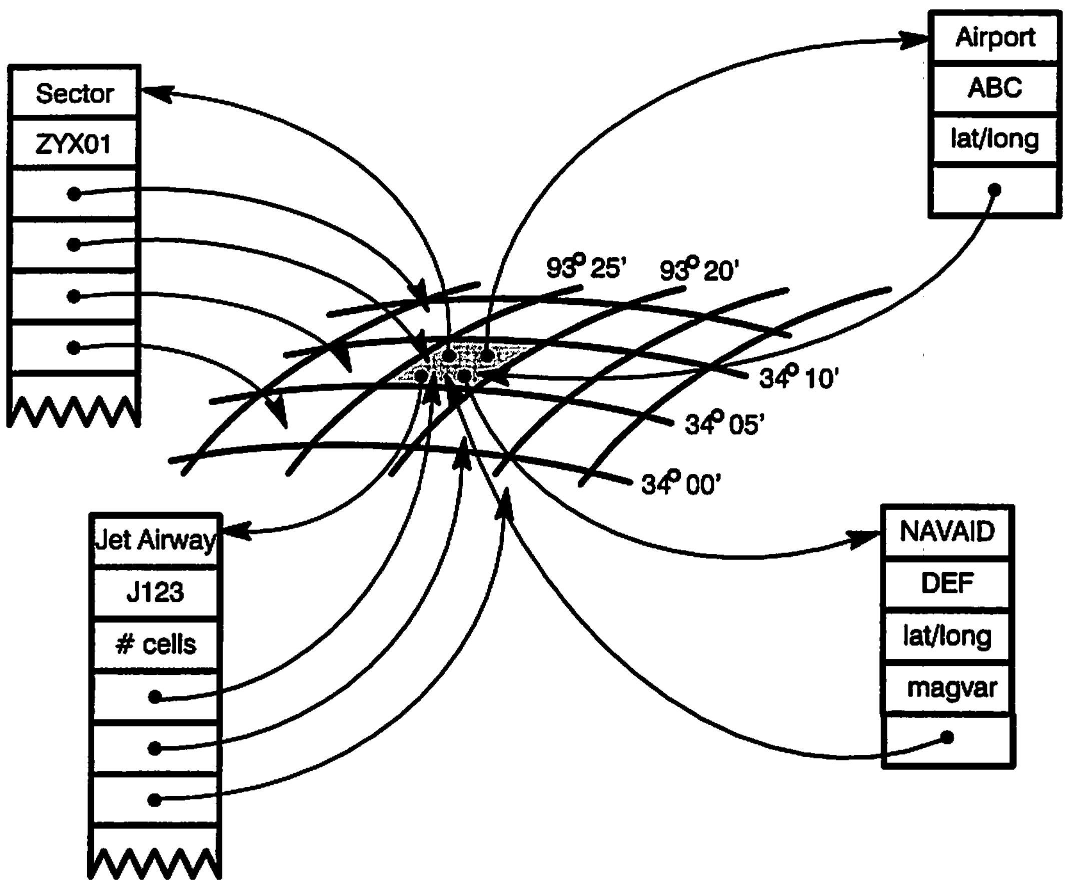

Figure 6-1. Grid Database Structure

6.3 Field 10 Processor

6.3.1 Overview

The field 10 processor scans a field 10 identifying the type of each element by its position and form. The field 10 processor knows whether a field 10 element is a fix or route by its position. The first element must be a fix, and the subsequent elements must be alternating routes and fixes. The type of route or fix is determined by examining the form of the element. Once the type is known, the field 10 processor looks up the element in the grid database, finds its location, and determines how it contributes to the flight path in the given context. As the field 10 processor parses the field 10, it builds a list of the waypoints and the grid cells through which the path passes, which is then used to generate the flight events as described in Section 7.

The field 10 processor was designed to be relatively error insensitive, since errors are frequently encountered. The errors include incorrectly filed flight plans as well as data not found in the grid database. The philosophy of the field 10 processor is to approximate the flight path as long as a reasonable amount of the field 10 can be interpreted. For example, assume a field 10 includes an airway that cannot be found in the grid database. If the route is preceded and followed by a fix which can be identified, the field 10 processor connects the fixes along a straight line and continues processing the field 10. If the fix following this route is implied, the path cannot be reasonably estimated, since a waypoint is completely missing. Therefore, the route is rejected.

The remainder of this section describes how each of the fix types is processed (Section 6.3.2), how each of the route types is processed (Section 6.3.3), how the special fields are processed (Section 6.3.4), how fixes are checked for reasonableness (Section 6.3.5), and how errors are handled (Section 6.3.6).

6.3.2 Fix Processing

The field 10 processor identifies and processes the fix types as follows:

(1) Airport — identified by position and form. An airport name must consist of three or four characters, and must be the first or last fix in a flight plan. A fix which meets these criteria is looked up as an airport. If found, the type is confirmed as an airport, and the location is made the first waypoint. If not found, the field 10 processor looks up the fix as NAVAID or adapted. If the fix is found, its type is set as NAVAID or adapted, and its location becomes the first waypoint. If the fix is not found and it is the first fix in the field 10, the field 10 is rejected. If the fix is the last fix in the field 10, the previous part of the field 10 interpretation is kept.

(2) NAVAID — identified by form; it must be three to five characters beginning with a letter. A fix which meets these criteria is looked up in the grid database as a NAVAID. If the fix is found, its type is confirmed as NAVAID and its location is checked for reasonableness (see Section 6.3.5); if multiple NAVAIDs are defined under that name, each is checked to see if it is reasonable, and the most reasonable is identified. If a reasonable fix is found, its position is added to the waypoint list. If no reasonable fix is found, the field 10 processor attempts to interpret it as a SUA. If the fix is not a SUA, the field 10 processor continues processing the field 10 as if the fix were implied; see item (8).

(3) Adapted — identified by form; it must be three to five characters beginning with a letter. A fix which meets these criteria is looked up as an adapted fix. If the fix is found, its type is confirmed as adapted and the location is checked for reasonableness (see Section 6.3.5); if multiple adapted fixes are defined under that name, each is checked for reasonableness, and the most reasonable is identified. If a reasonable fix is found, its position is added to the waypoint list. If no reasonable fix is found, the field 10 processor attempts to continue processing the field 10 as if the fix were implied; see item (8).

(4) Fix-radial-distance — identified by form; it must be three letters followed by six digits. The three letters are a NAVAID name and are looked up as such; if multiple definitions exist, each is saved. The first three digits are the radial (bearing in degrees from north) from the NAVAID. The last three digits are the distance (in

nautical miles) from the NAVAID. If NAVAIDs are found in the database, the locations are used with the radial and distance to compute the possible coordinates of the fix. The computation is made by following a great circle along the given radial for the given distance. The computed positions are then checked for reasonableness (see Section 6.3.5). If a reasonable position is found, it is added to the waypoint list. If no reasonable position is computed or if the NAVAID is not found, the field 10 processor attempts to continue processing the field 10 as if the fix were implied; see item (8).

(5) SUA — identified by form; it must begin with an **R**, **A**, **P**, or **W**, followed by two to four digits, and may be suffixed by a letter. Fixes of this form are looked up as SUAs. If the SUA is found, the type is noted and the approximate center point of the SUA is checked for reasonableness (see Section 6.3.5). (The center point is used for approximating the flight path, since the true point of entry into the SUA is not usually known.) If a reasonable SUA is found, its position is added to the waypoint list. If the SUA is not found, the field 10 processor continues processing the field 10 as if the fix were implied; see item (8).

(6) Latitude/Longitude — identified by form; it must consist of four digits (two digits for degrees and two for minutes) followed by **N** or **S**, a slash (/), and five digits (three digits for degrees and two for minutes) followed by **E** or **W**. The fix type is noted, and the position is checked for reasonableness (see Section 6.3.5). If the point is reasonable, it is added to the waypoint list.

(7) FIR (Flight Information Region) — identified by form; it must begin with **FIR**, and may be suffixed by numbers or letters. Fixes of this form are looked up as FIRs. If the FIR is found, the type is noted and the approximate center point of the FIR is checked for reasonableness (see Section 6.3.5). (The center point is used for approximating the flight path, since the true point of entry into the FIR is not usually known.) If a reasonable FIR is found, its position is added to the waypoint list. If the FIR is not found, the field 10 processor continues processing the field 10 as if the fix were implied; see item (8).

(8) Implied Fix — consists of an empty fix field. The implied fix processing is also applied to any fixes which cannot be interpreted on their own. The location of an implied fix is determined by finding the intersection of the previous and following routes. The intersection of two routes is found by traversing the grid cells belonging to the first route starting at the current position until a grid cell is found which also contains the second route. (The manner in which route intersections are determined is illustrated by the example in Section 6.2.) If the intersection can be found, the position is added to the waypoint list. If the intersection cannot be found (e.g., if one or both of the routes surrounding the fix is implied), the field 10 is rejected.

The ETMS grid database may contain multiple definitions for NAVAIDs and adapted fixes. It is therefore necessary for the field 10 processor to determine which of these is being referenced by a particular field 10. Also, it is possible that a field 10 is referencing a NAVAID or adapted fix not defined in the grid database. The field 10 processor therefore checks every NAVAID and adapted fix for reasonableness. Fixes which are not reasonable are not used. Multiple fix definitions are resolved by picking the closest reasonable fix. Fix checking is described in Section 6.3.5.

A fix of any type other than airport or implied may have an airborne delay field associated with it. The delay field specifies an airborne delay that the flight will be taking at that fix. If an airborne delay is encountered, the amount of delay is saved along with its fix association. The delay is used during the event time modeling, described in Section 7.

6.3.3 Route Processing

The field 10 processor identifies and processes the route types as follows:

(1) Airways — include the jet, Victor, and color coded airways. The airways are identified by form; they must begin with a **J**, **V**, **A**, **B**, **G**, **R**, **BR**, **RTE**, **C**, or **IR** followed by one to four digits, and may be suffixed by a letter. An airway of one of these forms is looked up in the grid database. If not found, the field 10 processor tries processing the field 10 as if the airway were implied; see item (4) below. If found, the airway definition is used to determine the flight path from the current position to the next fix or, if the next fix is implied, to the intersection of the next route. The field 10 processor adds the waypoints along the flight path to the flight's waypoint list as it traverses the path.

(2) Fix-radial — identified by form; it must begin with three letters and end with three digits. A route of this form may be processed in several ways. If the fix-radial is preceded and followed by known fixes, the fix-radial is ignored, and the field 10 is processed as if the fix-radial were an implied route.

    If the fix following the fix-radial is implied, then a route intersection must be found. To expedite the determination of fix-radial route intersections, all commonly used fix-radials are built into the grid database as if they were airways. The route definition is constructed by computing a point 350 miles out along the radial, using this point and the reference fix as the end points of a one-segment route. Therefore, for fix-radials that are built into the grid database, route intersections may be determined as for airways. (In fact, the only time the field 10 processor can determine the intersection of a fix-radial with a conventional airway is when the fix-radial has been built into the grid database.)

    If the fix following the fix-radial is implied, and the next route is also a fix-radial, and one or both of the fix-radials has not been built into the grid database, then the intersection of the fix-radials is computed using spherical trigonometry. The NAVAIDs referenced in the fix-radials are looked up in the grid database for the locations on which to base the computation. If a NAVAID cannot be found, the field 10 is rejected. If the intersection can be computed, it is used as a known fix, as just described.

(3) SID or STAR — identified by form and position. A SID can only appear in the second field (first route field) and must be followed by a named fix. A STAR can only appear in the next to last field (last route field) and must be preceded by a named fix. Either a SID or STAR must consist of a variable number of letters suffixed by a digit. A SID or STAR that meets these criteria is looked up in the database by using both the SID or STAR name and the associated fix. If the route is not found or does not meet the form criteria, the field 10 processor continues as if the SID or STAR were an implied route; see item (4) below. If found, the route can be the root segment of the SID or STAR or a transition for the SID or STAR.

If the root segment is found, the SID or STAR is used to determine the path in a way similar to the airways. An additional step must be performed to connect the front end of the SID to the origin airport or to connect the tail end of a STAR to the destination airport; these connections are made by a straight line. If a transition is found, the transition is also processed much like an airway. However, in the case of a SID, the root segment is processed first and then connected to the transition. Conversely, for a STAR, the transition is processed first and connected to the root segment.

(4) Implied Route — consists of an empty route field. The implied route processing is also applied to an airway, SID, or STAR that cannot be found in the database, and for processing fix-radials that connect two known fixes. An implied route can be processed only if the previous and next fixes are not implied. The field 10 processor simply connects the previous and next fix with a great circle, computing the list of grid cells through which the great circle passes.

6.3.4 Special Code Processing

The field 10 processor processes the special route fields as follows:

(1) — a tailoring indicator (/) can only appear as the second element (first route) in a field 10. If there is a tailoring indicator, the origin airport that appears before it is ignored. The field 10 processor processes the portion of the field 10 following the tailoring indicator. The flight is marked as **en route** at the beginning. (The **en route** indicator prevents the flight profile processing from causing the flight to ascend at the beginning of its defined path, as described in Section 7.)

(2) **AFIL** or **AFILE** — can only appear in the first field of a field 10 and is always followed by a tailoring indicator. **AFIL** or **AFILE** indicates that the flight plan was filed in the air and is ignored by the field 10 processor.

(3) **DCT**, **DIRCT**, or **DRCT** — can appear in a fix or route field, but generally appears in the first field following a tailoring indicator. **DCT**, **DIRCT**, or **DRCT** stands for *direct navigation*. If the **DCT**, **DIRCT**, or **DRCT** occupies the first field following a tailoring indicator, it is ignored. If the **DCT**, **DIRCT**, or **DRCT** occupies a meaningful route or fix field, the field is treated as an implied route or fix.

(4) **DR** — can appear in a fix or route field but generally appears in the first field following a tailoring indicator. **DR** stands for *dead reckoning*. If the DR occupies the first field following a tailoring indicator, it is ignored. If the DR occupies a meaningful route or fix field, the field is treated as an implied route or fix.

(5) **RNAV** — can appear in a fix or route field but generally appears in the first field following a tailoring indicator. **RNAV** stands for *random area navigation*. If the **RNAV** occupies the first field following a tailoring indicator, it is ignored. If the **RNAV** occupies a meaningful route or fix field, the field is treated as an implied route or fix.

(6) **RV** — can appear in a fix or route field but generally appears in the first field following a tailoring indicator. **RV** stands for *radar vector*. If the **RV** occupies the first field following a tailoring indicator, it is ignored. If the **RV** occupies a meaningful route or fix field, the field is treated as an implied route or fix.

(7) **TLTP** — can appear in a fix or route field. **TLTP** stands for *too long; see teletype*. If the **TLTP** occupies the first field following a tailoring indicator, it is ignored. If the **TLTP** occupies a meaningful route or fix field, the field is treated as an implied route or fix.

(8) **VFR** or **AVFR** — can appear in any field of a field 10, although it is generally at the beginning or the end. The effect of a **VFR** or **AVFR** in the first route field is identical to the tailoring indicator described in item (1). The effect of a **VFR** or **AVFR** in any other field is to cause the field 10 processor not to process the field 10 beyond that point. The flight path is processed up to the **VFR** or **AVFR** as usual, and marked as **en route** at that point. (The **en route** indicator prevents the flight profile processing from causing the flight to descend at the end of its defined flight path, as described in Section 7.)

(9) **XXX** — can only appear in a route field. The field 10 processor processes the field 10 as usual up to the **XXX** and ignores any portion of the field 10 after the **XXX**. The flight is marked as **en route** at the end of the processed flight path.

(10) **ZZZZ** — can appear in a fix or route field. **ZZZZ** indicates that the flight is participating in a search and rescue operation. If the **ZZZZ** occupies the first field following a tailoring indicator, it is ignored. If the **ZZZZ** occupies a meaningful route or fix field, the field is treated as an implied route or fix.

6.3.5 Fix Checking

The field 10 processor checks each fix for reasonableness as it traverses the field 10. The initial fix (i.e., the origin airport) is assumed to be correct. The next fix is checked relative to the initial fix. If it is accepted, it is used to check the next fix, and so on. A fix can appear to be unreasonable in two general ways: it can be too far away from the previously accepted fix, or it can be considered to deviate too far from the path towards the destination airport. The exact criteria for deciding whether a fix is reasonable have been determined empirically by examining many flight plans. These criteria vary depending on many factors regarding the specific flight.

The most common flight category for fix checking is the *domestic, point-to-point flight*. (Point-to-point is used to refer to a flight, such as an airline flight, which goes from its origin airport to its destination airport in a fairly direct way.) The fix checks for a domestic, point-to-point flight are as follows:

(1) If a fix is connected to the previous fix by an airway, it must be within 3500 miles of the previous fix.

(2) If a fix is connected to the previous fix by an implied route, it must be within 1750 miles of the previous fix.

(3) If a fix is further than 200 miles from the previous fix, it undergoes a bearing check: the bearing from the previous fix to the fix in question must be within 85 degrees on either side of the bearing from the previous fix to the destination airport.

(4) If a fix is within 200 miles of the previous fix, it will be accepted independent of the bearing.

(5) If the distance between the fix in question and the previous fix is more than 175 miles greater than the distance from the previous fix to the destination airport, it will be rejected.

The next category of flights for duplicate fix checking is the *international, point-to-point* flight. The main difference between international and domestic flights is the density of the fixes; in oceanic traffic, for example, fixes may be a widely spaced set of latitude/longitude pairs. International, point-to-point flights are processed the same as domestic flights except that the distance between any two fixes is allowed to be up to 3500 miles; that is, the limit check in item (2) above is changed to 3500 miles.

The final category of flight for fix checking is the *circular* flight. The simplest case of a circular flight is a flight which departs from and arrives at the same airport; bearing checks are impossible for these flights. Flights whose origin and destination airports are within 250 miles of each other and whose flight times are much greater than that necessary to fly from the origin airport to destination airport tend to have convoluted flight paths which make bearing checks unfeasible; these are also treated as circular flights. Finally, military flights (identified by flight ID and aircraft type) whose flight times are much greater than that necessary to fly from the origin airport to destination airport typically show a great degree of maneuvering and are processed as circular flights. Fix checking for circular flights ensures that each fix is within 1200 miles of the previous fix.

NOTE: Flight plans for circular flights are never received for flights outside the United States.

6.3.6 Error Handling

The field 10 processor may detect a number of errors during its processing. Some of the errors are recoverable, some are immediately fatal. The following list summarizes these errors, shows when they are fatal or recoverable, and gives the frequency of fatal errors.

NOTE: The error frequencies are approximate and will vary depending on the contents of the field 10s and the source geographical data.

(1) Bad message syntax — Field 10 has an unrecognizable format (e.g., has three dots between two fix fields).

(a) Fatal: always

(b) Recoverable: never

(c) Frequency: 0.05%

(2) Unrecognizable fix or route — A fix or route name has an unrecognizable format (e.g., fix name has seven letters).

(a) Fatal: always

(b) Recoverable: never

(c) Frequency: 0.0002%

(3) Unknown route — An airway cannot be found in the grid database.

(a) Fatal: if next to an implied or unknown fix

(b) Recoverable: recovered by connecting the two fixes, if surrounded by two known fixes

(c) Frequency: 0.15% are fatal

(4) Unknown origin airport — The origin airport cannot be found in the grid database.

(a) Fatal: always

(b) Recoverable: never

(c) Frequency: 0.10%

(5) Unknown fix or destination airport — A named fix or destination airport cannot be found in the grid database, or a fix is not reasonable.

(a) Fatal: if surrounded by one airway and one implied (or unknown) route

(b) Recoverable: recovered by finding the route intersection, if surrounded by two routes. If surrounded by two implied routes, recovered by connecting the previous and next fixes. If destination airport or last fix, recovered by truncating the route.

(c) Frequency: 0.15% are fatal

(6) Route shortened to nothing — The destination airport and all intermediate fixes are rejected by error (5).

(a) Fatal: always

(b) Recoverable: never

(c) Frequency: 0.01%

(7) No route intersection — Two routes are expected to cross (e.g., due to an implied fix) and do not.

(a) Fatal: always

(b) Recoverable: never

(c) Frequency: 0.10%

(8) Fix not on route — A fix is expected to be on a route (e.g., they are adjacent fields) and is not.

(a) Fatal: if fix is more than 10 grid cells (50 minutes of latitude or longitude) from the route and if fix on the other side of the route is implied

(b) Recoverable: if fix is within 10 grid cells, fix is connected to route by a straight line. If not, and there is a fix on the other side of the route, the route is rejected, and the fixes are connected.

(c) Frequency: 0.01% are fatal

(9) Internal error — There are several, obscure items that check the internal consistency of the processing.

(a) Fatal: always

(b) Recoverable: never

(c) Frequency: .0045%

Section 7
Flight Modeling

The ETMS flight modeling uses the waypoint list produced by the field 10 processing along with the other flight data extracted from the flight data messages to predict the impact of the flight on the airports, sectors, and fixes along its flight path. The modeling of a flight consists of three steps: determining the altitude and speed profile of the flight (i.e., what altitude and speed the flight will have at any point along its flight path); determining the flight events; and computing the event times. The flight events are airport arrivals and departures, sector entries and exits, ARTCC entries and exits, airway entries and exits, and fix crossings. Each flight event is defined by the event type, the position in latitude and longitude, the speed of the aircraft at the event, the altitude of the aircraft at the event, and the time of the event.

The flight modeling is based in part on data extracted from the flight data messages and the results are stored in the flight database; both subjects are described in Section 5 — *Flight Database Maintenance*. The flight modeling relies heavily on the *waypoint list* produced by the field 10 processor, as described in Section 6. The ETMS grid database, described in Section 6.2, plays a central role in the determination of the flight events. The ETMS uses the flight event data to support the display of aircraft situation data (e.g., show all flights going through a given sector) as described in Section 10 — *Displaying Data*, and to compute traffic demands as described in Section 8 — *Traffic Demand and Alert Processing*.

The remainder of Section 7 has three main sections. Section 7.1 describes how a flight profile is assigned to a flight. Section 7.2 describes how the flight events are generated. Section 7.3 describes how the flight event times are computed.

7.1 Flight Profile Modeling

The flight profile modeling estimates the altitude and speed of a flight as a function of its distance from the origin or destination airports. For the purposes of profile modeling, a flight can be broken down into three segments: the *ascent phase*, where the aircraft climbs from its origin airport to some altitude; the *en route phase*, where the aircraft travels at some altitude and speed; and the *descent phase*, where the aircraft descends to the destination airport. The flight profile modeling gets the cruising altitude and speed from the flight message data stored in the flight database. The ascent and descent profile data is stored in the aircraft profile database described in Section 7.1.1. Section 7.1.2 describes how the aircraft profile data and cruising altitude and speed are used to determine the profile for a specific flight.

7.1.1 Aircraft Profile Database

The aircraft profile database contains ascent and descent profiles to be applied to the various aircraft types. An ascent profile specifies the altitude and speed of a departing flight as a function of its distance from the origin airport. Similarly, a descent profile specifies the altitude and speed of an arriving flight as a function of its distance from the destination airport.

The aircraft profile database is derived from several sources. *FAA Handbook 7340* provides a complete list of FAA aircraft type designators and information associated with each one. The information extracted for each aircraft type consists of weight class (small, large, heavy), climb rate and descent rate. Publication 7340 formerly listed an item called the aircraft category. Although the aircraft category is no longer listed in Publication 7340, it is being manually maintained for new aircraft. The last item divides aircraft into nine groups: single-engine piston, multi-engine piston, single-engine turboprop, multi-engine turboprop, civilian turbojet, military fighter type turbojet, military cargo/bomber type turbojet, special performance turbojet, and helicopters.

The second source of aircraft data is from the Integrated Noise Model (INM). The INM is a set of aircraft ascent profile models that were developed for the FAA for studying noise levels at airports. The INM includes very detailed ascent profiles for 43 aircraft types which span the range of weight classes and categories. The INM contains up to seven ascent profiles for each aircraft type depending on the trip length. The profiles specify the distance, speed, altitude, and thrust for each breakpoint of the ascent up to 10,000 feet.

The ascent profiles up to 10,000 feet from the INM are used as given. These ascent profiles are then extended to the absolute ceiling for each aircraft type according to the standard fastest climb condition defined in aircraft dynamics literature.[†] To generate the altitudes and speeds as a function of the distance from the airport, a fastest climb algorithm was developed which uses the maximum thrust, wing surface area, wing span, and weight for the aircraft type, and the maximum steady state load factor and drag coefficient typical for aircraft of that category.

The aircraft types in Publication 7340 that do not have ascent profiles in the INM are then matched to the closest available profile. The matching is done by finding a best fit using climb rate and descent rate with a known profile of the given aircraft category.

The altitude portion of the descent profile does not vary by aircraft type. It is assumed that all flights use a four-degree glide slope from the cruising altitude to 12,000 feet, level off from 48 to 38 miles out, and then follow a three-degree slope into the airport. The speed portion of the descent profile is generated according to aircraft category and cruising speed. The values used for generating descent profiles are shown in Table 7-1 and are applied as described below.

During the initial descent from cruising altitude, it is assumed that a flight holds a constant Mach speed until it reaches a given indicated air speed (IAS), which is then held constant until the flight reaches 12,000 feet. The flight is then assumed to decelerate at a constant rate to 250

[†] Hale, Francis J., *Aircraft Performance, Selection, and Design,* John Wiley & Sons, Inc., New York, N.Y. 10022, 1984, pp. 74–84.

knots while at 12,000 feet. The flight is then assumed to descend at constant deceleration until it reaches the specified speed at ten miles out. The speed is held constant from 10 miles out to landing. During the flight profile modeling, the profile with the Mach speed closest to the cruising speed within the proper aircraft category is used for a given flight. If the cruising speed of a flight is less than any of the provided values, the flight is modeled as descending at its constant cruising speed until it reaches 12,000 feet.

Table 7-1. Aircraft Descent Profile Data

Aircraft Types	Initial Descent (Mach/IAS)	Speed Leaving 12000 feet (IAS)	Speed 10 Miles Out (IAS)	Landing Speed (IAS)
Heavy Jets, Fighters	.85/350 (kts) .80/330 .75/310 .70/290	250 (kts) 250 250 250	140 (kts) 140 140 140	140 (kts) 140 140 140
Large Jets	.85/350 .80/330 .75/310 .70/290	250 250 250 250	120 120 120 120	120 120 120 120
Turboprops, Piston props, Helicopters	.70/290	250	90	90

7.1.2 Determining a Flight's Profile

The flight profile is modeled using the aircraft profile models, the filed cruising altitude and speed, and the distances along the flight path. A graphic representation of the flight profile modeling is shown in Figure 7-1. If the first element of the flight path is an airport, the flight is assumed to be on the ground. If the first element is any other fix type, the flight is assumed to be already en route at its filed speed and altitude. If the flight is on the ground, an ascent profile is chosen using the aircraft type and the length of the flight. As the flight path is followed, the ascent profile provides the altitude, speed, and phase (ascent, en route, descent) as a function of the distance from the origin airport. The ascent is followed until it intersects the filed cruising altitude. The flight is assumed to be at the filed cruising altitude and speed for the entire en route phase.

If the last element in the flight path is an airport, the flight is modeled as landing at the airport; if the last element is any other fix type, the flight is assumed to hold its cruising speed and altitude through the last fix. If a landing is called for, the flight profile model checks at each event for an intersection between the current flight path and the descent profile. The descent phase can begin during the ascent or en route phase. Figure 7-1 shows a flight descending from its en route phase. Once the descent phase begins, the descent profile provides the altitude, speed, and phase as a function of the distance from the destination airport.

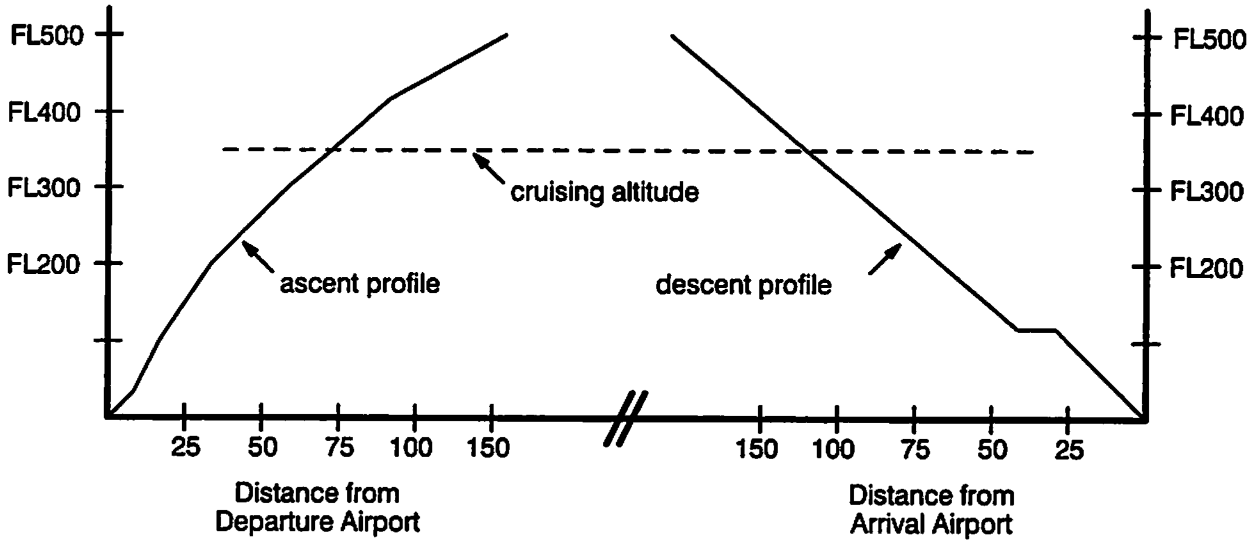

Figure 7-1. Flight Profile Modeling

7.2 Flight Event Generator

The flight event generator traverses the grid cells which connect the waypoints of the flight as determined by the field 10 processor and creates a list of events for the flight. At each cell, the altitude of the flight is determined by the flight profile model. The event generator then determines whether any events should be generated. The information generated for each event is summarized in Table 7-2. The event information includes all the data needed for the subsequent flight time and traffic demand computations. The following sections describe how each type of flight event is generated.

7.2.1 Airport Events

An event is always generated for the origin and destination airports. The element type, name, and position are known from the grid database. The type of event is a departure for the origin airport, and an arrival for the destination airport. The altitude of an airport event is always set to zero; the speed is known from the ascent or descent model. The distance and bearing from the previous event are undefined for a departure, and are computed from the previous event for an arrival.

7.2.2 Fix Events

A fix event is generated for any fix contained in a grid cell along the flight path. The element type, name, and position for any fix are known from the grid database. The event kind is determined to be a low altitude crossing, high altitude crossing, or super-high altitude crossing by examining the altitude of the flight position. (A low fix is between 0 and 23,900 feet, a high fix is between 24,000 and 34,900 feet, and a super-high fix is between 35,000 and 99,900 feet.) The altitude, speed, and phase are known from the flight profile model. The distances and bearings from the previous events are computed. If an airborne delay time has been saved by the field 10 processor for this fix, the delay time is assigned to the event.

Table 7-2. Flight Event Data

Field Name	Contents
Element Type	airport NAVAID adapted fix waypoint ARTCC low sector high sector super-high sector jet airway Victor airway
Element Name	–
Event Kind	If airport: departure, arrival If fix: low crossing, high crossing, super-high crossing If ARTCC, sector, or airway: entry, exit
Location	latitude/longitude
Altitude	–
Speed	–
Distance	n.m. from previous event
Bearing	deg. from previous event
Phase	takeoff ascent en route descent landing
Waypoint Flag	true/false
Airborne Delay	minutes (fixes only)

7.2.3 Airway Events

An airway event is generated when a flight path joins or leaves a jet or Victor airway. The element type and name for the airway are known from the grid database. The altitude, speed, and phase of the flight are known from the flight profile model. The distances and bearings from the previous events are computed once the locations are known.

The occurrence and location of the airway events are then determined. The event generator examines each cell along a flight path for the existence of an airway. If the flight's altitude is below 18,000 ft., the event generator looks for only Victor airways; if the flight is at or above 18,000 ft., the event generator looks for only jet airways. When the event generator encounters a new airway it must decide whether the flight is merely crossing the airway or truly following it. As shown in Figure 7-2, a flight path and an airway can contain some common number of grid cells (in this example, three) even though it is only crossing the airway. Therefore, the event generator requires that a flight path coincide with an airway for at least *five* grid cells before deciding the flight is really following the airway. When this condition is met, the event generator creates an *airway entry* event located at the first of the five grid cells which were found to intersect. The flight is likewise considered to be on the airway until that airway does not appear in five consecutive grid cells in the flight path. An *airway exit* event is generated with the location of the last grid cell to contain the airway.

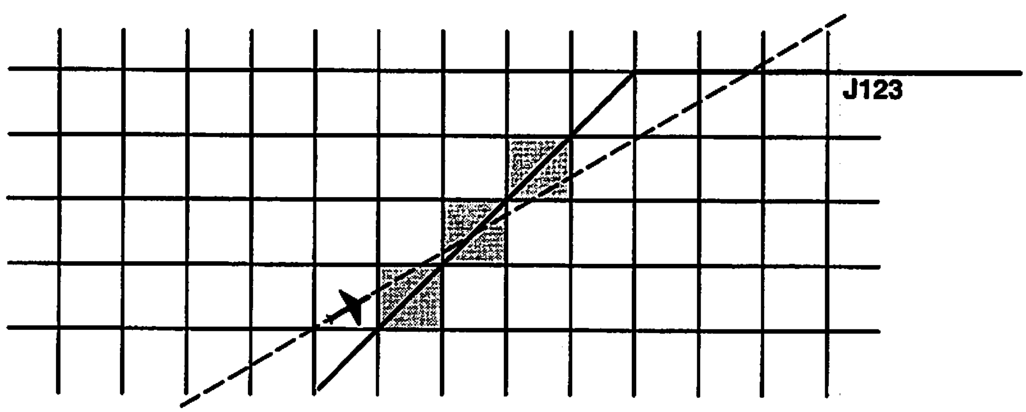

Figure 7-2. Airway Intersection

7.2.4 Sector Events

The event generator determines sector events by following the flight path through the grid database. At each grid cell, the altitude of the flight is determined by the flight profile model. The event generator then checks the flight altitude against the altitude limits for each sector containing that grid cell to determine which sector the flight is currently in. The assigned sector for the current grid cell is then compared to the sector determined for the previous cell. If the flight is in the same sector, no event is generated; if different, a sector exit is generated for the previous sector and an entry for the current sector. The sector events are generated for all sectors. The sector names are known from the grid database. The position of the event is defined to be the center point of the current grid cell. The altitude, speed, and phase are known from the flight profile model. The distances and bearings from the previous events are computed.

Some high sectors are covered by super-high sectors; some are not. The ETMS ensures that all flights at super-high altitudes are assigned to a sector even if no super-high sector is defined at the flight location. When the event generator determines that an estimated flight altitude is above all sectors in a grid cell, it assigns the flight to the highest sector in that grid cell.

The internal representation of the sector boundaries can cause unwanted sector events to be generated when a flight path coincides closely with a sector boundary, as shown in Figure 7-3. The boundary between sectors A and B is actually a straight line, but is approximated as the boundary of the grid cells which compose the sectors. If the event generator were to use the internal sector boundary literally, it would generate numerous sector exits and entries between sectors A and B for the illustrated flight path.

The event generator avoids this by checking the distance between each sector entry and exit; if the distance is found to be less than ten miles, the events are not generated. Therefore, if the flight starts in sector A, goes through a sequence of close entry/exit pairs, and ends in sector A, the flight is modeled as always being in sector A. If the flight starts in sector A, goes through a sequence of close entry/exit pairs, and ends in sector B, it is modeled as having one exit from sector A and one entry to sector B. In the latter case, the exact point at which the exit/entry is modeled becomes somewhat arbitrary. An additional effect of this algorithm is that flights that cut the corner of a sector for less than ten miles are modeled as never entering that sector.

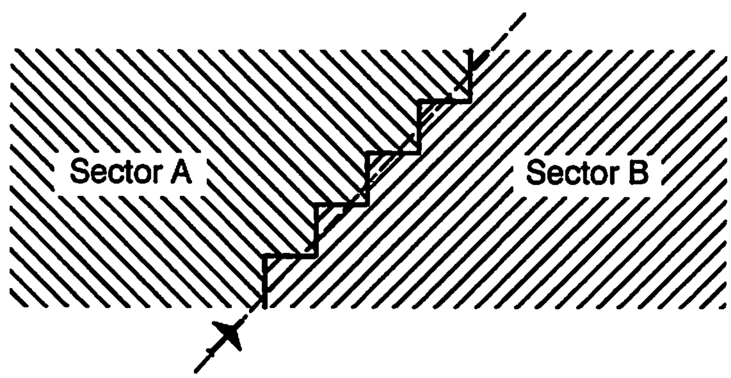

Figure 7-3. Sector Boundary Events

7.2.5 Arrival Fix Events

Arrival fix events are generated after all the other events have been generated for a flight; this is done without referencing the grid. Arrival fix events are generated only for flights arriving at an airport that has defined arrival fixes. All flights arriving at such an airport are assigned to the closest arrival fix regardless of whether the flight path, as specified by the field 10, physically intersects one of the arrival fixes. Because the arrival fix may not lie on the flight path, arrival fix events are processed and maintained separately from the other flight events.

The arrival fix event is determined as follows. The airspace around an airport can be thought of as a circular region with a radius of 30 miles and with the airport as the center; the airspace is divided into pie-shaped slices as shown in Figure 7-4. The number of slices is equal to the number of arrival fixes. The sides of the slices are drawn as lines which bisect the angle created by two adjacent fixes and the airport as the vertex. Each pie slice is associated with an arrival fix. If arrival fixes are defined for different aircraft categories (for example, jets, props, etc.), a different pie is created for each defined aircraft category.

After the event list has been generated, the event generator starts from the destination airport and works back along the flight path thirty nautical miles to determine the point **x** at which the flight path intersects the edge of the pie. The event generator then determines which slice the intersecting point lies in and assigns the flight to the arrival fix associated with that slice. Figure 7-4 shows that the flight is assigned to arrival fix D. The location of the event is defined as the location of the arrival fix, although the flight path does not physically cross the arrival fix. An arrival fix event has no altitude or speed associated with it.

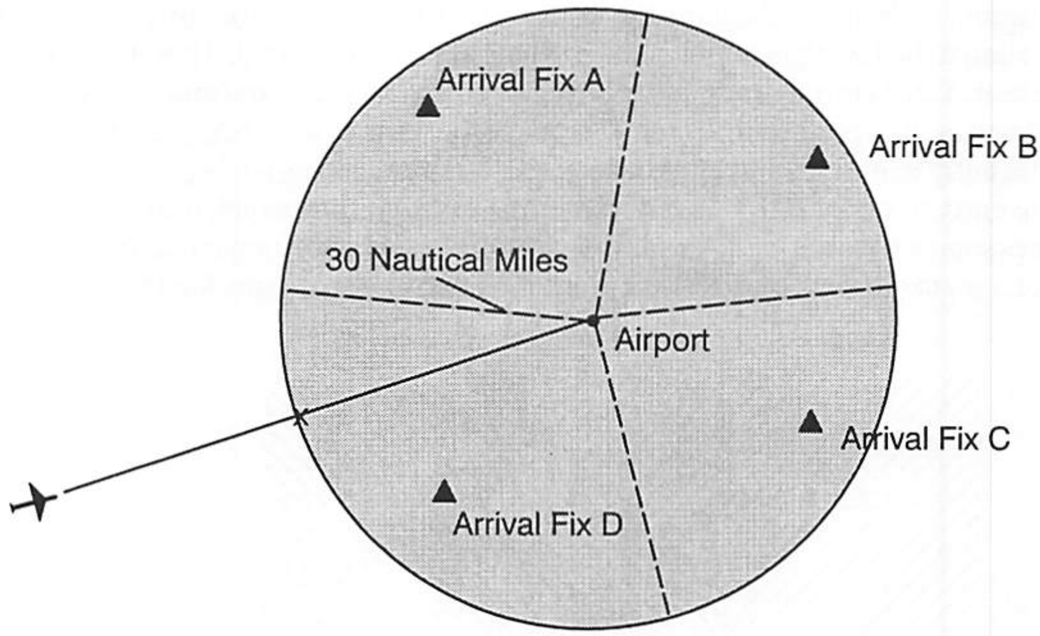

Figure 7-4. Arrival Fix Processing

7.3 Flight Time Modeling

The description of the ETMS flight-time modeling is divided into three sections. Section 7.3.1 describes how the ETMS models departure times. Section 7.3.2 describes how the ETMS predicts times for the other events in a flight's event list. Section 7.3.3 describes how the ETMS updates event times based on the data received in position update (TZ) messages.

7.3.1 Departure Time Modeling

The ETMS receives four types of departure times from incoming flight data messages: *scheduled* departure times from FSs, *filed* departure times from FZs, *controlled* departure times from EDCTs, and *actual* departure times from DZs. The departure times can mean two different things depending on the message and flight type:

(1) *Wheels-up* times refer to the time that a flight actually takes off. Controlled and actual departure times are always wheels-up times. Filed departure times are wheels-up times only for unscheduled carriers (mainly military and general aviation users). Scheduled departure times are never wheels-up times.

(2) *Gate pushback* times refer to the time that a flight expects to push back from the gate at a terminal after loading passengers. Scheduled departure times are always gate pushback times. Filed departure times are gate pushback times only for scheduled carriers. Controlled and actual departure times are never gate pushback times.

The ETMS expresses all departure event times as wheels-up times. The ETMS converts gate pushback times to wheels-up times by adding a taxi time estimate. Taxi time estimates are determined empirically from historical data for each airline operating at each pacing airport. A taxi time of ten minutes is used for all other airlines and airports. The actual taxi time estimation process is described in Appendix B. Taxi times are applied only to gate pushback times; that is, scheduled departure times and filed departure times for scheduled carriers. Wheels-up departure times are not adjusted.

NOTE: Other departure time factors, such as departure queueing delays, are not factored into the departure event time modeling.

The ETMS does not model other departure delays, such as queueing delays, in the sense of predicting those delays. However, the ETMS does track observed departure delays (see Section 5.3.3 — Flights Not Activated). When the ETMS observes that a flight has not been activated five minutes after its expected departure time, the ETMS adds five minutes to the modeled departure time. As long as the flight is not activated, the ETMS checks for delays and pushes the modeled departure time back until one hour after the original proposed departure time. Each time the ETMS re-computes the modeled departure time, it re-computes the other modeled event times accordingly.

7.3.2 En Route Time Modeling

The event time computations are based on data that has been assigned to the events by the event generator (see Section 7.2), namely, the speed at the event and the distance from the previous event. During the ascent and descent phases, the flight may be either accelerating or decelerating. Also, the distance between events varies. The acceleration or deceleration is approximated between events by integrating intermediate values computed at regularly spaced intervals between the two events.

During the en route phase of the flight, the event times are computed using the assigned cruising speed adjusted by grid wind values. The grid winds data provides wind vectors (speed and direction) estimated for different altitude strata at a matrix of locations covering the continental U.S. grid winds estimates are provided for the next 12 hours. For each en route event, the ETMS applies the wind vector for the nearest location and the appropriate altitude. The wind vector is combined with the aircraft vector to compute an estimated *speed over ground* for the event. The ETMS uses the speed over ground with the distance between events to compute the next event time. If the speeds over ground are different for two consecutive en route events, the average of the two speeds is used.

> **NOTE:** The grid winds data is used for FS and FZ messages. However, the grid winds data is not used to adjust speeds during the ascent and descent of the flight.

If an event has an airborne delay assigned to it, the delay time is added to that event time. That is, the delay is considered to occur before the event. Airborne delays can only be assigned to fix events.

An exception is made for computing the time of an arrival fix event, if one exists. The time at the arrival fix is assigned to be 15 minutes earlier than the arrival time (that is, the event time at the destination airport). The time is not computed as part of the normal event list because the flight path may not actually cross the arrival fix (see section 7.2.5). Although the arrival fix time may not be consistent with the other flight events around the arrival fix, it assures that the order of the flights arriving at the destination airport is preserved at each arrival fix.

7.3.3 Position Update Processing

The position update messages (TZs) contain the tracked position and speed for an active flight. The ETMS uses the TZ data to update the predicted time and speed of future events and to estimate the actual time and speed of past events.

The TZ data presents a unique set of problems for the ETMS. All other flight data used for the modeling of flights is defined strictly in the context of the modeled flight events; that is, every position, speed, altitude, and time is associated with an ETMS-defined event. However, TZs are received at arbitrary times (from an ETMS perspective) and arbitrary positions. The ETMS must therefore first determine how to relate a TZ's data to the pre-defined event list, then how to update the event times.

An example of TZ processing is shown in Figure 7-5. The TZ processing first checks whether the TZ position is close enough to the flight path (as defined by the event list) to be accepted. The TZ processing checks the position by computing the perpendicular distance from the position to the closest part of the flight path (in this case, the segment between events 3 and 4); if this distance is less than 15 miles, the TZ is processed normally. The TZ processing then finds the *last actual event* (closest previous event, event 3 in this case) and estimates the time of that event based on the distance from the TZ position to the event, the TZ speed over ground, and the TZ time. The TZ processing next uses the speed over ground from the TZ to update the cruising speeds in the last actual event as well as the remaining *predicted events* that are in the en route phase of the flight. Finally, the TZ processing re-computes the times for the predicted events in the manner described in Section 7.3.2.

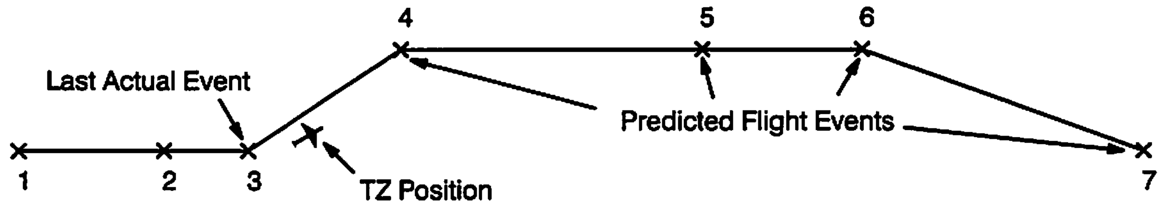

Figure 7-5. Example of TZ Processing

Once a last actual event has been identified, the TZ processing will not change the estimates for that event. Referring again to the example, if another TZ is received before the flight reaches event 4, the values estimated for event 3 will not be changed. The TZ processing will, however, update speeds for the predicted events, extrapolate the predicted time for event 4 using the TZ speed, time, and position, and update the times for events 5, 6, and 7.

If a TZ is found to be more than 15 miles from the flight path, the TZ processing monitors subsequent TZs to determine whether the flight is deviating from or coming back to the flight path. If the flight is found to be deviating from the flight path by more than 0.5 miles per minute, the TZs are ignored (i.e., the event times are not changed using the contents of the TZ). If the flight is moving parallel to the flight path or coming back to the flight path, the TZs are processed as described above.

> **NOTE:** In many cases, when a flight is found to be deviating from its flight path, the next update to the flight path (usually from a UZ message) is found to coincide with the TZs. The normal TZ processing is resumed in this case.

The TZ processing is further complicated by the existence of airborne delays. A flight usually maneuvers far off its filed flight path during an airborne delay. Therefore, when a flight reaches a fix where it is expected to take a delay, the TZs are ignored until the delay time has expired. The TZ processing then returns to normal, as described above.

Another processing exception is for flights that file flight plans while flying VFR. The event list begins while the flight is en route, yet TZs are often received for positions before the beginning of the event list. In these cases, the first event time is extrapolated from the TZ speed, position, and time. The remaining event times are computed as before.

Section 8

Traffic Demand and Alert Processing

The ETMS uses the flight event data produced during the flight modeling to estimate the *traffic demand* at each monitored airport, sector, and fix, and to generate an *alert* whenever a traffic demand is projected to exceed a pre-defined *alert threshold*. Airport traffic demands are defined as the number of arrivals and departures in a 15-minute interval. Sector traffic demands are defined as the peak number of aircraft in a sector at any one time during a 15-minute interval. Fix traffic demands are defined as the number of fix crossings within a certain altitude range during a 15-minute interval; the altitude ranges used correspond to the altitude limits of the sectors that overlay the fixes. Alert thresholds are defined in the same manner as the traffic demands.

The ETMS computes the traffic demands from the flight event data determined by the ETMS flight modeling (see Section 7) based on the ETMS flight database processing (see Section 5). The ETMS keeps a traffic demand database (TDB) which contains the traffic demand count and alert threshold for each element, event type, and 15-minute interval. When a projected demand exceeds a threshold, the ETMS displays an alert for that element on the *ASD*. The traffic management specialist can request data about the alert through the *ASD*; the alert data includes a comparison of the demands and the alerts, and flight lists for the alerted element. The display of the alerts and the alert data is described in Section 10.

This section consists of four parts. Section 8.1 describes the airports, sectors, and fixes that are monitored. Section 8.2 describes the processing of the traffic demand data. Section 8.3 describes the alert thresholds. Section 8.4 describes how the alerts are generated.

8.1 Monitored Elements

The ETMS map database and grid database contain an enormous number of airports, sectors, and fixes. Maintaining traffic demand counts and lists for all known elements would require huge processing resources and would be of questionable benefit. Instead, the ETMS maintains traffic demand data only for elements determined to be of reasonable benefit to traffic management. The *monitored elements* are specified in a manually maintained list. The monitored elements list is generated according to the criteria described below; however, elements can be added to or deleted from the list according to direction from traffic management.

(1) Monitored Airports — The monitored airports consist of all U.S. airports with *scheduled* flights, all Canadian and Mexican airports with scheduled international flights, and several manually selected international airports.

(2) Monitored Sectors — The ETMS monitors all sectors in the airspace over the CONUS. The monitored sectors correspond to the sectors displayed on the *ASD*

in the low, high, and superhigh-sector, map overlays. Oceanic sectors are not monitored.

(3) Monitored Fixes — The ETMS monitors all fixes defined by traffic management as monitored fixes. They may or may not correspond to arrival fixes. Furthermore, the monitored traffic for a fix consists only of those flights whose flight path actually crosses the fix; that is, arrival fix events are not passed to the TDB for inclusion in the fix counts. The traffic for a fix is maintained in categories which correspond to the sector strata which overlay the fix; that is, if a fix has low, high, and superhigh sectors over it, it will have low, high, and superhigh traffic counts. This is all borne out by the fact that the specialist can get low, high, and superhigh fix alerts but cannot get an arrival fix alert.

8.2 Traffic Demand Processing

The ETMS maintains the traffic demand database so that it always reflects the state of the data in the flight database. When the ETMS adds a flight to the flight database (e.g., when an FZ is received), it also adds it to the traffic demand database. When the ETMS updates a flight in the flight database (e.g., when a DZ is received), the traffic demand database is updated correspondingly. If a flight is cancelled in the flight database (e.g., when an RZ is received), the flight is removed from the traffic demand database. In every case, the information (flight ID and status) needed to update the traffic demand database is in the flight database or is produced by the ETMS flight modeling (the element names, the event types, and the event times).

The ETMS maintains separate traffic demand counts for flights with statuses of *scheduled*, *proposed*, and *active*. The ETMS determines the appropriate count to update, for a flight, by examining the flight status assigned in the flight database (see Section 5). The ETMS uses the separate counts to distinguish alerts caused by flights in the air (active flights) from alerts caused by flights on the ground (scheduled and proposed); the *ASD* displays the former as red alerts and the latter as yellow alerts (see Section 10).

The maintenance of the traffic demands for airports and fixes is the most straightforward, since those demands consist simply of counts of the flight events (arrivals, departures, and crossings) as they appear in the event lists. When the ETMS adds a flight to the flight database, it performs a separate *add* transaction for every airport and fix event in the flight's event list. Similarly, when a flight is cancelled, the ETMS performs separate *delete* transactions for each airport and fix event. When the ETMS changes a flight status or updates a flight event list, it performs a delete transaction for each previous airport and fix event, and it performs an add transaction for each updated airport and fix event.

Figure 8-1 illustrates the processing for an add transaction of an airport departure event for a proposed flight. The ETMS uses the element type and name to look up the airport in the traffic demand database. The ETMS computes the 15-minute interval into which the event falls from the event time. The ETMS identifies the flight status as proposed and the event type as a departure. Accordingly, the ETMS increments the proposed departure count for the computed interval and inserts the flight into the departure flight list for that interval (one flight list is maintained for flights of all statuses). If the transaction type were *delete*, the same steps would be performed except that the count would be decremented, and the flight would be removed from the flight list. The fix demand processing is similar to the airport processing except that instead of arrival and departure events, there are low, high, and super-high crossings.

The maintenance of the sector traffic demands is more complex than for airports and fixes because the sector demands (i.e., the peak numbers of flights at any one time during an interval) are not just counts of the sector events (i.e., sector entries and exits). To determine the peak demands, the ETMS processes the sector events in pairs. When the ETMS adds a flight to the flight database, it performs an add transaction for the sector entry and exit pairs. When a flight is cancelled, the ETMS performs a delete transaction for sector entry and exit pairs.

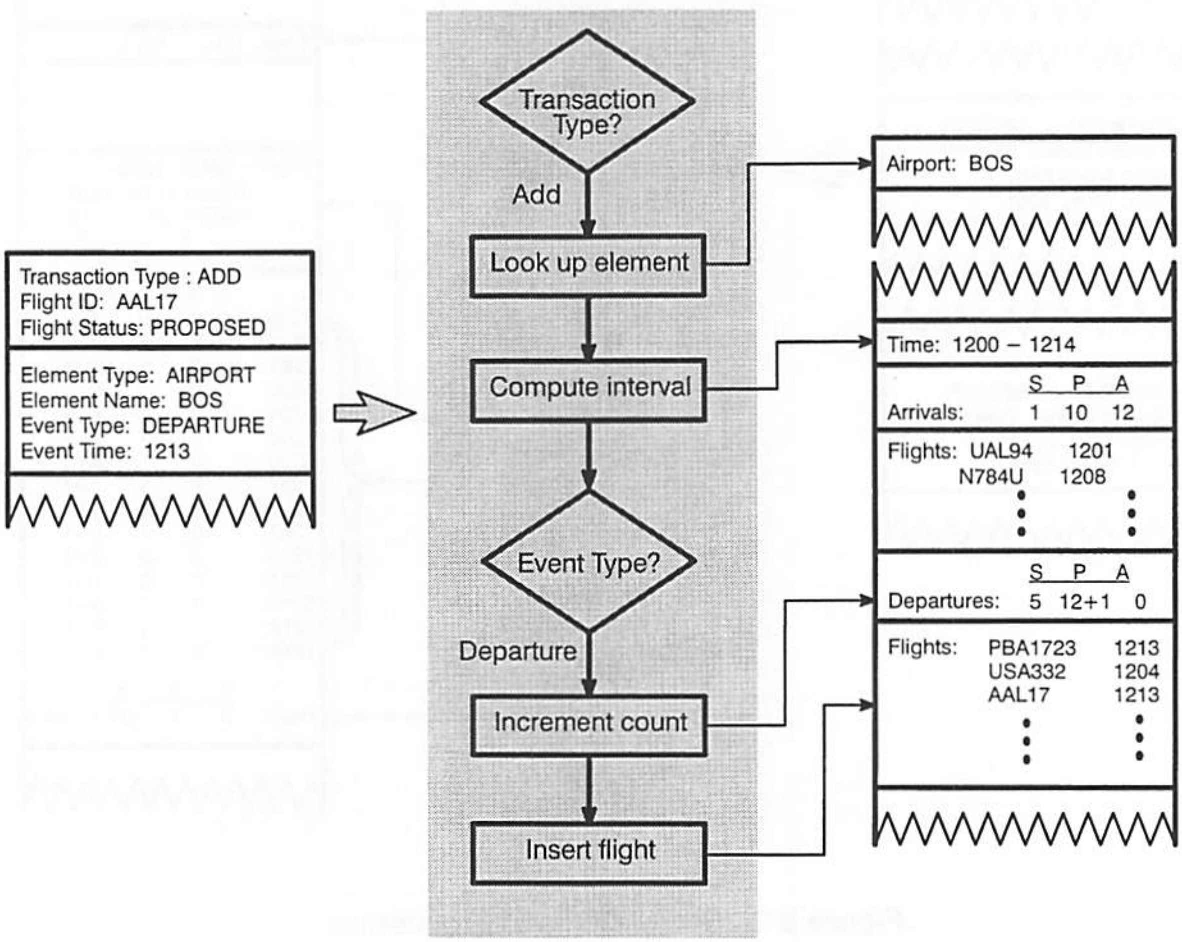

Figure 8-1. Airport Demand Processing

When the ETMS updates a flight's event list, it performs a delete transaction for each previous sector entry and exit pair and an add transaction for each updated sector entry and exit pair. The ETMS maintains separate demand counts by flight status for sectors as it does for airports and fixes.

Figure 8-2 illustrates the processing of an add transaction for a sector entry and exit pair for an active flight. The ETMS initially processes the transaction as it does for the airport transaction. The ETMS uses the element type and name to look up the sector in the traffic demand database. The ETMS computes the 15-minute interval into which the entry and exit events fall from the event times and inserts the flight into the flight lists for those intervals.

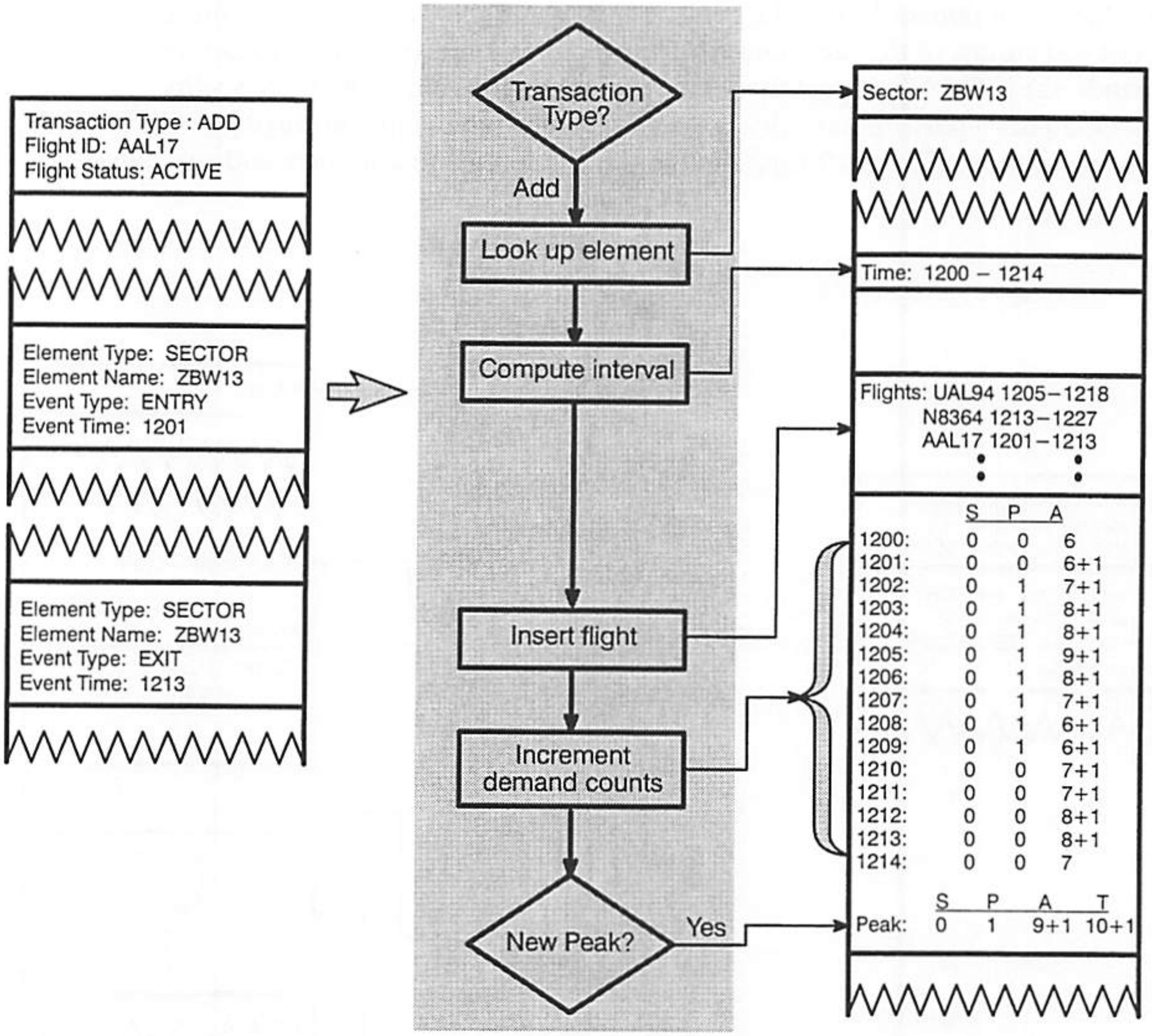

Figure 8-2. Sector Demand Processing

At this point, the processing of the sector transaction departs significantly from that of the airport. The sector traffic demand database contains minute-by-minute counts of the number of flights in the sector; a separate set of counts is maintained for each flight status. In this example, the ETMS updates the active demand counts, starting at the sector entry time (1201) and ending with the sector exit time (1213). The ETMS then checks the updated active demand counts to see if any exceed the previous active peak demand; if so, the active peak is set to the new value. Similarly, the ETMS checks whether there is a new *total* peak. The ETMS determines the total peak by summing the scheduled, proposed, and active count for each minute, and by taking the maximum of those sums.

A sector delete transaction is handled in a manner similar to an add, but counts are decremented instead of incremented. The ETMS removes the flight from the flight list, decrements the minute-by-minute demand counts, and checks to see if any peak values should be reduced.

The example in Figure 8-2 shows the case where the sector entry and exit events are in the same interval, which is not always the case. When the sector events happen in two different intervals, the ETMS computes the entry and exit intervals. The ETMS inserts the flight into the flight list for the entry and exit intervals and any intervals in between. As before, the ETMS updates the minute-by-minute demand counts from the entry time to the exit time. The ETMS then checks the minute-by-minute demand counts against the peak demands for each time interval involved, and updates the peak counts as necessary.

8.3 Alert Thresholds

An alert threshold is defined as the level of traffic demand required at an element for it to be brought to the attention of a traffic management specialist. The alert thresholds are sometimes referred to as the *capacities* of the elements, but are not capacities in the classical sense. Alert thresholds for an airport are defined by the numbers of arrivals and departures in a 15-minute interval (two separate values). Alert thresholds for a sector are defined by the number of aircraft in the sector at any one time. Alert thresholds for a fix are defined by the number of any low, high, or super-high flights (up to three separate values) crossing the fix in each 15-minute interval.

NOTE: Not all airspace is divided into low, high, and super-high airspaces. In some cases the airspace altitude parameters are from 000 to 999. In other cases airspace is divided into low and high (with the top altitude as high as 999) airspace only.

The ETMS maintains two sets of alert thresholds for each element, event type, and 15-minute interval: the *nominal* threshold and *today's* threshold. The nominal thresholds are a set of default values which are stored in a manually maintained data file and read into the traffic demand database. Each day, the ETMS uses the nominal thresholds to initially set today's values. During the day, the ETMS uses today's values to check traffic demands and generate alerts. Authorized traffic management specialists can change today's by using the **CAPS** command available on the *ASD*.

8.4 Alert Generation

The ETMS compares traffic demand values to alert thresholds every five minutes to see if any alerts exist. The ETMS generates an alert if a traffic demand exceeds an alert threshold for any 15-minute interval in the next length of time as defined by the specialist (up to nine hours). For airports and fixes, the ETMS generates a red alert only if the active demand exceeds the threshold; it generates a yellow alert if the total demand (scheduled plus proposed plus active flights) is greater than the threshold. For sectors, the ETMS generates a red alert only if the active peak demand exceeds the threshold; it generates a yellow alert if the total peak demand exceeds the threshold. A traffic management specialist can alter the number of alerts displayed on his or her *ASD* by specifying the desired element types, or by specifying a time range for the alerts other than the one-hour default.

The ETMS also detects when alerted situations are relieved. If the traffic demand is less than the alert threshold for an interval that was previously alerted, the alert for that interval is disabled. When no remaining intervals are alerted for a previously alerted element, that element disappears from any *ASD* that is currently displaying alerts.

The ETMS also monitors whether alerts for specific 15-minute intervals have been turned green by authorized traffic management specialists. When an authorized specialist has used the **turn green** command to indicate that an alert is being resolved for a specific 15-minute interval, the appropriate time bar interval changes color: from red to green with a red horizontal stripe or from yellow to green with a yellow horizontal stripe. The stripe indicates that there used to be an alert for the element during that 15-minute interval. This allows the user to more easily monitor that element.

If other 15-minute intervals shown on the time bar are still alerted, the alert icon on the map will remain red or yellow. However, if all alerted 15-minute intervals are turned green, the icon on the map will also turn green.

Once a specialist turns an alerted interval green, the system will not re-alert the element for that time interval, even if the situation in that interval becomes much worse. However, if the monitor alert bar charts and text reports indicate that the situation has degraded, the specialist can choose to toggle off the **turn green** command to indicate that the element should be monitored again. Afterwards, alerts could again be generated for the interval.

The effects of the **turn green** command will appear immediately on the *ASD* of the specialist who issued the command. During the next monitor/alert update, such effects will appear on all other ASD displays at all ETMS sites.

Section 9

Traffic Management Data Processing

The ETMS traffic management data processing performs the steps specific to the implementation of a ground delay program. The FAA issues ground delay programs in order to prevent aircraft from arriving at an airport at a rate well above the rate at which they can feasibly land. A traffic management specialist at the ATCSCC initiates a ground delay program by running the Selected Controlled Departure Time (SCDT) program. The results of the program are sent to the TMUs at the ARTCCs, the airlines, and the ETMS via the IFCN. The ETMS saves these flight controls and determines how to apply them to the flight and traffic databases. The ETMS subsequently issues control time (CT) messages to the NAS causing the controlled departure times to take effect. The ETMS also receives substitution requests (SIs) from the airlines, which specify changes to the issued controls. The ETMS verifies the SIs by using rules determined by FAA policy; if accepted, the ETMS applies the changes and issues the appropriate CTs. Finally, the traffic management specialist can use ETMS commands to control the traffic management data processing.

The traffic management data processing is closely related to the other flight data processing. The manner in which the ETMS uses the traffic management data to update the flight database is fully described in Section 5 —*Flight Database Maintenance*. The contents of the EDCT and SI messages are described in Section 5.1.4 —*EDCT Messages*, and Section 5.1.5 —*Flight Substitution Messages*. The traffic management data processing is also dependent on the traffic demand database described in Section 8.2 — *Traffic Demand Processing*.

Section 9.1 provides a detailed description of how the flight controls are applied. Section 9.2 describes how the ETMS validates the SI messages. Section 9.3 describes the processing of the traffic management commands.

9.1 Applying Flight Controls

The ETMS receives three types of flight controls: Estimated Departure Clearance Times (EDCTs), Fuel Advisory delay tables (FAs), and flight substitution requests (SIs). EDCTs and FAs are generated by the SCDT program at the ATCSCC. The EDCTs are flight-specific controlled departure times for flights known at the time that the SCDT program is run. The FAs are average delays to be applied to flights that file flight plans after the SCDT program is run. The SIs are specified by flight cancellations (CNXs), substitutions (SUBs), replacements (RPLs), and exchanges (EXCs), which the airlines generate in response to EDCTs. The SIs allow an airline to react to the ground delay program to reflect its own dispatching priorities. The ETMS uses the flight controls, in combination with NAS flight plan messages to issue CTs. The ETMS sends CTs to the NAS host computer at the flight's departure ARTCC, which in turn prints the control times on the flight strips sent to the air traffic controllers for implementation. The following sections describe how the ETMS generates CTs for EDCTs, FAs, and SIs. Section 5 — *Flight Database Maintenance* describes how the ETMS updates the flight database with control data.

9.1.1 EDCT Processing

The SCDT program computes ground delays using a flight list obtained from the ETMS traffic demand database. The flights in the ETMS list that receive EDCTs may have a status of either *scheduled*, *filed*, or *controlled*. When the ETMS receives the EDCTs, it checks the flight status. If a flight is *filed*, the ETMS immediately issues a CT for the flight and notes that a CT was sent. If a flight is *scheduled*, the ETMS stores the controlled departure time and waits for a flight plan to be received. When the flight plan matching the EDCT is eventually received, the ETMS issues the corresponding CT and records that a CT was sent. If a flight already has a *controlled* status, the ETMS overrides the previous control with the new controlled departure time. The ETMS then checks to see if a CT has already been sent; if so, a new CT is issued, if not, the ETMS waits for the flight plan.

9.1.2 FA Processing

An FA specifies the arrival airport involved in the ground delay program, the average delays for the time intervals being controlled, and the departure ARTCCs eligible for controlled delays. When the ETMS receives a flight plan for which it does not have an EDCT, it determines the flight's eligibility for an FA delay; i.e., it checks to see if the flight is arriving at a controlled airport during a controlled time interval and departing from an eligible ARTCC. If so, the ETMS adds the average control delay to the flight's filed departure time to produce a controlled departure time. An FA Table is sent to the host computer. The host computer checks the EDCTs and issues a CT for the FA delay.

The traffic management specialist runs the SCDT program by first requesting a list of arrivals from the ETMS. Only the flights in the arrival list will be assigned EDCTs; the FAs are intended to ensure that flights not known at the time will get equitable delays. However, between the time the arrival list is requested and the time the FAs are issued, the ETMS may receive additional flight plans that should be controlled. The ETMS detects these flights by getting a new arrival list after the FAs are received. By checking the new list against the EDCT list, the ETMS can identify additional flights that should be delayed using the FA data. The ETMS processes these flights as described above.

9.1.3 SI Processing

The substitution requests (SIs) contain flight cancellations (CNXs), substitutions (SUBs), replacements (RPLs), and exchanges (EXCs), specified by an airline for its own flights that have received EDCTs. These flight substitution messages are used in pairs: a CNX with a SUB, an RPL with a SUB, or an EXC with an EXC. The bulk of the ETMS SI processing is the validation of the SI, described in Section 9.2. Once the ETMS determines that an SI is valid, the application of the changes is straightforward. Flights that are cancelled by a CNX are ignored with respect to applying the flight control; i.e., the ETMS takes no action regardless of whether or not a CT has been issued. Flights that are substituted or exchanged are processed exactly like EDCTs received for flights with a *controlled* status. The ETMS overrides the previous controlled departure time with the time from the SUB or EXC. If a CT has already been sent for the flight, the ETMS issues a new one with the new time; if not, the ETMS waits for the flight plan to be received.

9.2 SI Validation

The airline use of SI requests is carefully controlled by FAA policy. The FAA wishes to allow the airline a reasonable amount of flexibility so that the airline can operate efficiently and meet its customers' needs; however, too much flexibility could negate the intended effect of the ground delay program: to provide a smooth arrival flow at an acceptable rate. Therefore, the FAA has defined a specific set of rules to govern the use of the SI request. The ETMS checks any SI request against the rules before performing any processing of the messages. SI processing will terminate if an error is encountered. Substitutions approved before encountering the error are processed as normal.

An SI request contains four types of messages, used in combination with each other to specify a sequence of actions desired by the airline. An airline uses a CNX message to cancel a flight that has received a controlled departure time, a SUB message to substitute one flight's controlled arrival time for another's, an RPL message to indicate a flight whose arrival time is available for use by another flight (via a SUB message), and an EXC message to exchange the arrival times of two flights. The messages are used in pairs. A CNX followed by a SUB requests that a flight be cancelled and another flight be allowed to use the cancelled flight's arrival slot. An RPL followed by a SUB requests that a flight be allowed to use the arrival slot for a flight that has previously been SUB'd. An EXC followed by an EXC requests that the arrival slots of the flights be exchanged. The message pairs (CNX/SUB or RPL/SUB) are used in a cascading fashion: one flight is cancelled and all subsequent flights are moved up into the previous flight's arrival slot.

The SI validation rules control the sequence and the content of the messages as follows:

(1) The SI messages must be specified in pairs of messages consisting of a CNX followed by a SUB, an RPL followed by a SUB, or an EXC followed by an EXC.

(2) A set of substitutions must start with a CNX/SUB pair that can be followed by any number of RPL/SUB pairs. (This is not applicable to EXC/EXC.)

(3) A flight that appears in an RPL message must have already appeared in a SUB message.

(4) The arrival time requested in a SUB message must be between the arrival time and 20 minutes after the arrival time in the preceding CNX or RPL message. The same window applies to EXC/EXC.

(5) The arrival time requested in a SUB message must be changed by at least one minute from the arrival time already assigned to that flight.

(6) Any flight appearing in a CNX, RPL, SUB, or EXC message must already have an EDCT assigned to it.

9.3 Traffic Management Commands

The traffic management specialist can use ETMS commands to examine traffic management data and to control the traffic management data processing.

The ETMS maintains all data regarding the ground delay programs for the current day. The traffic management commands allow the traffic management specialist to query the status of a particular flight or the status of a program at a particular airport. In the latter case, the specialist can request summary data for an airport or a detailed report of all flights for the airport. The specialist can also request a list of all currently controlled airports.

The traffic management specialist may purge controls for an airport after a ground delay program has been issued. When the ETMS receives a **purge** command, the controlled departure times for the specified airport and time range are marked as purged. When flight plans are received for the previously controlled flights, CT messages are not issued. However, controls that have already been implemented (i.e., the CTs have already been sent) will not be undone by a **purge** command.

The traffic management specialist may suspend (**sub off**) the SI privilege for an airport or for all airports. When SIs are suspended, the ETMS will reject all SIs received from airlines for the specified airport (or all airports). The ETMS will accept a **sub off** command only if a ground delay program is in effect for the specified airport (or some airport). If the specialist issues new EDCTs for an airport that currently has SIs suspended, the ETMS will apply the suspension to the new EDCTs as well. The specialist can reverse an SI suspension by entering a **sub on** command for an airport (or all airports). If no **sub on** command is received, the suspension will be in effect for the remainder of the day; i.e., the ETMS will automatically lift the suspension at the end of the current day.

NOTE: For the traffic management data processing, the ETMS considers the end of day to be 0800 UTC (0400 EST, 0100 PST).

The *EMail* function allows the specialist to create and send advisory and general messages to addresses on ARINC and NADIN networks as well as to any other ETMS node. Such messages may be constructed in free format or pre-stored messages may be selected. *EMail* execution is initiated by selecting the E–Mail icon on the *Tool Manager* bar on the user's workstation. Since *EMail* uses the *Router* function to transmit messages through *ETMS Communications* software, the *Router* must run on every workstation used to send or receive messages. The *EMail* process must be connected to a *Node Switch* to allow *EMail* to communicate with the *Router* for message transmission through the *ETMS Communications* software.

Section 10

Displaying Data

The ETMS displays data to the traffic management specialist through the *ASD* in three general forms: graphics drawn on a map, bar charts, and text reports. The *ASD* may display data automatically or only when requested by the traffic management specialist depending on the data type. The *ASD* provides three ways for the traffic management specialist to request data and control the way the data is displayed: *menus*, *keyboard commands*, and *semicolon commands*.

The ETMS also displays data to the specialist through the other *ETMS User Functions*: *Delay Manager*, *Traffic Management (TM) Shell*, and *Electronic Mail*. *The Delay Manager* displays bar charts and text reports automatically or by request. The *TM Shell* displays text reports by request. The *Electronic Mail* can be used to send and receive text messages. The specialist can access any of the *User Functions* by means of the *Tool Manager*.

An overview of the data displayed by the ETMS is provided in Section 2. The ETMS user documentation provides a complete description of what data the traffic management specialist can see and how the traffic management specialist interacts with the *ASD* and the other *User Functions*.

This section discusses how the data displays are generated. Section 10.1 describes the display of geographical data. Section 10.2 describes the generation of aircraft situation data displays. Section 10.3 describes the display of the monitor/alert data. Section 10.4 describes the presentation of the request reports. Section 10.5 describes the display of weather data.

10.1 Maps Database

The *ASD* displays geographical data graphically on the screen when requested by the traffic management specialist. The *ASD* can overlay the various geographical data types with each other and with aircraft situation and alert data. The traffic management specialist can request map overlay displays by using menus and keyboard commands. All this is included in the maps database.

The *ASD* displays geographical data, specified in spherical coordinates (i.e., latitudes and longitudes), on an essentially flat screen. To do so, the *ASD* projects the spherical coordinates onto a plane using an *Albers, equal-area projection*. The Albers projection is a conical projection, which intersects the earth in a circle with a given center point. Traffic management specialists can choose one of three center points for the projection: one appropriate for the contiguous United States (CONUS), one appropriate for Alaska, and one appropriate for the Atlantic Ocean area. The Albers projection is best suited for showing a large rectangular area with minimal distortion. Therefore the Albers projection as used by the *ASD* provides a reasonably small amount of distortion over the United States at the expense of extreme distortion for other parts of the world. Being an equal-area projection, the Albers projection shows geographic areas according to their proper relative sizes (e.g., a region that looks twice as big as another is really twice as big as the other).

> NOTE: The projected coordinates are used only for displaying the data. The internal representation of the geographical data used in the flight path processing is in spherical coordinates and does not contain any distortion.

The *ASD* places *location identifiers* on the display as follows. For elements which are defined by a point (i.e., NAVAIDs, fixes, airports), the *ASD* draws the location identifier so that the lower, left corner of the location identifier appears at the projected latitude and longitude of the element. For elements which are defined as a boundary (i.e., sectors, ARTCCs, SUAs), the *ASD* determines an approximate center point for the element. The *ASD* then draws the location identifier with its lower, left corner at the approximated center point.

When a traffic management specialist requests some type of geographical data on the *ASD*, the *ASD* enhances the display by deciding what elements of that type to draw and whether to draw the location identifiers. The decision is based on the element type and the *zoom level* (i.e., the amount of area being currently displayed: zoom level 1 shows the largest area, about 4000 miles horizontally, zoom level 2 shows the next largest, about 2000 miles, etc.) of the display. The display characteristics for the different map overlay types are as follows:

(1) Airports — Because of the extremely large number of airports known for the U.S., the *ASD* only displays names for those which have no numbers in the name (e.g., the name AK07 is not displayed). Such airport names are shown at any zoom level. At the lowest zoom level, the airport runways are shown.

(2) NAVAIDS — All NAVAIDs are shown starting at zoom level 3 (1200 miles).

(3) Arrival Fixes — All arrival fixes are shown starting at zoom level 3 (1200 miles).

(4) Departure Fixes — All departure fixes are shown starting at zoom level 3 (1200 miles).

(5) Runways — All known runways are shown at any zoom level. No location names can be displayed for the runways.

(6) ARTCCs — The ARTCC boundaries are shown at all zoom levels. The ARTCC names are toggled on or off (by using the shift/tilde).

(7) Sectors — The sector boundaries are shown at all zoom levels. The sector names are toggled on or off (by using the shift/tilde).

(8) Military Areas (SUAs) — The military area boundaries are shown at all zoom levels. The military area names are toggled on or off (by using the shift/tilde).

(9) Airways — The airway paths are drawn at all zoom levels. The airway names are toggled on or off (by using the shift/tilde).

(10) TRACONS — The TRACON boundaries are shown at all zoom levels. The TRACON names are toggled on or off (by using the shift/tilde).

(11) Grid Overlays — The grid overlays can by toggled on and off (by using a menu or quick key). The grid limits are displayed by solid lines.

10.2 Aircraft Situation Data

The *ASD* displays aircraft situation data graphically on the screen overlaid with geographical and alert data, as directed by the traffic management specialist. The traffic management specialist can turn the aircraft situation data on or off using a menu entry, a keyboard command, or a semicolon command. Once the aircraft situation data is turned on, the *ASD* automatically refreshes the display with the latest flight positions until the traffic management specialist turns it off. The traffic management specialist can use other menu entries, keyboard commands, and semicolon commands to control the flights that are displayed and the flight data that is displayed. The *ASD* can also replay aircraft situation data up to six hours old, as requested by the traffic management specialist.

The ETMS generates an aircraft situation update for the *ASD*s typically every one minute. The aircraft situation update contains an entry for every flight in the flight database that is currently marked active. Each flight entry includes the flight ID, aircraft type, altitude, speed, heading, field 10, waypoints, and estimated position. When a traffic management specialist has requested that aircraft situation data be displayed, the *ASD* automatically updates the display according to the parameters that the traffic management specialist has selected. The display parameters control the flights that get displayed, the color used to draw them, and the data displayed for those flights.

The *ASD* shows a flight on the display by drawing an icon at its estimated positions. The *ASD* enhances the flight display by drawing either a dot icon or an aircraft icon depending on the zoom level and number of flights. The *ASD* uses dot icons at zoom levels 1, 2, and 3 unless a specific flight subset (e.g., arrivals at ORD) has been selected. The *ASD* always uses aircraft icons at zoom levels 4 and below. When the *ASD* draws aircraft icons, the icon is pointed in the approximate direction that the flight is heading. Other data which may be displayed for flights includes the data block (containing flight ID, speed, altitude, aircraft type, and minutes to destination), the origin and destination pair, field 10, the waypoints (drawn graphically), and the last reported position.

The ETMS estimates the flight positions used by the *ASD*, because the aircraft situation data updates (TZs, TOs) received from the NAS are not synchronized. The ETMS receives an aircraft situation data update for each flight every five minutes; however, one flight will be updated at different times from another. Therefore, to provide a more consistent display, the ETMS estimates the positions for all the flights at the time it generates the display update.

The ETMS estimates flight positions in a straightforward manner. The time over which the flight must be extrapolated is the time of the update minus the time of the last position report. The difference (or delta time) is then used with the last reported speed and heading to compute the deltas in latitude and longitude. The deltas are applied to the last reported latitude and longitude to produce the estimated position.

A danger in extrapolating the aircraft positions is that if the ETMS were to stop receiving position updates for a flight, the flight would appear to continue to move with each *ASD* update. However, the estimated position for that flight would not have the same accuracy as positions estimated for flights with position updates. The ETMS therefore checks the last position update time when generating the update file. If a position update has not been received for more than seven minutes, a flag is set in the update file indicating that the flight is a *ghost*. The *ASD* draws the ghost aircraft on the display as hollow outlines, rather than solid (filled) aircraft. The traffic management specialist can then visually distinguish the more accurate positions from those which might be less accurate.

When the user selects a flight subset (e.g., flights arriving at ORD) for display, the *ASD* simply scans the flight data in the aircraft situation update to determine which flights meet the selection criteria. The selection criteria include some of the flight data that can be displayed (aircraft type, origin, destination, and altitude) and some data that can not be displayed (sectors, fixes, ARTCCs, or airways traversed). The ETMS includes these additional data items in the aircraft situation updates for this purpose. The ETMS extracts these items from the event list produced during the flight modeling.

The *ASD* displays old aircraft situation data by using ETMS aircraft situation updates that were previously generated and stored. The *ASD* retrieves the updates specified by the traffic management specialist and replays them at a rate specified by the traffic management specialist. The display of old flight data looks exactly like live flight data and can be controlled in all the same ways as live data. Additionally, the traffic management specialist can hold and continue the replay of old flight data.

10.3 Monitor/Alert Displays

The *ASD* displays alerts graphically on the screen overlaid with geographical data and aircraft situation data as directed by the traffic management specialist. The traffic management specialist can turn the alert display on or off using a menu entry, a keyboard command, or a semicolon command. Once the alert display is turned on, the *ASD* automatically refreshes the display every five minutes with the latest alerts until the traffic management specialist turns it off. The traffic management specialist can use other menu entries, keyboard commands, and semicolon commands to control what alerts are displayed and to request other monitor/alert data.

The *ASD* draws alerts on the display as icons. The shape of the icon indicates the alert type: circles for airports, triangles for fixes, and sector boundaries for sectors. The color of the alert indicates the alert status. The *ASD* displays new alerts in red only if the alert threshold is exceeded by the projected *active* traffic demand. If the threshold is only exceeded by including the proposed and/or scheduled demands, the *ASD* draws a yellow icon.

The *ASD* provides additional alert data beyond the icons. The traffic management specialist can examine which 15-minute intervals are alerted using a *time bar* display, a *bar chart* of the projected traffic demands and the alert thresholds by 15-minute interval, a *flight list* report of all the flights that compose the traffic demand at the alerted element, and the current flight positions of the flights which compose the traffic demand for a specified 15-minute interval. (The *ASD* also allows the traffic management specialist to examine the same information for non-alerted elements, which appear in magenta on the display.) The *ASD* draws the time bar, bar chart, and flight positions directly on the map display. The reports are displayed in a separate window, as described in Section 10.4.

Authorized traffic management specialists can turn specific 15-minute intervals of the time bar green. This indicates that the specialist is resolving the alert. Intervals that have been turned to green also have a red or yellow horizontal stripe which indicates that they had been alerted. If all intervals on the time bar are green, the alert icon on the *ASD* will also turn green. The user who turns an interval green will see the effects immediately. Other users at all the other sites will see the effects after the next monitor/alert update.

When the user's alerts are projected for two hours or less, the *ASD* gets the data for the time bar and bar chart displays from the alert updates. However, when the user's alerts are projected for more than two hours, the *ASD* gets such data from the traffic demand database. The *ASD* gets the data for the list report and flight positions from the traffic demand and flight databases. Because the ETMS generates the alert updates every five minutes, while it updates the flight and traffic demand databases continuously, it is possible to see inconsistencies between the time bar/bar chart and the report/flight positions. It is also possible to see inconsistencies between a report and the flight positions, because the ETMS may perform a database update between the time these displays are generated.

10.4 Request Reports

The ETMS produces request reports when requested by a traffic management specialist through the *ASD* or the *TM Shell*. Using the *ASD*, the specialist can enter a request by using the semicolon method.

After making a request via the *ASD*, the traffic management specialist can continue to perform other *ASD* functions. When the ETMS is finished generating the report, the *ASD* notifies the traffic management specialist by displaying a report name icon. The traffic management specialist can view the report in a separate window. The *ASD* continues to show the map display, perform aircraft situation data and/or alert data updates, and interact with the traffic management specialist in the original window. However, the *ASD* reduces the size of the original window to make room on the screen for the report window. When the traffic management specialist is done with the report, the *ASD* closes the report window and returns the map window to its original size.

The specialist can quickly request a report by using the *TM Shell*. The specialist can control the *TM Shell*, the *ASD*, and other *User Functions* independently in separate windows.

The traffic management specialist can request reports containing flight data for a specified airport, fix, sector, or ARTCC. The ETMS processing of a report request consists of two steps: retrieving the data and generating the report.

NOTE: The ETMS user documentation contains a complete description of the six request report commands.

When requested data falls within the previous or next 12 hours, the ETMS retrieves the requested data from the live traffic demand and flight databases. Therefore, within the current 24-hour window, the traffic management specialist can see up-to-date flight data derived from the airline schedules, the NAS messages, the EDCT messages, the flight modeling, and the traffic demand processing. However, because the reports are derived from the traffic demand database, the specialist can see reports only for those airports, sectors, and fixes that are defined as monitored (see section 8.1).

Reports for arrival fixes are processed in a different manner than other reports because the arrival fixes are not monitored elements in the traffic database. The ETMS generates data for an arrival fix by first getting the flights which are arriving at the airport associated with the arrival fix. Once the flights have been retrieved, the ETMS examines each flight to determine which arrival fix it uses. Only those flights which use the requested arrival fix are used to generate the requested report.

When the requested data is outside the current 24-hour window, the ETMS retrieves the data from the airline schedule database; therefore, the traffic management specialist can see only the static values contained there. Also, because the airline schedules do not contain the flight event lists that the live flight database contains, the traffic management specialist cannot get sector and fix reports for flights outside the current 24-hour window.

Once the flight data has been retrieved, the ETMS formats, sorts, and counts the data as requested by the traffic management specialist. The specialist can request several types of reports:

(1) Flight list (LIST) — The traffic management specialist can view a list of flights and associated information and can specify the data items to be shown in the report and how to sort the report.

(2) Flight counts (COUNT) — The specialist can view a count of the number of flights departing from, arriving at, or traversing an airport, fix, sector, or ARTCC and can specify the criteria for counting the flights (e.g., count by airline).

(3) Arrival delay prediction (ARRD) — The specialist uses this report to assess potential traffic congestion.

(4) Fix loading (FIXL) — The specialist can view a count of the number of flights traversing a specified arrival fix or all arrival fixes for a specified airport.

(5) Verify time (VT) — The specialist can observe discrepancies between the *actual* flight arrival and departure times and a specified *time type* (e.g., controlled time). The specialist can specify the flights, time type, and the report format.

(6) List flight plan (LIFP) — The specialist can view lists of flight plans for selected aircraft.

(7) Flight list original (LISTO) — The specialist can view a list of flights and associated information as in a flight list (LIST), except that original times are displayed instead of controlled times for controlled flights that have not departed.

10.5 Weather Displays

The ETMS produces weather reports when requested by the traffic management specialist through the *ASD* using a menu entry or semicolon command. Once the traffic management specialist requests a weather report, the *ASD* suspends normal operation and replaces the map display with two text windows, one showing the surface observations and the other showing the most recent terminal forecast. When the traffic management specialist is finished with the weather report, the *ASD* returns to the previous map display. If there were flight-position or alert-data updates while the weather report was being displayed, the *ASD* performs the display updates when the weather report is exited.

The ETMS maintains a full set of the current terminal forecasts and surface observations as provided by NOAA's Environmental Research Laboratories (ERL). The ETMS maintains the weather data in the encoded form provided by the ERL. When a traffic management specialist requests the weather reports for a particular airport, the ETMS retrieves the latest terminal forecast and surface observations received for that airport and formats a report using standard weather terminology or in the LONG English format.

The ETMS provides an overlay showing the location, direction, and altitude of jet stream winds of 70 knots or more. The ETMS provides separate overlays of radar-determined precipitation levels and labels for them. This information is provided by the ERL and is available through menu commands. NOWRAD6 is another weather feature provided by the ETMS.

The ETMS provides an overlay showing locations of lightning strikes. The ETMS receives lightning information from the ERL and is available through menu commands.

10.6 Display Data Sources

Table 10-1 summarizes the data sources for all the ETMS display data types. The ETMS uses several databases to provide the data displays.

(1) Geographical Data — When a traffic management specialist requests the display of a map overlay, the ETMS accesses the data directly from the workstation's memory and disk.

(2) Aircraft Situation Data — The ETMS maintains a flight database for currently *active* flights on the file server at each site. Every three minutes, the file server generates an aircraft situation data update, which is stored in files on the file server's disk. The ETMS then notifies each *ASD* at the site that a new aircraft situation data update is ready. The *ASD*s read the data from the file server's disk.

(3) Monitor/Alert Displays — The ETMS accesses monitor/alert data from the file server and from the central site. The ETMS gets the data for the basic alert display (showing the alerted locations) from the alert summary data on the file server. The ETMS gets time bar and bar chart data for the next two hours also from the alert summary data on the file server. The ETMS gets time bar and bar chart data beyond two hours from the traffic demands database at the central site. The ETMS gets report and alerted flight data from both the traffic demands database at the central site and the flight database on the file server. The ETMS goes to the traffic demands database for the IDs of the flights that belong in the report or alerted flight display, and then goes to the flight database for the other flight data necessary to generate the displays. The ETMS also maintains the alert threshold (i.e., capacity) data in the traffic demands database at the central site.

(4) Request Reports — The ETMS gets the request report data in four different ways, depending on the time range of the data requested and the type of request. For requests where part or all of the time range is within plus or minus 12 hours of the current time, the ETMS first gets the flight IDs from the traffic demands database at the central site, then gets the other necessary data for those flights from the flights database on the file server, and then gets all necessary data from the airline schedule database for all remaining flights (flights that are not in the traffic database). For requests where the entire time range is beyond 12 hours of the current time, the ETMS gets all the necessary data from the airline schedule database at the central site.

There are two types of requests that allow the user to specify the data source. For Hubsite Requests, the flights database at the central site (rather than the flights database on the fileserver) is used to retrieve the other necessary data for the flights retrieved from the traffic demands database. For schedule database requests, the ETMS gets all report data from the airlines schedule database regardless of the time range requested.

NOTE: The data available in a request report is dependent on the data source. Flights accessed from the airline schedule database will not have any NAS message data (e.g., proposed times, actual times, etc.).

(5) Weather Displays — The ETMS accesses weather display data from the weather database maintained on the file server at each site. Note that the weather graphic displays are retrieved locally, unlike the SAs and FTs, which reside at the central site.

Table 10-1. ETMS Display Data Sources

Data Type	Time Range	Databases	Location
Map Overlay	n/a	Map Overlay	Workstation
Aircraft Situation	n/a	Flight Database	File Server
M/A Summary	+2 hours	Alert Summaries	File Server
M/A Time Bar	+2 hours	Alert Summaries	File Server
M/A Time Bar	>2 hours	Traffic Demands	Central Site
M/A Bar Chart	+2 hours	Alert Summaries	File Server
M/A Bar Chart	>2 hours	Traffic Demands	Central Site
M/A Report	n/a	Traffic Demands	Central Site
		Flight Database	File Server
M/A Alerted Flights	n/a	Traffic Demands	Central Site
		Flight Database	File Server
M/A Capacities	+24 hours	Traffic Demands	Central Site
Request Reports	+ or −12 hours	Traffic Demands	Central Site
		Flight Database	File Server
Request Reports	>12 hours	Airline Schedules	Central Site
Hubsite Requests	+ or −12 hours	Traffic Demands	Central Site
		Flight Database	Central Site
Hubsite Requests	>12 hours	Airline Schedules	Central Site
SDB Requests	n/a	Airline Schedules	Central Site
Weather Displays	current and/or past	Weather Data	File Server
Weather Reports	+24 hours	Weather Data	Central Site
M/A Flight IDs	n/a	Traffic Demands	Central Site
Traffic Mgmt Reports	n/a	EDCT Database	Central Site

Section 11

Processing Overview

Until this point, the *ETMS Functional Description* has described the algorithms used by the ETMS with little concern about how the ETMS is implemented. This section provides an introduction to the system implementation by describing the ETMS processes, what processing they perform, where they are located, what data they use, and what data they produce. This section is only an introduction to the ETMS processing. The *ETMS System Design Document* should be consulted for a complete description of the ETMS design and implementation.

This section presents the ETMS processing in a hierarchical fashion. Section 11.1 groups the ETMS processing into three main categories of functions: the *ETMS Central Functions*, the *ETMS Field-site Functions*, and the *Ancillary Support Functions*. Sections 11.2 and 11.3 describe the types of functions that compose the *ETMS Central Functions* and the *ETMS Field-site Functions*, respectively. Each type of function is further described in a subsequent section: Section 11.4 describes the *ETMS Communications Functions*, Section 11.5 describes the *External Communications Functions*, Section 11.6 describes the *User Interface Functions*, Section 11.7 describes the *Schedule Database (SDB) Functions*, and Section 11.8 describes the *Traffic Model Functions*. Section 11.9 presents the *Traffic Management (TM) Functions*. Section 11.10 presents the *Ancillary Support Functions*. Section 11.11 presents the distribution of the ETMS databases.

11.1 Summary

The ETMS processing functions are organized in three groups, as shown in Figure 11-1.

The *ETMS Central Functions* are real-time, continuously running functions that are located at the Volpe Center and support the entire ETMS network. The *ETMS Central Functions* process the real-time incoming data, maintain a large, distributed database, perform the traffic modeling, and transmit processed data to the field sites where traffic management is performed.

The *ETMS Field-site Functions* are located at every field site where traffic management is performed. The *ETMS Field-site Functions* run continuously. The *ETMS Field-site Functions* maintain local databases of data received from the *ETMS Central Functions* and make the ETMS data available to the traffic management specialist through the *ASD*.

The *Ancillary Support Functions* are off-line functions that are run periodically when data is received. The *Ancillary Support Functions* produce static databases and files for data such as the airline schedule data and geographical data. The *ETMS Central Functions* and *ETMS Field-site Functions* use the static data during the real-time processing. The *Ancillary Support Functions* are run only at the Volpe Center.

The general approach to the ETMS architecture is to perform the processing at the central site and distribute the resultant data to the field sites. Centralizing the processing saves greatly on the computing resources required. Continuously distributing the data to the field sites provides the best response time to traffic management specialist requests for data. However, this approach is not followed rigidly. Some databases have been centralized because of their very large sizes and the relatively low frequency of data access.

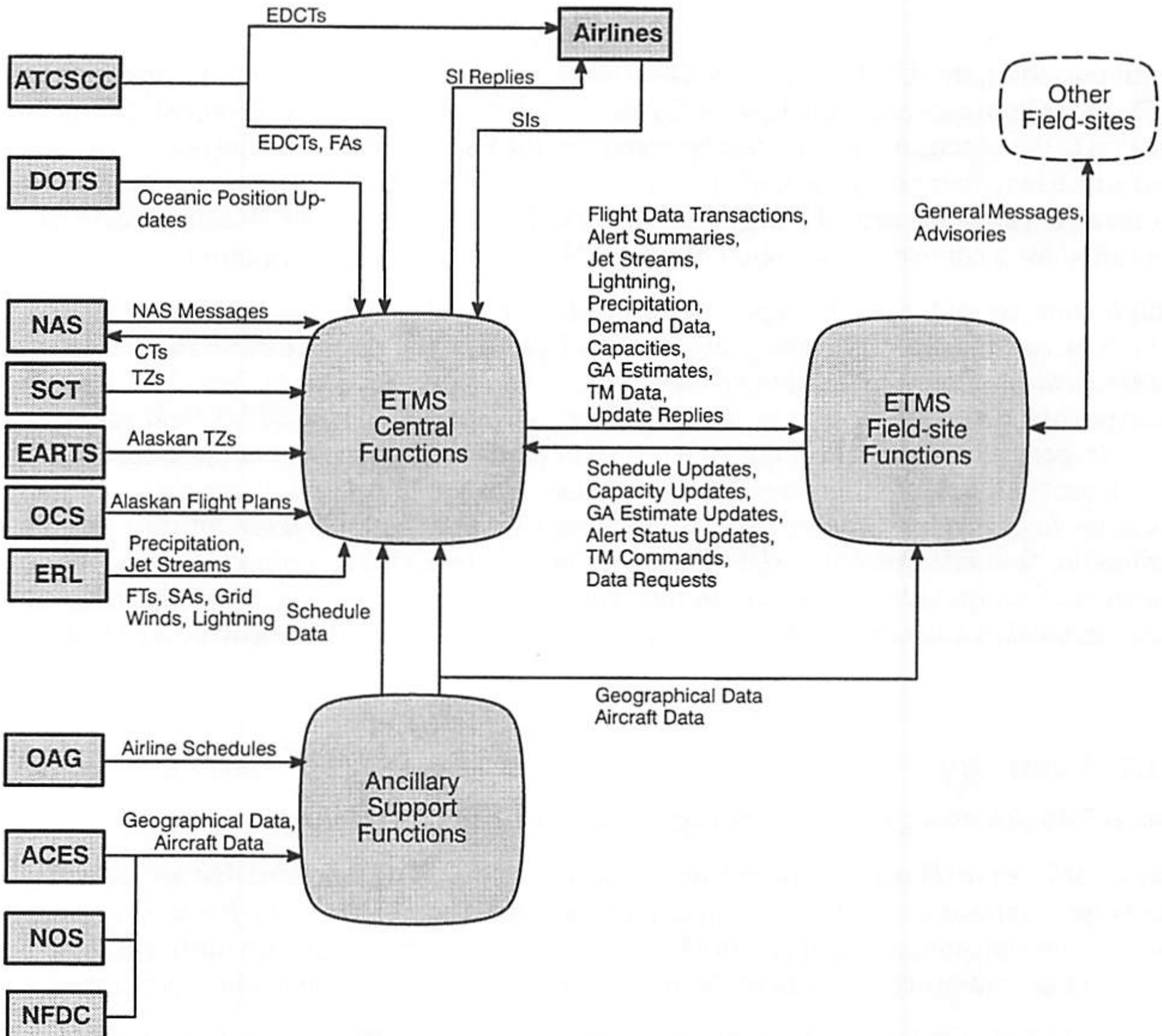

Figure 11-1. Data Flow of the ETMS

The ETMS continuously distributes all data that is pertinent to individual flights for the next 12 hours to each field site as flight data transactions. The ETMS also continuously distributes alert summaries, jet streams, precipitation, and lightning data.

The ETMS maintains all other data only at the central site. This data includes schedule data beyond the next twelve hours, traffic demands for airports, sectors, and fixes, capacities, general aviation (GA) estimates, traffic management data concerning ground delay programs, and weather reports: surface observations (SAs) and terminal forecasts (FTs). The *ETMS Field-site Functions* answer a traffic management specialist's request for this data by requesting the data from the central site. The specialist can also update data on the central site databases. When a traffic management specialist enters capacities, GA estimates, alert status changes, airline schedule updates at a field site, or a traffic management command, the *ETMS Field-site Functions* send the values to the *ETMS Central Functions*, which update the databases and send back a reply.

11.2 ETMS Central Functions

The *ETMS Central Functions* are organized in six groups as shown in Figure 11-2.

The *ETMS Communications Functions* handle all communications between the ETMS central and field sites. The *ETMS Communications Functions* distribute certain types of data automatically to *User Interface Functions* at all field sites; these data are the flight data transactions, alert summary data, jet streams, precipitation, and lightning. The *ETMS Communications Functions* pass commands and data requests entered by traffic management specialists through *User Interface Functions* at field sites to the appropriate functions at the central site; the *ETMS Communications Functions* also send update replies (SAs, FTs, TM data, TM replies, detailed alert data, flight lists, etc.) from the central functions back to the requesting sites. The commands and data requests that are transmitted from the field sites are demand data requests, schedule updates, schedule data requests, capacity updates, capacity requests, GA estimate updates, GA estimate requests, TM requests, TM commands, and alert status updates.

The *ETMS Communications Functions* also handle all message traffic among ETMS components. The components can be located in geographically diverse positions. The *ETMS Communications Functions* implement a Wide Area Network (WAN) form of message switching that utilizes Local Area Network (LAN) technologies wherever possible. Some external interfaces that are not suitable to this unified WAN technology are handled by the *External Communications Functions* described below.

NOTE: Message traffic between components at the same site are shown as not going through the ETMS Communications Functions in Figure 11-2 to maintain the simplicity of the diagram.

The *External Communications Functions* handle communications with systems that are external to the ETMS. The external communications components do not use the standard ETMS communications protocols and therefore require specialized interface converters. The *External Communications Functions* receive NAS messages from the NAS, oceanic position updates (TOs) from DOTS, and Estimated Departure Clearance Times (EDCTs) and Fuel Advisory messages (FAs) from the ATCSCC, weather data from the ERL, and substitution requests (SIs) from the airlines. The *External Communications Functions* send control times (CTs) via the IFCN to the NAS and SI replies to the airlines. The *External Communications Functions* send the NAS messages, oceanic position updates (TOs), and grid winds weather data to the *Traffic Model Functions,* send the EDCTs, FAs, and SIs to the *TM Functions,* and send other weather data to the *ETMS Communications Functions*. The *External Communications Functions* also receive schedule messages from the *SDB Functions* and send them to the *Traffic Model Functions*. The schedule messages are routed this way to include them in the stream of NAS messages sent to the *Traffic Model Functions*.

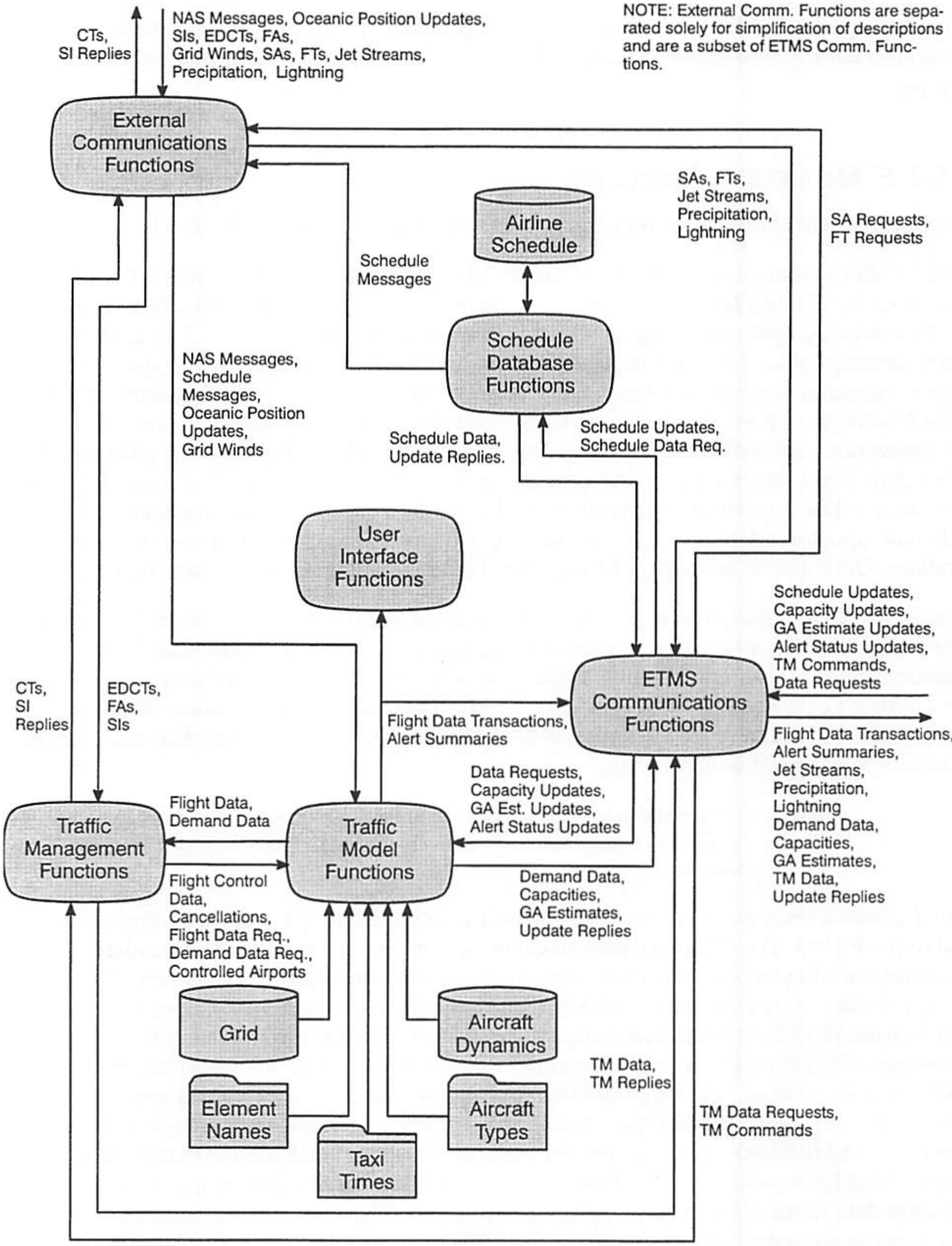

Figure 11-2. Data Flow of the *ETMS Central Functions*

The main role of the *User Interface Functions* is to maintain and display data for the traffic management specialist. This is performed at the field site and is described in Section 11.3. The *User Interface Functions* also exist at the central site to support system monitoring and testing.

The *SDB Functions* integrate the airline schedule data into the ETMS. The *Ancillary Support Functions* (not shown) periodically build an airline schedule database from data received from the OAG. The *SDB Functions* use the airline schedule database to feed *schedule messages* for flights that are scheduled to depart within 12 hours of the current time on the current day to the *External Communications Functions*.

The *SDB Functions* also perform maintenance of the airline schedule database. Traffic management specialists may enter schedule updates to add/edit, cancel, inhibit, or activate scheduled flights through the field-site *User Interface Functions*. The field-site *User Interface Functions* send the updates through the *ETMS Communications Functions* to the *SDB Functions*, which modifies the database accordingly and sends a reply to the specialist. If the database change affects a flight that has already been sent to the *External Communications Functions*, the *SDB Functions* send a schedule message to indicate the change.

The *SDB Functions* also process schedule data requests from field-site *User Interface Functions* to support the request reports. The *SDB Functions* look up the schedule data for the requested time interval, date, and airport; and send the data back to the field-site *User Interface Functions* that made the request.

The *Traffic Model Functions* perform the detailed traffic modeling for ETMS. The *Traffic Model Functions* use the following dynamic data received from the *External Communications Functions*:

(1) NAS messages — messages generated by the NAS computers including flight plans, departures, arrivals, etc.

(2) Schedule messages — messages generated by the *SDB Functions* to feed scheduled airline flights for the next 12 hours into the traffic modeling

(3) Oceanic position updates — messages obtained from DOTS providing position updates for oceanic flights

(4) Grid Winds — data describing the winds aloft

The *Traffic Model Functions* use the following dynamic data received from the *TM Functions:*

(1) Flight control data — controlled departure times from EDCTs and SIs

(2) Cancellations — flight cancellations from SIs

The *Traffic Model Functions* use five static data sources produced by the *Ancillary Support Functions*:

(1) Grid database — geographical data for flight path processing

(2) Aircraft dynamics database — data for modeling flight ascent and descent profiles

(3) Element names file — names of airports, fixes, and sectors

(4) Aircraft types file — definitions of all aircraft designators

(5) Taxi times file — estimated times for taxiing from gates to runways

The *Traffic Model Functions* interpret all incoming data and save them in internal databases. The *Traffic Model Functions* use the input data to maintain a best estimate of the performance of each flight and the traffic demands at each element. The *Traffic Model Functions* send flight data transactions and alert summaries to the local *User Interface Functions* and through the *ETMS Communications Functions* to the remote *User Interface Functions*. The flight data transactions contain data from the incoming messages along with processed flight data that is not directly available from the incoming messages, including the waypoints of the flight path and estimated departure and arrival times. The alert summaries contain summary demand and capacity values for all alerted airports, sectors, and fixes. The *Traffic Model Functions* provide flight and demand data to the *TM Functions.*

The *Traffic Model Functions* process commands and data requests entered by traffic management specialists through field-site *User Interface Functions*. The commands allow a traffic management specialist to update capacities for any airport, sector, or fix; to update GA estimates for any airport; to change alert statuses; to update the schedule database; or to purge a ground delay program. The data requests allow a traffic management specialist to see existing capacities or GA estimates. (A traffic management specialist can cause the *User Interface Functions* to make demand data requests by requesting monitor/alert data and request reports.)

The *TM Functions* implement controlled departure times determined by a traffic management specialist at the ATCSCC by issuing control times to the NAS. The *TM Functions* also apply changes to the control times issued by the airlines according to traffic management policy. The *TM Functions* receive EDCTs, FAs, and SIs from the *External Communication Functions* and receive flight data and demand data from the *Traffic Model Functions*. The *TM Functions* send flight control data, flight and demand data requests, and cancellations to the *Traffic Model Functions* and send SI replies and CTs to the *External Communications Functions*. The *TM Functions* allow a specialist to request control data and cancel flight controls. The *ETMS Communications Functions* send TM commands and data requests to the *TM Functions*, which in turn send back TM data and replies.

11.3 *ETMS Field-site Functions*

The *ETMS Field-site Functions* are organized in two groups as shown in Figure 11-3.

The *ETMS Communications Functions* perform all communications with the central site. All data available to the traffic management specialist through the *User Interface Functions* are transmitted from the central site through the *ETMS Communications Functions*. The *ETMS Communications Functions* send all data received from the central site to the *User Interface Functions*, and all data requests, updates, and TM commands received from the *User Interface Functions* are sent to the central site.

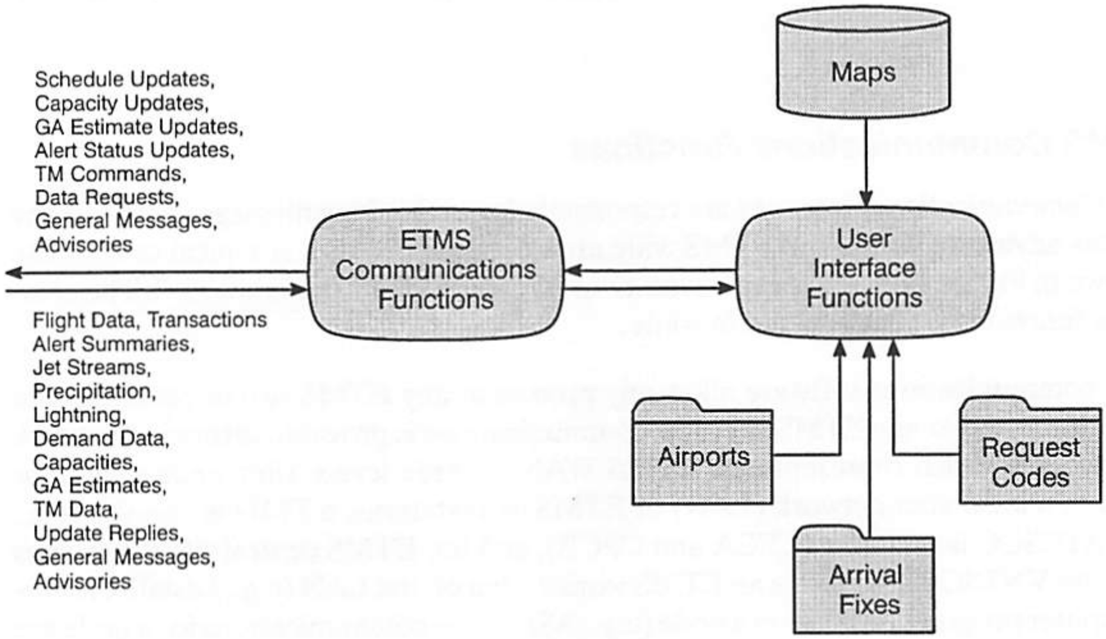

Figure 11-3. Data Flow of the *ETMS Field-site Functions*

The *User Interface Functions* receive flight data transactions, alert summaries, jet streams, precipitation, and lightning automatically; i.e., the *User Interface Functions* do not request these types of data. The *User Interface Functions* maintain these types of data in tables at the field site and make them available to the traffic management specialist through the *ASD*. The *User Interface Functions* request the remaining data from the central site as needed to support a traffic management specialist's data requests. The *User Interface Functions* also use data from the following four static data sources produced by the *Ancillary Support Functions*:

(1) Maps database — geographical data used to draw map overlays on the screen

(2) Airports file — the name and location of each airport, used to *ghost* flights towards their destinations

(3) Codes file — many character codes (e.g., airport names, airline names, and aircraft designators) used to process requests for list/count reports

(4) Arrivals fixes — data used to associate arrival fixes with airports

The *User Interface Functions* provide a variety of graphical and textual displays of the data, as described in Section 2, Functional Overview. The *User Interface Functions* save replay files, which allow a traffic management specialist to replay aircraft situation data for the past. The *User Interface Functions* can support multiple traffic management specialist workstations at each site.

11.4 *ETMS Communications Functions*

The *ETMS Communications Functions* are responsible for exchanging messages between any two processes anywhere within the ETMS wide area network (WAN). A typical central site setup is shown in Figure 11-4, and a representative field site is shown in Figure 11-5. The communications functions are highlighted in white.

The ETMS communications software allow any process at any ETMS site to communicate with any other process at any ETMS site. The communications is provided through a network addressing scheme which represents the ETMS WAN in three levels: *sites*, *nodes*, and *processes*. A *site* is a local area network (LAN) of ETMS workstations; a TMU is one site (e.g., ZBW), the ATCSCC is two sites (CFCA and CFCB), and the ETMS central site is two sites (VNTSCA and VNTSCB). A *node* is an ETMS workstation on the LAN (e.g., //dsk01). A *process* is a computer program running on a node (e.g., ASD). The communications for a node are handled by a single *Node Switch* process. All processes on that node must be connected to the *Node Switch* to use the ETMS communications (Figures 11-4 and 11-5 show the connection of ETMS processes to the *Node Switches*). The communications for a site are handled by a single *Site Switch* process. All nodes at that site must be connected to the *Site Switch* to use the ETMS communications.

A process sends a message to another process by attaching a network address to the message and sending it to the local *Node Switch*. The *Node Switch* determines whether the destination address is another process on the same node. If so, it passes the message directly to the destination process; if not, it passes the message to the local *Site Switch*. Similarly, the *Site Switch* determines whether the destination address is a node at the same site. If so, it passes the message to the *Node Switch* on the destination node; if not, it sends the message to a *Site Switch* at the destination site.

A process does not need to know the exact address in order to send messages or get data. The ETMS communications software provides a set of pre-defined services which a process can address. When a process sends a message to a service, the communications software will find the nearest instance of the service and send the message there. The service might exist on the same node, another node at that site, or at another site.

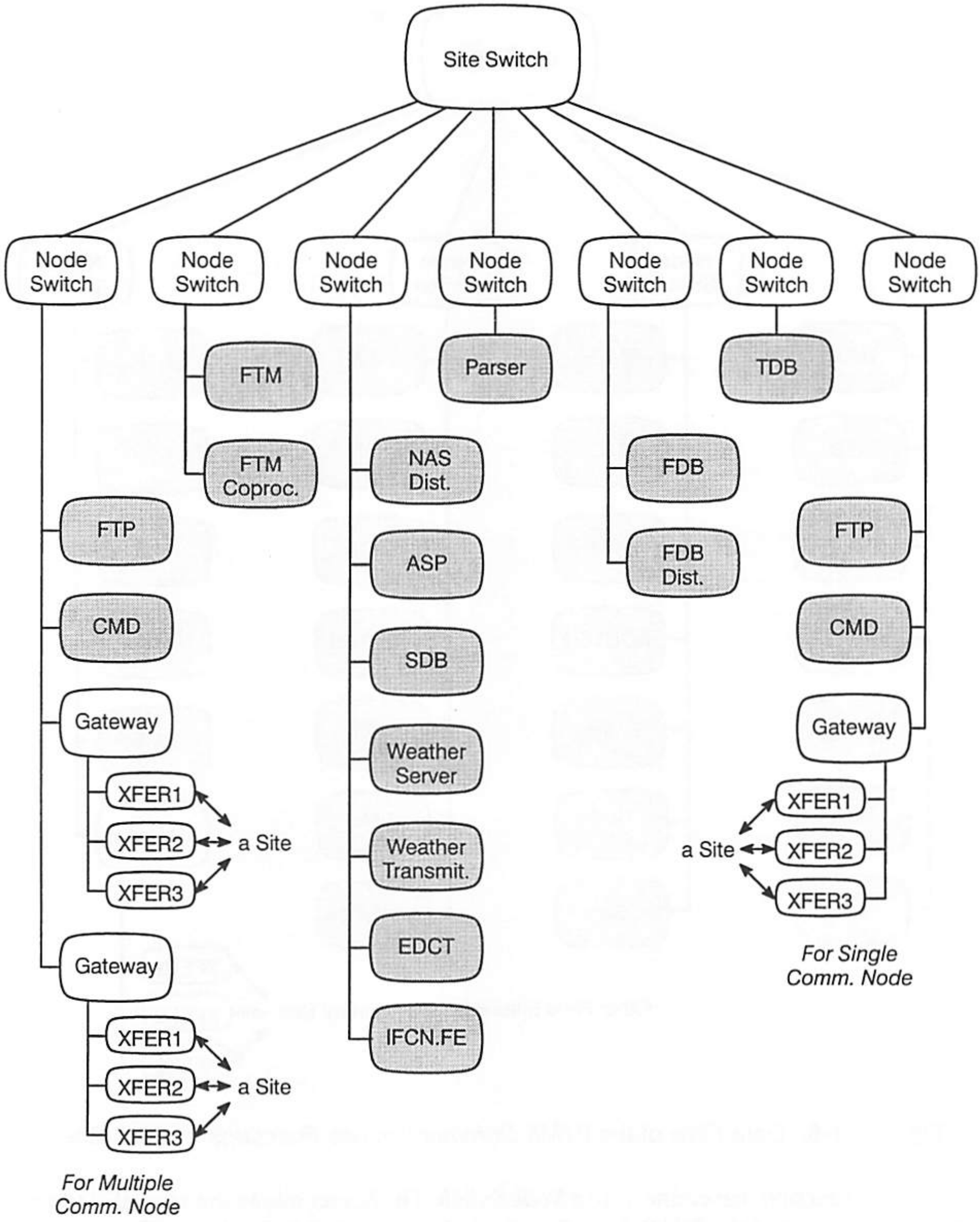

Figure 11-4. Data Flow of the *ETMS Communications Functions* — Central Site

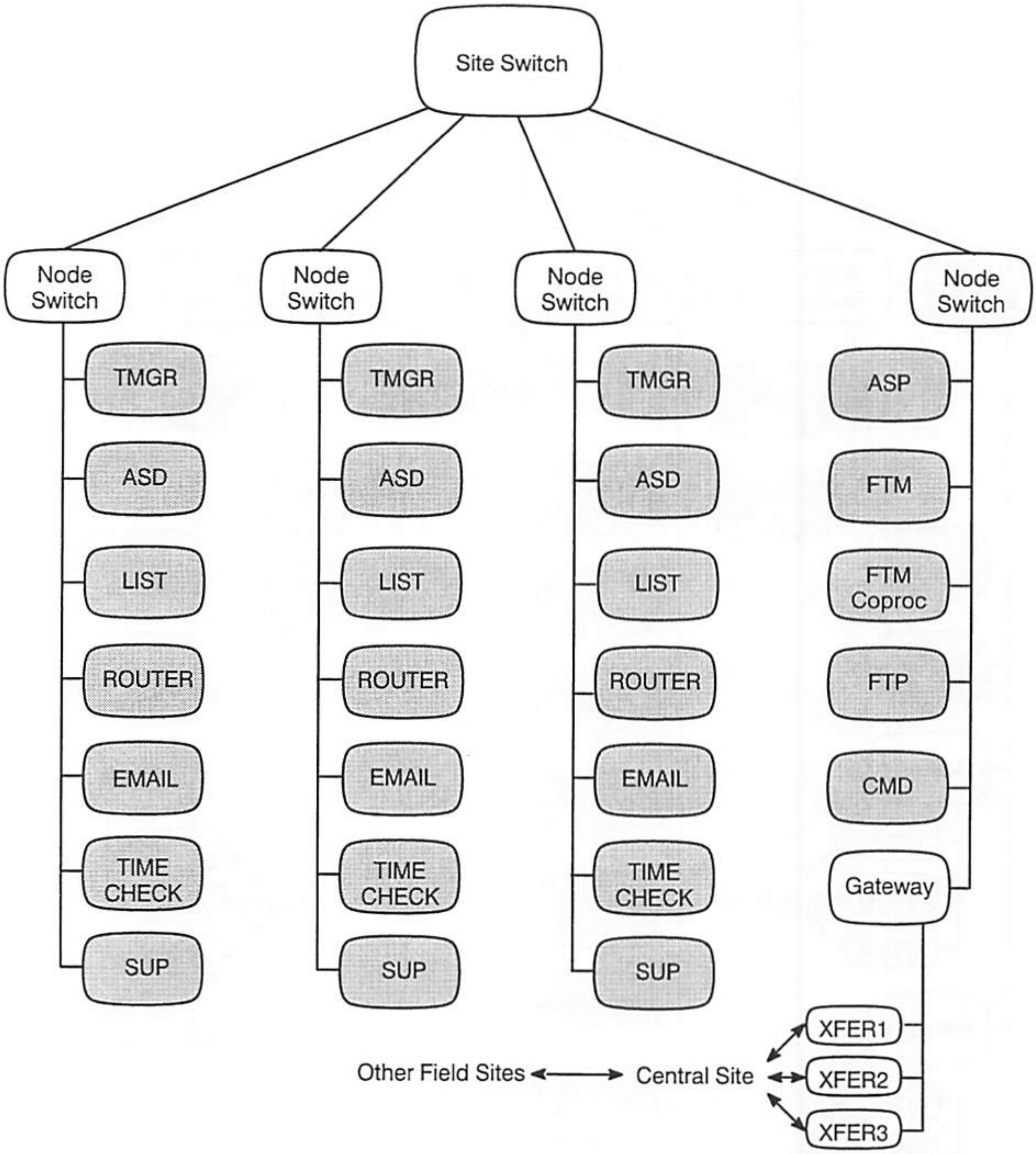

Figure 11-5. Data Flow of the *ETMS Communications Functions* — Field Site

The *Router* is a function that connects to a *Node Switch*. The *Router* allows the specialist to send and receive messages at the *ETMS User Functions*. It is responsible for the notification of all messages sent from the *EMail* function destined for another *Router*, received messages sent from another *Router*, and any error messages encountered. The *Router* is created as a server process, which remains running after a user is logged out. The *Router* function is represented by an icon on the users workstation. Initialization and information messages are defaulted to a low priority. All received messages are set to a medium priority, and error messages are set to a high priority. The *Router* must be connected to a Node Switch to allow *Router* to communicate with *EMail* for message transmission through the ETMS communications software. Since *Router* is used by *EMail* to transmit messages through *ETMS*, *Router* must run on every workstation used to send or receive messages.

The communications path between two sites depends on which sites they are and is determined by the configuration of the physical communications links. Each ETMS field site is connected to the ETMS central site by a satellite link and/or land line. No ETMS field sites are directly connected to each other. Therefore, when the ETMS central site and an ETMS field site communicate, the *Site Switches* can pass messages using a direct connection. However, when a *Site Switch* at an ETMS field site sends a message to another ETMS field site, there is no direct connection. The field *Site Switch* in this case sends the message to the central *Site Switch* which then relays the message to the destination field *Site Switch*.

The ETMS central site communicates with each field site using a dedicated 56 kbps satellite link. A communication *Gateway* process handles the communications for each site. There are two configurations for communications nodes at the ETMS central site (see Figure 11-4). The original design, the *Single Comm. Node*, calls for a one-to-one correspondence, that is, one communications node is reserved to communicate with one field site. The more recent design, the *Multiple Comm. Node*, allows one communications node with special hardware to communicate with two field sites. At the field site (see Figure 11-5), the communications gateway runs on the fileserver node.

There are three asynchronous communications lines (serial ports) attached to each fileserver. They are passed through a statistical multiplexor, which combines the three data streams into the 56 kbps data link. Each line is handled by an *XFER* process. All traffic on a communications link is leveled across the three serial ports by the *Gateway* process; that is, the *Gateway* process evenly distributes messages on the three serial ports and no longer allocates a port per message type.

The types of messages carried on the communications lines from the central site to a field site are flight data transactions, alert summaries, and weather data. A field site typically sends request/reply messages for report data. Users at any site are allowed to send messages and files to users at any other site. Files are restricted to certain directories and file type (ASCII). Security functions allow other files to be transferred under proper authorization.

The communications functions support logistical functions such as file transfer and remote command execution. The remote command execution is used by ETMS operations personnel to diagnose and correct system problems.

A dedicated process, the *Flight Database (FDB) Distribution* process, handles the distribution of database updates from the ETMS central site to the ETMS field sites (see Figure 11-6). The *FDB Distribution* process receives flight data transactions from the central-site *Traffic Model Functions* and sends them to the *User Interface Functions* at the various field sites. The *FDB Distribution* process also receives flight data requests from various field-site *User Interface Functions* and relays them to the central-site *Traffic Model Functions*, which respond in the form of flight data transactions.

Figure 11-6. Data Flow of the *FDB Distribution* Process

11.5 *External Communications Functions*

External Communications Functions handle external components that do not comply with the ETMS WAN packet switching technologies and therefore require a protocol driver and converter to interface with the ETMS. The *External Communications Functions* are shown in Figure 11-7. All communications between the *External Communications Functions* and other ETMS functions are done through the *ETMS Communications Functions.*

The *NAS Data Acquisition* function receives the NAS data from the ARTCCs. The ARTCCs transmit the NAS messages on 2400 bps lines, using a specialized protocol from the 20 ARTCCs to the FAA Technical Center. One protocol conversion unit (PCU) is placed on each of the 20 data lines. The PCU converts the specialized synchronous INTO/INTI data stream into asynchronous ASCII data. This data is split into two 20-line data feeds, each feed going to a DN4500 processor. The *NAS Data Acquisition* process on each DN4500 processor buffers and multiplexes the 20 lines of data. The *NAS Data Acquisition* process sends the multiplexed data from each DN4500 via independent 56KB communications links to the ETMS central site facility utilizing the ETMS communications functions. The *NAS Distribution* processes at the central site combine the NAS messages with schedule messages (received from the *SDB Function*) and TOs (from the *DOTS Parser*). The combined messages are sent to the *Parser* function on each processing string at the ETMS central site. The use of redundant DN4500 and communications links (one on land, the other on satellite) ensures no service interruption from hardware or communications transmission links.

The *IFCN Interface* receives EDCTs and FAs from *Autosend*. Traffic management specialists generate EDCTs and FAs at the ATCSCC using the Selected Controlled Departure Time (SCDT) function and transmit EDCTs and FAs using the *Autosend* function. The *Autosend* function sends copies of the EDCT and FA files to the ARTCCs, the airlines, and the ETMS central site through the IFCN. The *IFCN Interface* process at the central site notifies the *TM Functions* whenever a file is received. The *TM Functions* send out CTs based on the EDCTs and flight data; the CTs go through the *IFCN Interface* function to the IFCN and eventually to the ARTCC host computers.

The airlines generate SIs in response to EDCTs and send them out over ARINC. The *ARINC Interface* function receives the SI messages and sends them to the *TM Functions*. The *TM Functions* generate SI replies, which return back through the *ARINC Interface* function to the airlines, with copies to the ATCSCC printer.

The *ARINC Interface* function also receives oceanic position updates from DOTS over the ARINC line. The *ARINC Interface* sends the oceanic position updates to the *DOTS Parser*, which converts the updates into TOs and sends them to the *NAS Distribution* processes.

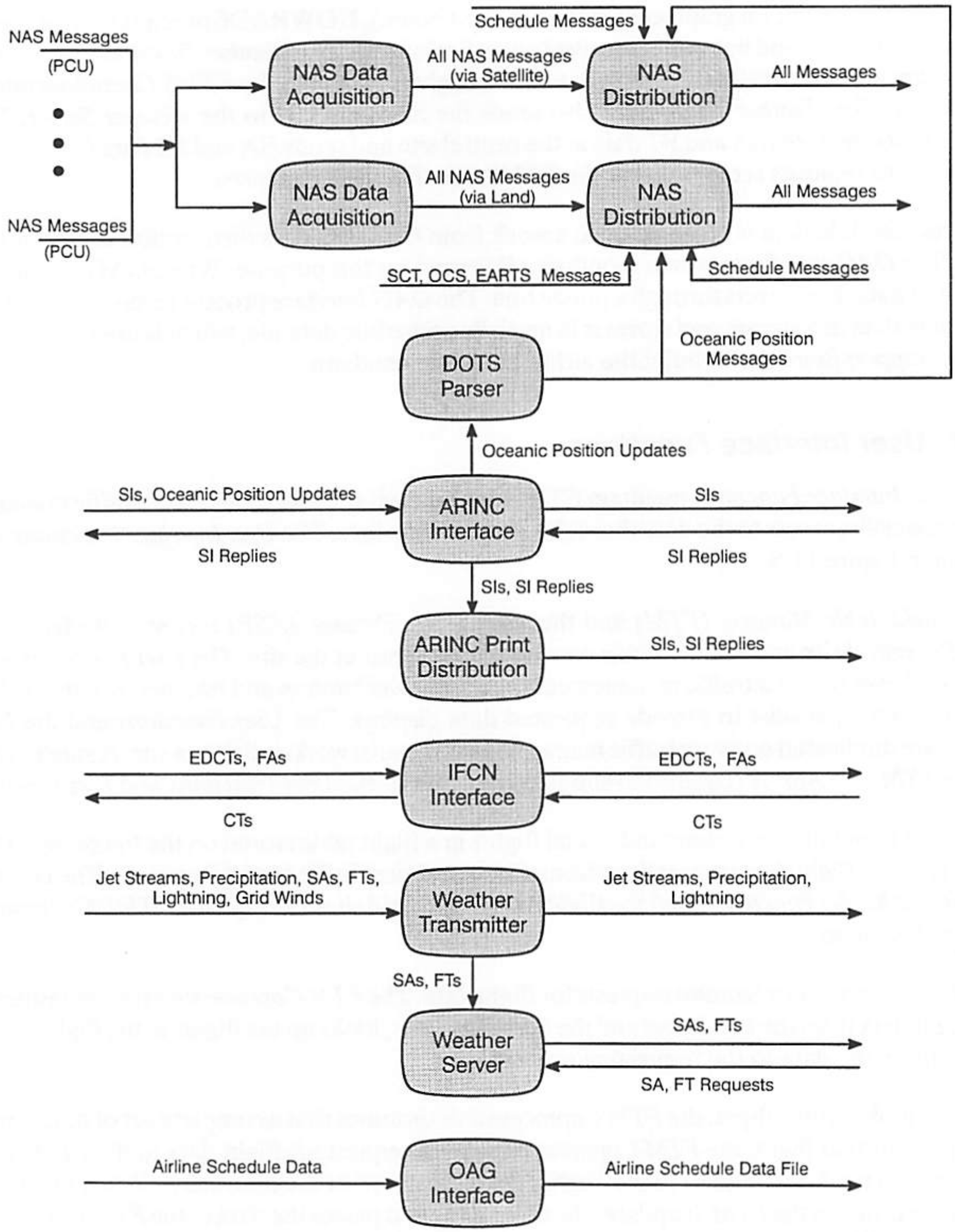

Figure 11-7. Data Flow of the *External Communications Functions* — Central Site

The weather distribution functions include a *Weather Transmitter* and a *Weather Server*. The *Weather Transmitter* receives weather data from ERL in Boulder, Colorado. The ERL data includes SAs (files received hourly), FTs (received every nine hours), grid winds (received every three hours), jet stream graphics (received every 3 hours), NOWRAD6 precipitation graphics (received hourly), and lightning (received every 5 minutes). The *Weather Transmitter* automatically forwards the jet stream, precipitation, and lightning data to the *ETMS Communications Functions*. The *Weather Transmitter* also sends the SAs, and FTs to the *Weather Server*. The *Weather Server* stores SA and FT data at the central site and sends SA and FT data (reports) in response to requests received from the *ETMS User Interface Functions.*

Airline schedule data is received once a week from the OAG data distribution center in Illinois. The *OAG Interface* process monitors a line used for this purpose. When OAG is ready to transmit data, it connects through a phone line. The *OAG Interface* process receives the airline schedule data in a stream and stores it in an airline schedule data file, which is used by the *Ancillary Support functions* to build the airline schedule database.

11.6 *User Interface Functions*

The *User Interface Functions* maintain ETMS data at each site and provide the traffic management specialist access to the data through a graphic interface. The *User Interface Functions* are shown in Figure 11-8.

The *Flight Table Manager* (*FTM*) and the *Alert Server Process* (*ASP*) receive data from the ETMS central site and maintain this data on the *file server* at the site. The *User Functions* and the *List Server* run on a traffic management specialist's *workstation* and interact with the traffic management specialist to provide requested data displays. The *User Functions* and the *List Server* are duplicated on every traffic management specialist workstation at a site. A single copy of the *FTM* and *ASP* is required to support all copies of the *User Functions* and *List Server.*

The *FTM* maintains data about individual flights in a flight table stored on the file server. The *FTM* receives flight data transactions from the central site *Traffic Model Functions*. The central site *Traffic Model Functions* send the flight data to the field sites through the *ETMS Communications Functions*.

The *FTM Coprocessor* handles requests for flight data. The *FTM Coprocessor* receives requests for flight data from the *List Server* and the *User Functions*, looks up the flights in the flight table, and returns the data to the requesting process.

If, while looking up a flight, the *FTM Coprocessor* determines that a complete set of data is not available for that flight, the *FTM Coprocessor* sends a request for flight data to the *FTM*. The *FTM* in turn sends the request to the *Traffic Model Functions* at the central site. When the flight data is returned to the *FTM*, it updates the flight table and passes the data to the *FTM Coprocessor.* The *FTM Coprocessor* will not return flight data to the requesting process until the missing flight data has been retrieved.

The *ASP* maintains the status and demand counts of all alerted elements for the next two hours. The *ASP* obtains the alert data from alert summary files shipped from the central site. (The central site *ASP* sends the alert files to the field sites through the *ETMS Communications Functions.*)

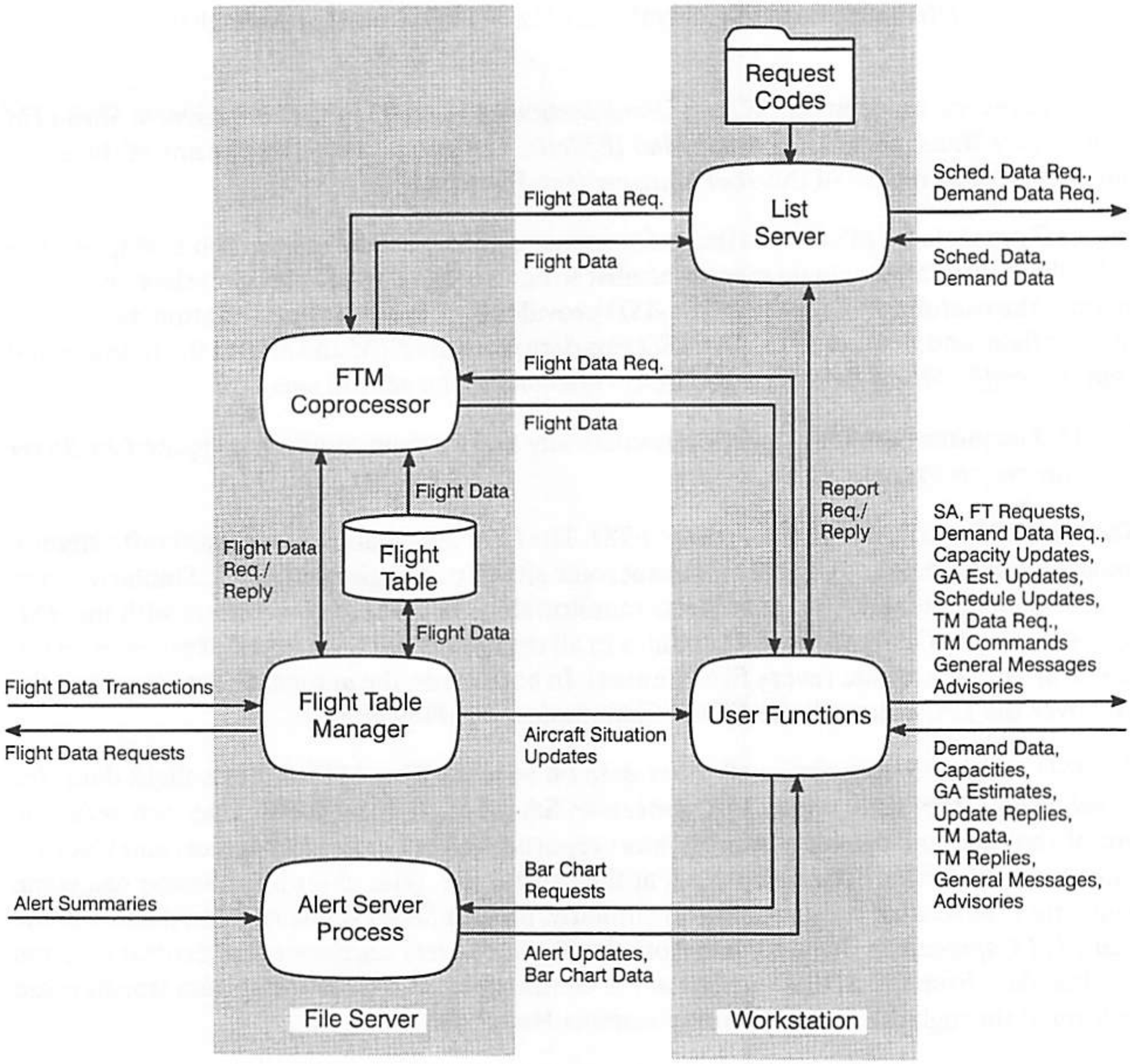

Figure 11-8. Data Flow of the *User Interface Functions* — Field Site

The *List Server* provides flight list, flight counts, arrival delay, time verification, fix loading, and list flight plan reports to the traffic management specialist. A traffic management specialist enters a request for a report through *User Functions*. The *User Functions* passes the request to the *List Server*, which gets the required data, processes the data, and formats the requested report. The *List Server* sends the report back to the *User Functions*, which presents the report to the traffic management specialist.

The *List Server* gets data for a report as follows. By default, if the requested report is for flight data within the next 12 hours, the *List Server* first gets a list of the flights from the *Traffic Model Functions* at the central site. The *List Server* then sends the flight list to the *FTM Coprocessor* which fills in the detailed data for each flight in the list. If the requested report is for flight data beyond 12 hours, or if the user has specifically requested schedule data, the *List Server* gets the data from the *SDB Functions* at the central site. (The *SDB Functions* provide all the data for the report.)

There are several independent ETMS *User Functions*: the *ASD*, *Traffic Management Shell (TM Shell)*, *Delay Manager*, and *Electronic Mail (EMail)*. The specialist can invoke any of these display functions by means of the *Tool Manager* (see Figure 11-9).

The *ASD* presents ETMS data to the traffic management specialist on a graphics display. The *ASD* allows the traffic management specialist to control the data displays in selective ways to enhance the usefulness of the data. The *ASD* provides displays of aircraft situation data, monitor/alert data, and weather data. The *ASD* gets data from the *FTM* and *ASP* at the field site and from the *Traffic Model Functions* and *SDB Functions* at the central site.

The *ASD* acquires data in two ways: automatically and through request/reply; the *List Server* uses only request/reply.

When the *ASD* starts, it registers with the *FTM*. The *FTM* automatically sends aircraft situation updates every three minutes (one minute at some sites) to all registered *ASDs*. Similarly, when a traffic management specialist requests monitor/alert data, the *ASD* registers with the *ASP*. The *ASP* automatically sends alert updates to all registered *ASDs* whenever it receives an update from the central site (every five minutes). In both cases, the *automatic* data is sent to the *ASD* over the LAN through the *ETMS Communications Functions*.

The *ASD* and *List Server* obtain all other data on request. The *ASD* requests flight data (for *alerted flights*) from the local *FTM Coprocessor*, SAs and FTs from the *Weather Server* (at the central site) demand data (for monitor/alert reports and bar charts), capacities, and GA estimates from the *Traffic Model Functions* at the central site. (Bar chart information can come from either the local or the central site.) Similarly, the *List Server* requests flight data from the local *FTM Coprocessor*, demand data from the *Traffic Model Functions* at the central site, and schedule data from the *SDB Functions* at the central site. All request/reply data transfers are performed through the *ETMS Communications Functions*.

The traffic management specialist can update data through the *ASD*. The *ASD* sends capacity, GA estimate, and alert status updates to the *Traffic Model Functions* at the central site, sends schedule database updates to the *SDB Functions* at the central site, and sends traffic management commands to purge ground delay programs to the *TM Functions*. For each update, the *ASD* receives a reply from the selected function either acknowledging the update or indicating an error in the update. The *ASD* sends the updates and receives the replies through the *ETMS Communications Functions*.

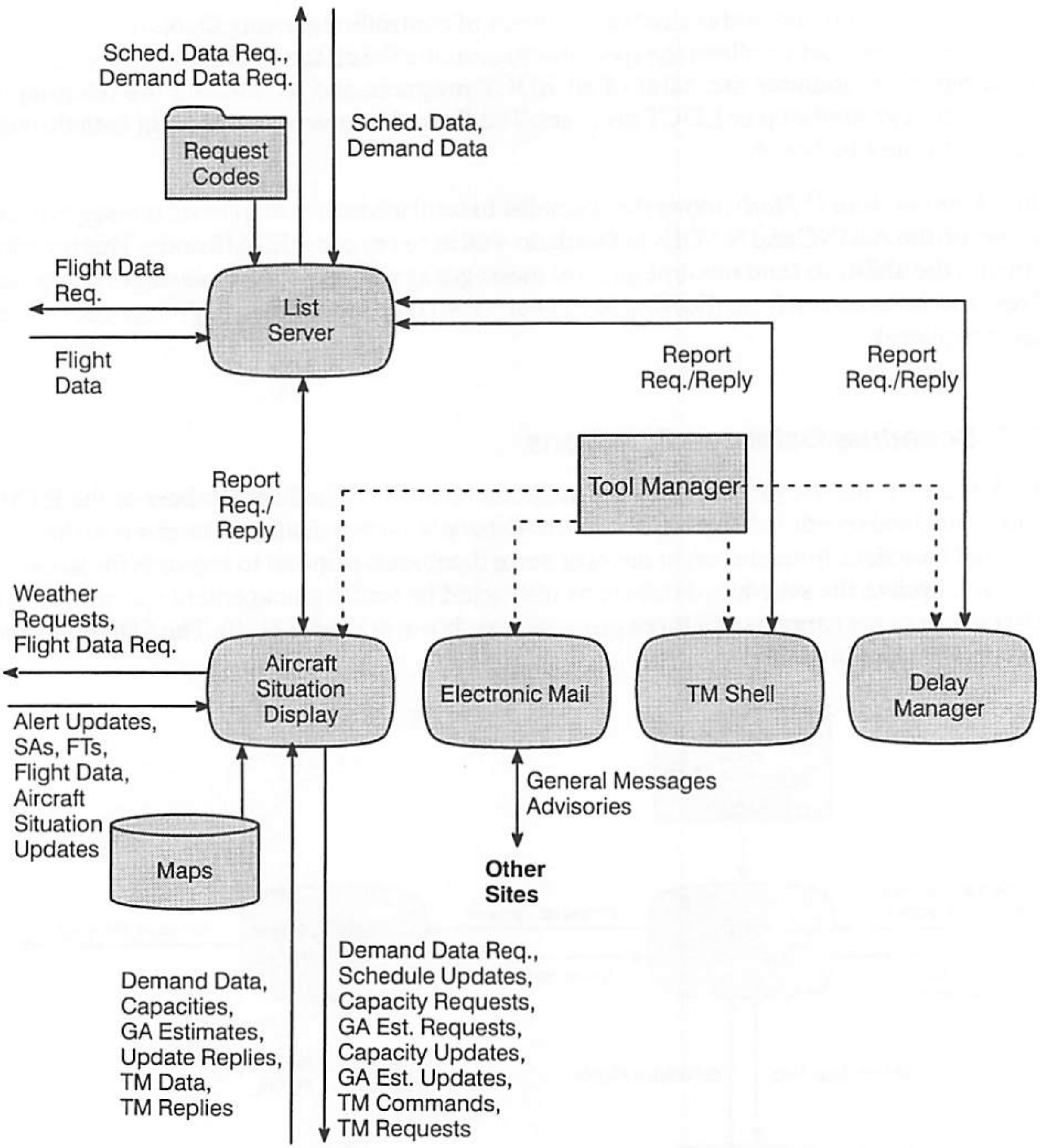

Figure 11-9. Data Flow of *ETMS User Functions* — Field Site

The Traffic Management Shell (*TM Shell*) allows the specialist to execute many of the non-graphical commands available in the *ASD*, without any of the graphical displays (overlays, boundaries, flights, etc.) ever appearing. This gives the specialist access to the same data available in the full *ASD*, while allowing it to be used in a more efficient manner. Using the *TM Shell*, the specialist can issue individual commands and execute script files. *TM Shell* accesses the ETMS data in the same manner as the *ASD*.

Delay Manager simulates and evaluates the effects of controlling arriving flights for any airport you choose. This function allows the specialist to evaluate the effects of implementing a ground stop program, to monitor the status of an EDCT program, and to simulate the releasing of flights from a ground stop or EDCT program. The *Delay Manager* accesses flight data through requests to the *List Server*.

The *Electronic Mail (EMail)* allows the specialist to send advisories or general messages to addresses on the ARINC and NADIN networks as well as to any other ETMS node. This function provides the ability to send one-line general messages as well as detailed messages composed of text and information from files that have been saved (e.g., advisories, flight reports, or message templates).

11.7 *Schedule Database Functions*

The *Schedule Database (SDB) Functions* maintain an airline schedule database at the ETMS central site, feed scheduled flights through the *External Communications Functions* to the *Traffic Model Functions* for inclusion in the near-term databases, respond to requests for schedule data, and update the schedule database as instructed by traffic management specialists. The *SDB Functions* are composed of three processes, as shown in Figure 11-10. The *SDB Functions* exist only at the central site.

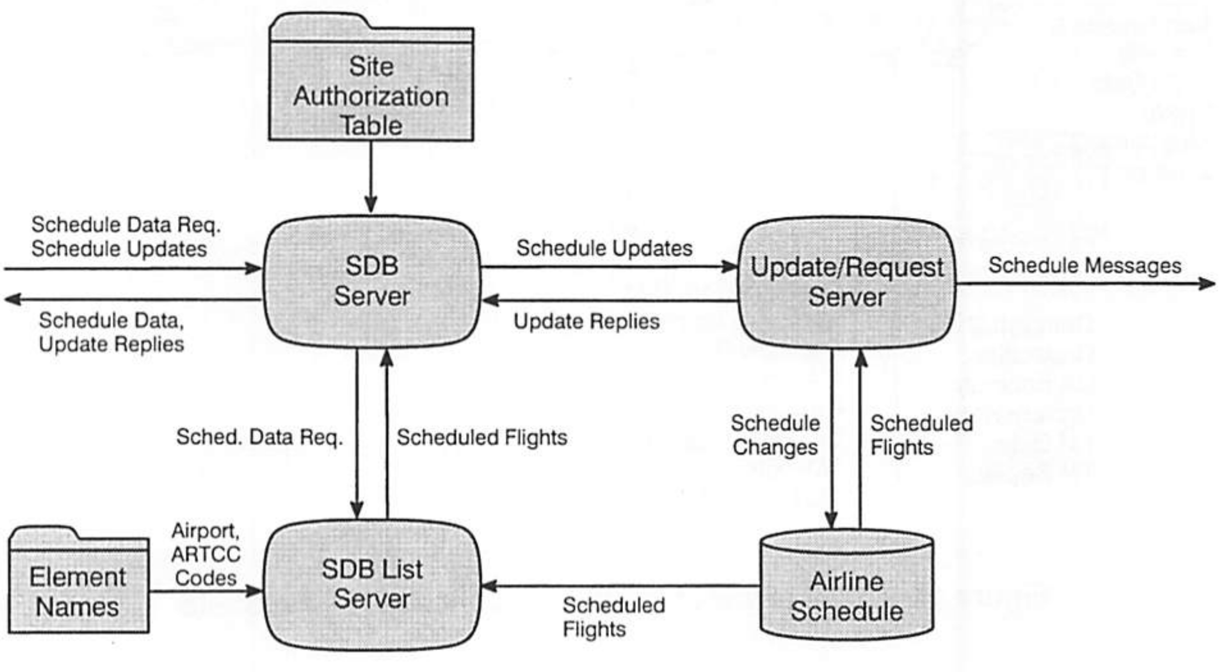

Figure 11-10. Data Flow of the *Schedule Database Functions*

The *SDB Server* handles all of the request/reply communications with other ETMS functions (i.e., all communications except the sending of schedule messages to the central site *User Interface Functions*). The *SDB Server* receives schedule data requests and schedule updates through the *ETMS Communications Functions*. The *SDB Server* sends schedule updates to the *Update/*

Request Server, first checking whether the update is from an authorized site; the *SDB Server* sends schedule data requests to the *SDB List Server*. When the *SDB Server* receives the schedule data back from the *SDB List Server*, it sends the data back to the requesting process through the *ETMS Communications Functions*. When the *SDB Server* receives the update reply back from the *Update/Request Server*, the *SDB Server* sends the reply back to the requesting process.

The *SDB List Server* receives schedule data requests from the *SDB Server*, looks up the requested flights in the airline schedule database, and returns the scheduled flights to the *SDB Server*. The schedule data request specifies airport(s) or ARTCC(s), whether to return arrivals and/or departures for the specified airport(s) or ARTCC(s), the date and time range for the flights, and optionally a filtering ARTCC. If the request is for an ARTCC, the *SDB List Server* returns all flights arriving at and/or departing all airports located within the specified ARTCC. If a filtering ARTCC is given, the *SDB List Server* checks each flight to see if the other end of the flight meets the filtering criteria. For example, if an arrival list is being prepared, the *SDB List Server* returns only those flights whose departure airport is within the filtering ARTCC.

The *Update/Request Server* feeds scheduled messages to the central site *External Communications Functions* and performs updates to the airline schedule database. The *Update/Request Server* automatically sends scheduled flights to the central site *External Communications Functions* when the flights' departure times fall within twelve hours of the current time. The *Update/Request Server* receives schedule updates from the *SDB Server* and makes the necessary changes to the airline schedule database. Schedule updates may cause a flight to be added, cancelled, inhibited, or activated. The *Update/Request Server* sends back a reply indicating either that the update was performed successfully or that an error was encountered. When a schedule update affects a flight scheduled to depart in the next twelve hours, the *Update/Request Server* also sends a schedule message to the *External Communications Functions*, indicating a flight has been added or cancelled.

The schedule messages are patterned to look like NAS messages. To add a scheduled flight, the *Update/Request Server* sends an *FS* message, which looks much like a NAS flight plan message (FZ). To cancel a scheduled flight, the *Update/Request Server* sends an *RS* message, which looks much like a NAS cancellation message (RZ). The *Update/Request Server* enhances the FS and RS messages by including the departure date of the flight. The departure date allows other ETMS functions to distinguish between occurrences of the same flight on different days. The schedule messages are passed to the *Traffic Model Functions* through the *External Communications Functions* so that they appear in the same stream of data as the NAS messages.

11.8 *Traffic Model Functions*

The *Traffic Model Functions* model the performance of flights; project the traffic demands at all airports, sectors, and fixes of interest to traffic management; and generate alerts for those airports, sectors, or fixes whose demand exceeds capacity. The *Traffic Model Functions* are shown in Figure 11-11.

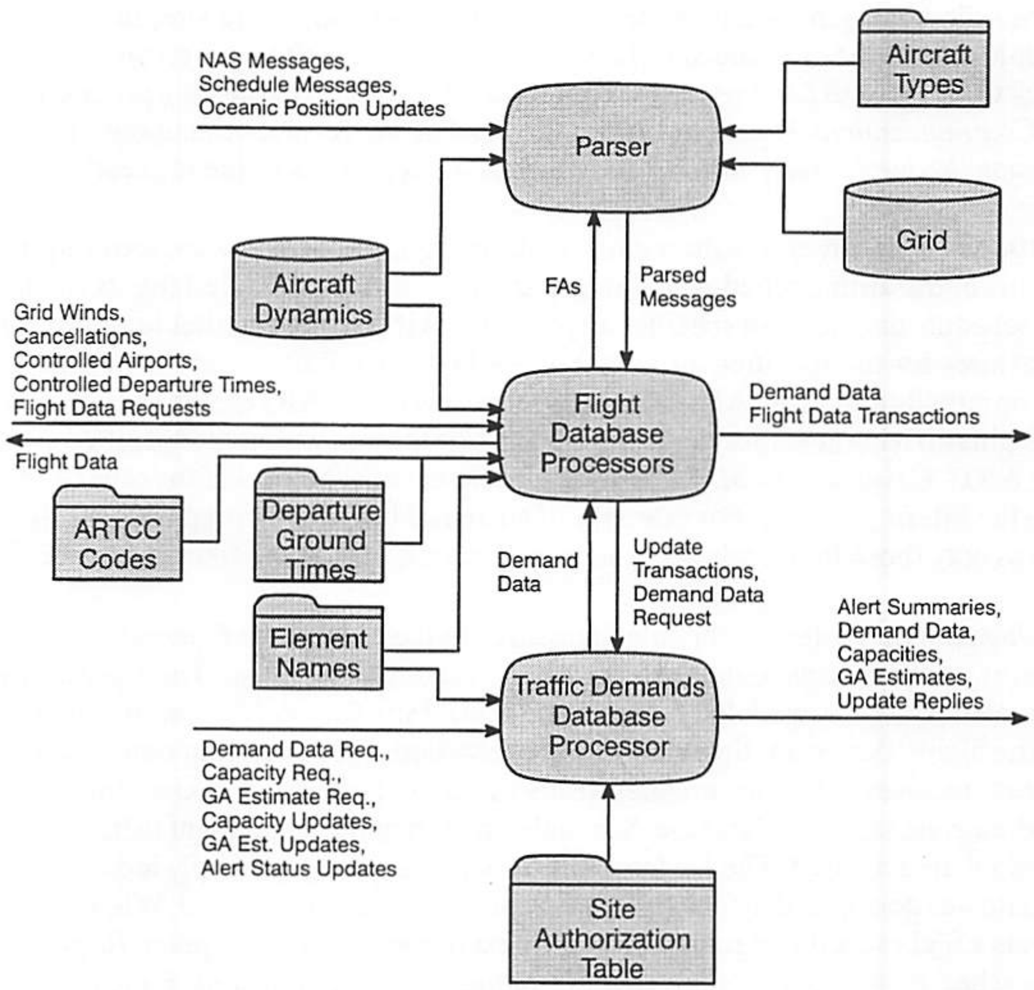

Figure 11-11. Data Flow of the *Traffic Model Functions*

The *Parser* receives the NAS messages, DOTS messages, and schedule messages from the *External Communications Functions*, extracts the contents of the messages, and converts the contents to the internal format required for the subsequent processing. If a flight path (field 10) is contained in a message, the *Parser* interprets the flight path using the geographical data from the grid database and flight profile (i.e., ascent and descent) data from the aircraft dynamics database to produce a detailed, three-dimensional path for the flight. The internal description of the flight path, known as an *event list*, includes all events of interest to traffic management; i.e., airport arrivals and departures, sector entries and exits, fix crossings, airway (i.e., jet or Victor airway) entries and exits, and ARTCC entries and exits. The *Parser* also uses the aircraft types file to determine the *user class* (i.e., commercial, general aviation, military), *weight class* (i.e., small, large, heavy), and *aircraft type class* (e.g., single-engine piston prop) for the aircraft designator in the message. The *Parser* packages all information extracted from the message and passes it to the *Flight Database (FDB) Processors*, which use the data to update the flight databases.

A NAS amendment message (AF) may not contain sufficient data for the *Parser* to produce an event list. For example, if an AF contains a field 10 but not an aircraft type, a new event list cannot be produced. In these cases, the *Parser* sends the data available in the AF to the *FDB Processors* without an event list. *The FDB Processors* then look up the needed data in the flight database and send it back to the *Parser* as a Fuel Advisory message (FA). The *Parser* then models the flight, produces an event list, and sends it back to the *FDB Processors*.

The *FDB Processors* maintain a database of all *active*, *proposed*, *scheduled*, and *controlled* flights for the next 12 hours and previous 12 hours.

> NOTE: Two identical *FDB Processors*, each processing approximately half of the flight data, are used to handle the computation load.

The *FDB Processors* build the flight database using the information received from the *Parser* and the controlled departure times and flight cancellations received from the *TM Functions*. The *FDB Processors* also maintain a database of current and predicted grid winds data extracted from the grid winds data received from the *Weather Server*. The *FDB Processors* use the flight data, the grid winds data, departure ground times, and data from the aircraft dynamics database to predict the event times for the flight. The *FDB Processors* re-compute a flight's event times whenever new data affecting the flight is received. Projected (future) event times are updated to actual (past) flight times as actual data is received in the NAS messages.

The *FDB Processors* send update transactions containing flight events and computed event times to the *Traffic Demands Database (TDB) Processor*, which uses the data to update the traffic demands database. The *FDB Processors* send an update transaction whenever a flight's events or event times change.

The *FDB Processors* also send flight data transactions containing data from the parsed messages and computed data through the *ETMS Communication Functions* to the central and field site *User Interface Functions*, which use the data to update the local flight tables. The *FDB Processors* send flight data transactions automatically for each parsed message received or as a reply to data requested by an *FTM*.

The *FDB Processors* send flight data, either automatically or on request, to the *TM Functions*, which use the data to apply controlled departure times. The *FDB Processors* send flight data automatically when both a controlled departure time (CDT) and a flight plan (FZ) have been received for a flight (causing a CT to be sent by the *TM Functions*). The order in which the CDT and FZ are received does not matter. The *FDB Processors* also send flight data to the *TM Functions* in response to a flight data request.

The *FDB Processors* also support the *TM Functions* in the application of Fuel Advisory messages (FAs). When the *TM Functions* receive an FA, they send a transaction to the *FDB Processors* containing the controlled airport and the controlled time periods. Upon receipt of a controlled airport transaction, the *FDB Processors* send a demand data request to the *TDB Processor*, which returns an arrival list for the controlled airport and times to the *TM Functions* (which apply fuel advisory delays to any flights whose flight plans have already been received). Flight plans received after this time are checked by the *FDB Processors* to see if a fuel advisory delay should be applied to the flight. If so, the *FDB Processors* send a flight data transaction to the *TM Functions*, which determine the amount of delay that should be applied to the flight. The new controlled departure time for the flight is sent back to the *FDB Processors* just as an EDCT would be. Normal EDCT processing then continues in the *FDB Processors* and the *TM Functions*.

> NOTE: The processing of fuel advisory delays is embedded in the *FDB Processors* to ensure that the checking for existing flight plans and the checking for new flight plans is completely synchronized.

The *TDB Processor* uses the update transactions from the *FDB Processors* to maintain the predicted and actual traffic demands at each monitored airport, sector, and fix. The *TDB Processor* keeps counts of events for each element in 15-minute intervals or *buckets*. The *TDB Processor* also keeps the list of flights that comprise the demand in each bucket. The *TDB Processor* keeps separate counts by flight status (*scheduled, proposed, actual*) and can maintain demand data for 24 hours into the past and for up to 40 hours into the future.

When the *TDB Processor* updates a demand count, it checks the demand against the capacity. If the capacity is exceeded by the active flight count, the *TDB Processor* sets a flag, indicating a *red* alert for that time interval. If the capacity is exceeded by the proposed flight count, the *TDB Processor* sets a flag, indicating a *yellow* alert for that time interval. Every five minutes, the *TDB Processor* scans all the elements and generates alert summary files that contain the demand counts for all alerted elements. The *TDB Processor* sends the alert summary files to the local *ASP*, which distributes them to the field sites.

The *TDB Processor* can receive alert status updates entered by traffic management specialists at field sites. The *TDB Processor* first checks whether the update is authorized. If the update is from the ATCSCC, it is always authorized. Otherwise, the *TDB Processor* looks up the element specified in the update in the element names file to find the ARTCC associated with the updated element. The update is only allowed if it was generated at the element's ARTCC or a TRACON within that ARTCC. If an update is authorized, the *TDB Processor* sets the alert flag for the specified time interval to *previously red* or *previously yellow*. Once an alert status has been updated, the *TDB Processor* will not change it back to its previous color regardless of changes in the demand count or capacity value within that time interval. However, if the changes in the demand counts or capacity cause a previously yellow alert to go red, the alert status will be changed to red. After an alert status update is processed, the *TDB Processor* generates a reply, which indicates that the update was allowed or that an error was encountered.

The *TDB Processor* maintains *default* and *today's* capacity and GA estimate values for each element by 15-minute intervals. The default values are pre-defined and cannot be changed by the traffic management specialist. The *TDB Processor* initially sets today's values to the default values each day; it then changes them according to any capacity updates and GA estimate updates received from traffic management specialists. The *TDB Processor* checks the authorization of capacity and GA estimates updates in the same manner as for alert status updates. The *TDB Processor* uses today's capacities for determining alerts. When the *TDB Processor* performs a capacity or GA estimate update, it generates a reply, indicating either that the update was performed successfully or that an error was encountered.

The *TDB Processor* processes requests for demand data, capacities, and GA estimates. The demand data includes the demand counts, capacities, GA estimates, list of flight IDs, and flight event times for the requested element and time intervals. Capacity and GA estimate replies include both the default and today's values for the requested element and time intervals. Demand data requests for ARTCCs, airports, sectors, and fixes are processed by the *TDB Processor.*

If a demand data request is for one or more ARTCCs, the *TDB Processor* looks up all airports located within the ARTCC(s) and produces the demand data for all those airports. An ARTCC demand data request may be accompanied by one or more filtering ARTCCs. If so, the *TDB Processor* assembles the demand data for the requested ARTCCs and passes the demand data with the filtering ARTCC(s) to the *FDB Processors*. The *FDB Processors* look up each flight and keep it only if it meets the filtering criteria (for example, a flight in an arrival list is kept if its departure airport is in a filtering ARTCC). The *FDB Processors* send the resultant, filtered set of demand data back to the requesting site.

11.9 Traffic Management Functions

The *Traffic Management (TM) Functions* process the EDCTs and (FAs from the ATCSCC and SIs from the airlines along with flight data and demand data from the *Traffic Model Functions* to generate CTs to the NAS host computers. The *TM Functions* also maintain a history of all flight control data. The *TM Functions* consist of the *Controls Request Server,* the *Apply Ground Controls Process*, and the *Archive Controls Data Process* and are shown in Figure 11-12.

The *Apply Ground Controls Process* receives EDCTs and FA delays from the ATCSCC and SIs from the airlines all via the *External Communication Functions*. The *Apply Ground Controls Process* checks SIs against rules determined by FAA policy, generates replies, and applies the legitimate SIs to previously received EDCTs. The flight control data (FAs and controlled departure times) are sent to the *Traffic Model Functions*. The *Traffic Model Functions* send flight data back to the *Apply Ground Controls Process* when a combination of a controlled time or delay and a flight plan exists. The *Apply Ground Controls Process* then generates a CT based on the flight data from the *Traffic Model Functions*. The *Apply Ground Controls Process* sends CTs and SI replies to the External Communication Function for transmittal to the NAS, the ATCSCC, and the airlines.

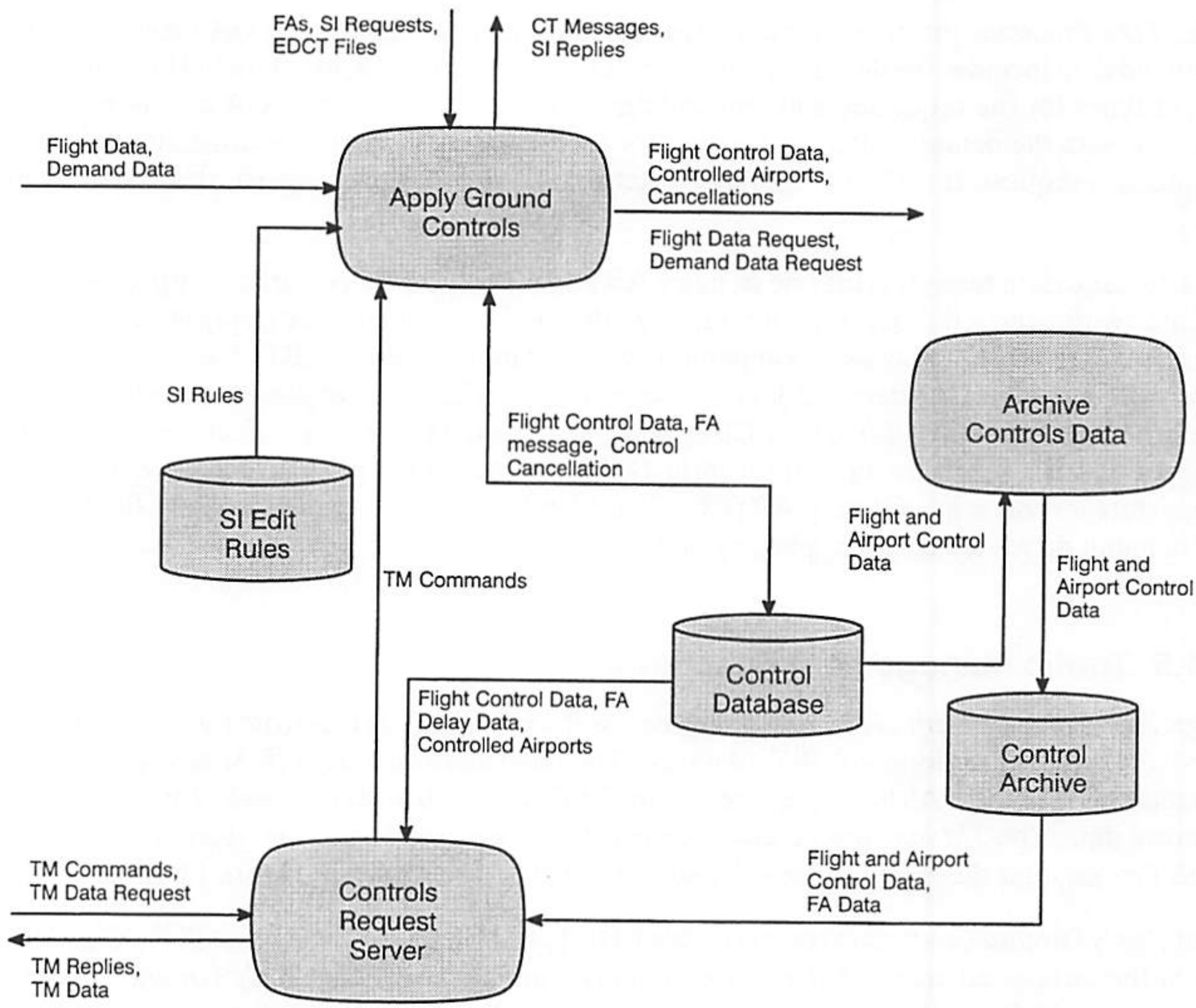

Figure 11-12. Data Flow of the *Traffic Management Functions*

The *Apply Ground Controls Process* also processes the TM Commands: **purge**, **sub off**, and **sub on** received from the ATCSCC (via the *ETMS Communications Function*). When a **purge** command is received, the *Apply Ground Controls Process* cancels previously received controls. The *Apply Ground Controls Process* also sends a reply to the ATCSCC acknowledging the **purge** command. When a **sub off** command is received, the *Apply Ground Control Process* rejects all subsequent SIs for the specified airport(s) until a **sub on** command is received. This process also receives TM commands from the *Controls Request Server.*

The *Controls Request Server* accepts TM commands and data requests from the *ETMS Communications Functions*, retrieves flight control data and FA data from the Control Database, and sends out TM data to the *ETMS Communications Functions*.

The *Archive Controls Data Process* receives flight and airport control data from the Control Database and saves a history in the Control Archive.

11.10 Ancillary Support Functions

The *Ancillary Support Functions* are needed to support the ETMS but do not run as part of the real-time processing. The *Ancillary Support Functions* are used to convert external data into static files and databases used by the run-time ETMS. The *Ancillary Support Functions* are run periodically. There are six *Ancillary Support Functions*, as shown in Figure 11-13.

The *Process Geographical Data* function processes the geographical data received from the NOS, the NFDC, and data files from the ARTCCs. The geographical data includes airport, NAVAID, and airspace fix locations; sector and SUA boundaries; runway layouts; and structured airway definitions.

The *Process Geographical Data* function reads the data from tapes received from NOS, NFDC, or the ARTCCs, extracts the necessary data, checks the data for errors, and processes the data into four specialized forms used by the run-time ETMS:

(1) Grid database — used by the *Parser* to interpret flight paths

(2) Maps database — used by the *ASD Display* to draw graphic display of geographical data on the screen

(3) Element names file — used by the *FDB Processors* and *TDB Processor* to convert element indexes to names before sending to *User Interface Functions*

(4) Airports file — used by the *FTM* to *ghost* aircraft towards their destination airports

The *Process Geographical Data* function also receives OAG airport definitions from the *Build Airline SDB* function. The *Process Geographical Data* function ensures that all airports referenced in the airline schedule database are defined in the maps and grid databases.

The *Process Geographical Data* function provides data to two other *Ancillary Support Functions*. The FAA airport definitions are sent to the *Build Airline SDB* function. The pacing airport, monitored airport, arrival fix, and ARTCC names are sent to the *Build Request Codes* function.

The *Build Airline SDB* function processes the airline schedule data file received from the OAG and produces the airline schedule database used to support the *SDB Functions*. The *Build Airline SDB* function reads the schedule data from the OAG data file, converts it to internal formats, checks for errors, assigns flight paths to the scheduled flights, and builds a set of indexes to the scheduled flights to support the types of access (e.g., by departure time, by airport) required during run time. The *Build Airline SDB* function receives the flight paths from the *Determine Flight Paths* function and assigns them to the scheduled flights based on the city pair and airline. The *Build Airline SDB* function also provides a list of the OAG airline designators to the *Build Request Codes* function and a list of OAG aircraft designators to the *Process Aircraft Type Data* function.

The *Determine Flight Paths* function uses past flight plans (FZs) to derive flight paths, cruising speeds, and cruising altitudes to be assigned to scheduled flights. The *Determine Flight Paths* function analyzes a recent sample (at least one week's worth) of flight plans to determine the most commonly filed flight plan data for each airline between each city pair. The resultant data is sent to the *Build Airline SDB* function for assignment to the scheduled flights. The method for determining flight plan data for scheduled flights is further described in Appendix A.

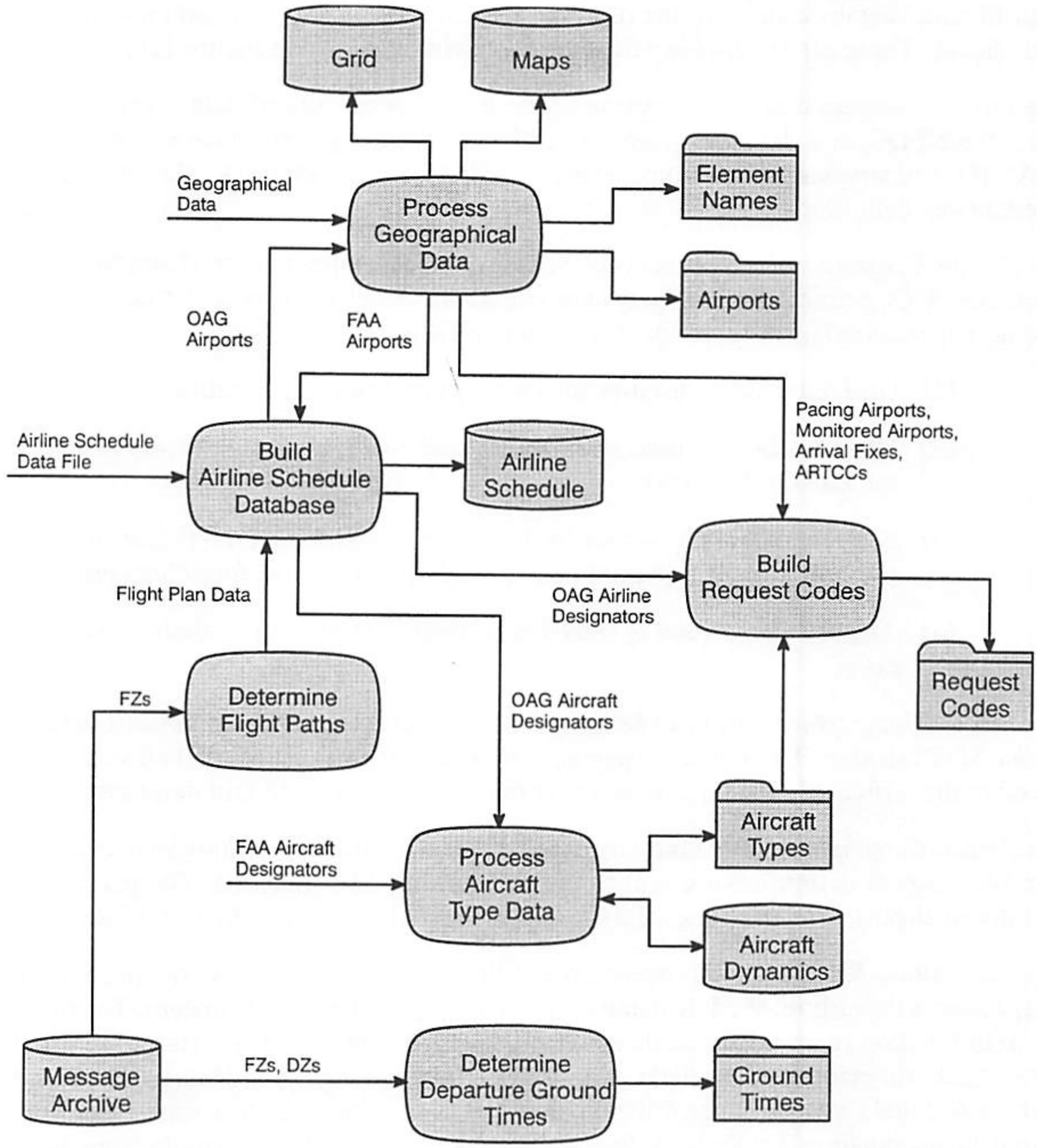

Figure 11-13. Data Flow of the *Ancillary Support Functions*

The *Process Aircraft Type Data* function updates the aircraft types file with new aircraft designators. The aircraft types file contains the user class, weight class, and aircraft type class for each aircraft designator. The aircraft types file is used by the *Parser* to determine the classes for individual flights; it is also used by the *Build Request Codes* function.

The *Build Request Codes* function uses data from several of the other *Ancillary Support Functions* to produce a file of *codes* that can appear as part of a report request made to the *List Server*. The *Build Request Codes* function combines pre-defined, special-use words (e.g., **list**, **count**, **info**) with the following data:

(1) Pacing airport names, monitored airport names, arrival fix names, and ARTCC names produced by the *Process Geographical Data* function

(2) Airline designators extracted from the OAG schedule data by the *Build Airline SDB* function

(3) Aircraft designators from the aircraft types file produced by the *Process Aircraft Type Data* function

The *Determine Departure Ground Times* function derives ground time data from archived historical flight data and is described in Appendix B.

The *Process Aircraft Type Data* function updates the aircraft dynamics database when new aircraft designators are defined by the FAA, or when aircraft designators that are not defined by the FAA appear in the OAG schedule data. The aircraft dynamics database contains a set of pre-defined ascent and descent profiles. When a new aircraft designator is defined, the *Process Aircraft Type Data* function assigns an ascent and descent profile to the aircraft type, based on parameters such as weight class, engine type, maximum ascent rate, and maximum descent rate. The aircraft dynamics database is used by the *Parser* and *FDB Processors* for flight profile and flight time modeling.

11.11 Data Distribution and Recovery

The ETMS system design directly impacts the response time for data displays after a request is made. The response time for a data request is most heavily affected by the physical location of the data.

For the purpose of understanding the response time to a data request, it is sufficient to consider the data to be in three places: the traffic management specialist's workstation, the *file server* at the traffic management specialist's site, and the ETMS central site. Data on the traffic management specialist's workstation can be accessed most quickly, since no communication is required (i.e., data is on a disk or in physical memory). Data on the file server at the site cannot be accessed as quickly for two reasons: the data must be communicated over the *local area network* (LAN) and the file server may be busy performing other functions. Data access from the central site is generally slowest, because the data must be communicated over land lines or satellite links, which operate at much slower speeds than a LAN.

Figure 11-14 illustrates the physical locations of the ETMS databases. The only ETMS data stored directly on the traffic management specialist's workstation is the map overlay data. The flight data used to generate the aircraft situation updates is stored on the file server at each site, as is the alert summary data and some of the weather data (all except SA and FT reports). All other data accessible through the *ASD* is stored on nodes at the central site.

11.11.1 Flight Transaction Messages

Flight data is transmitted from the central site to the field site file server by means of *flight transaction messages*. There are various types of flight transaction messages. These include position messages, route messages, time messages, block/altitude messages, TZ messages, cancel messages, total traffic management (TTM) messages, and critical messages. The last two types, TTM and critical, are used for data recovery.

11.11.2 Field Site Data Recovery

Situations can arise where the flight data at a field site is incomplete. The file server can detect such a situation in two ways. First, when the file server processes are first started up, they may discover that the flight tables are empty or that the flight tables may not have been updated for some time period. Second, when the file server provides flight data for a flight list generated at the central site, it may discover that it is missing data for a flight in the list. In either case, the file server will recover missing data from the central site databases. The file server can request recovery data in three ways:

- A complete recovery is requested when the file server has no flight data.
- A partial recovery is requested when the file server is missing data for a time range.
- A flight data recovery is requested when the file server is missing data for a specific flight or set of flights.

In each case, the requested data is transmitted from the central site. The central site sends the data through the normal flight data transactions that are used to send current flight data.

11.11.3 Central Site Data Recovery

Situations can arise at the central site where one of the redundant processing strings becomes non-functioning. In such cases, a support services specialist can initiate a recovery process, which uses the functioning string as the data source to repopulate the non-functioning string.

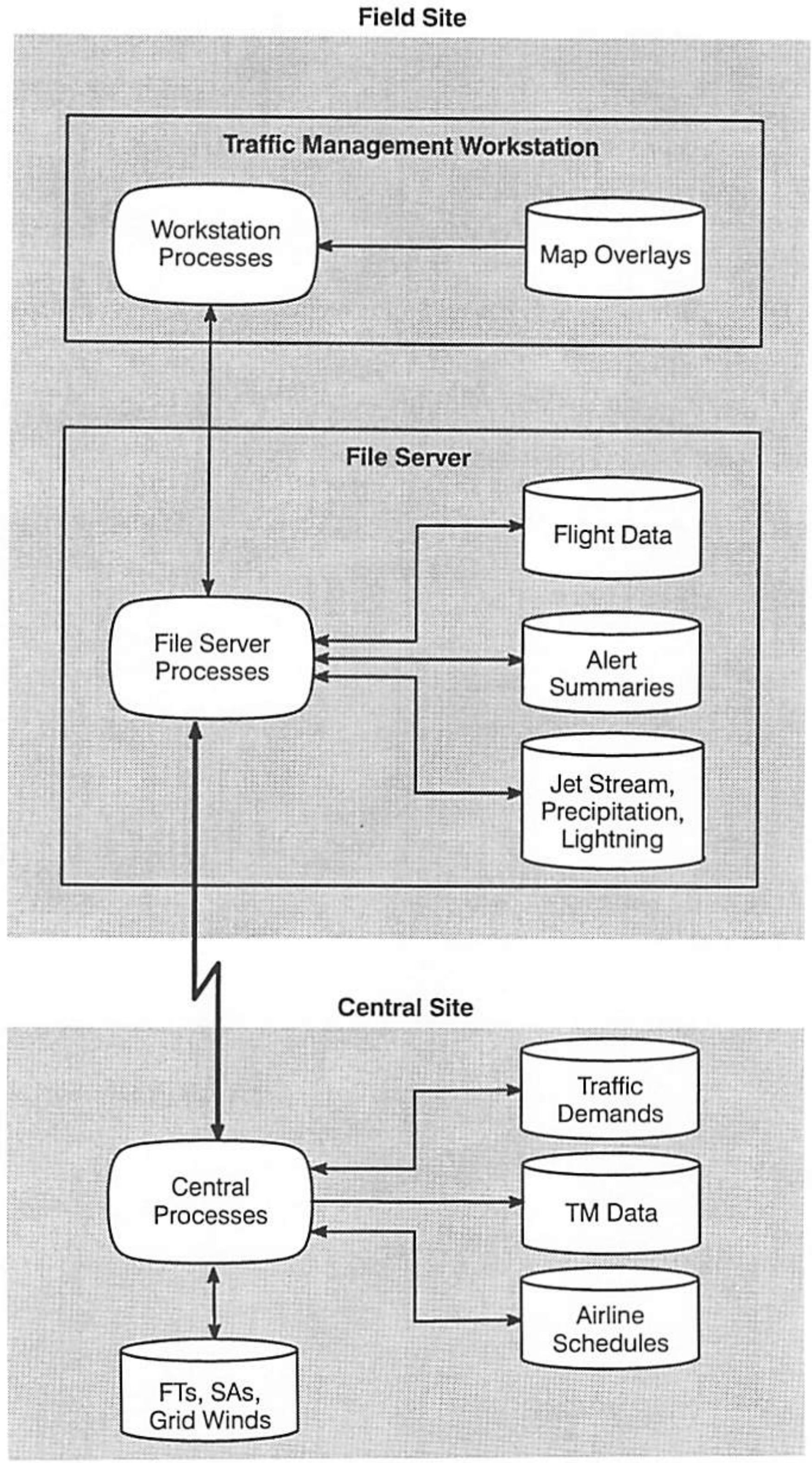

Figure 11-14. ETMS Database Distribution

Appendix A

Determination of Flight Plan Data for Scheduled Flights

In order to model scheduled flights (see Section 7) and to project traffic demands and alerts, the ETMS requires flight paths, ground speeds, and altitudes for flights in the airline schedule database. The ETMS uses this data to process scheduled flights as if a flight plan had been received. Because the OAG airline schedule data does not include flight paths, cruising speed, and cruising altitude, the ETMS must provide flight paths, cruising speed, and altitude for the scheduled flights.

A.1 Summary of Method

The ETMS determines flight paths, cruising speeds, and cruising altitudes for scheduled flights through analysis of historical data. The ETMS archives all flight IDs, aircraft types, origins, destinations, field 10s, speeds, and altitudes from flight plans as they are received. Every week, when new OAG data is processed, the ETMS counts how often each field 10 was filed during the previous two weeks by each airline servicing each city pair. The ETMS also computes the averages of all cruising speeds and altitudes that were filed with each field 10. The ETMS then selects the one or two most common field 10s for each airline and city pair for use with the schedule data. If the ETMS selects two field 10s, it saves the aircraft type associated with the more frequently used route. The ETMS also saves the average cruising speed and cruising altitude associated with each selected field 10 and the aircraft type associated with the more frequent route. When the ETMS feeds a scheduled flight into the live databases, the selected field 10 and the associated average cruising speed and altitude for the city pair and airline are assigned to the flight. If two field 10s are available, the ETMS chooses the route based on whether the aircraft type matches.

> NOTE: The ETMS considers two field 10s to be the same only if the character strings match exactly. It is possible for field 10s to specify the same flight path in many different forms. However, analysis has shown that an airline tends to file a flight path in literally the same form.

A.2 Analysis

The decision to select flight paths, speed, and altitude by airline and city pair was based on analysis of flight plan data. The possible, general factors identifying flight paths, cruising speeds, and cruising altitudes were found to be flight ID, city pair, airline, and aircraft type. City pair is a primary selecting factor for flight path. Cruising speed and altitude may be determined by a variety of factors. Therefore, the flight path analysis focused on identifying the most successful and manageable selection criteria from the following possibilities:

(1) City pair alone

(2) City pair and flight ID

(3) City pair and airline

(4) City pair and aircraft type

(5) City pair, airline, and aircraft type

The cruising speed and altitude analysis considered the same set of combinations plus:

(1) Aircraft type alone

(2) Airline and aircraft type

The analysis first focused on the selection of the field 10. It was assumed that flight ID would be a good selector, but the size of the required database made that choice undesirable (if it could be avoided). A cursory analysis of the use of aircraft type indicated that it was not a good primary selector for two reasons: there is generally not a lot of variation in aircraft types flying commercially between a given city pair, and the types that do fly are well distributed among the observed field 10s.

A quantitative analysis was then performed for the remaining two choices: city pair and city pair plus airline. The success of the selection criteria was measured as the frequency with which the field 10s assigned to scheduled flights by using the criteria were the same as the field 10s later received on flight plans for those flights (i.e., the percentage of flights with the same field 10 when grouped according to the selection criteria). A measurement was made using 4077 flight plans arriving at Denver (DEN) and yielded the following results:

(1) City pair alone was successful for 65% of flights

(2) City Pair and airline was successful for 94% of flights

It was concluded from these results that the additional database size and complexity caused by using airline as a selection criteria along with city pair was justified, and that the use of city pair and airline was sufficiently accurate (eliminating the need to use flight ID) to meet the needs of the ETMS.

Further analysis indicated that a small modification could improve the accuracy even further. For most city pair/airline combinations (91%), the same field 10 was used most of the time regardless of the aircraft type. The remaining airlines showed a tendency to use two field 10s most of the time, one field 10 for certain aircraft types, the other field 10 for other aircraft types. Therefore, the field 10 selection criteria was expanded to include these cases. Those airlines that tend to use only one field 10 are assigned that field 10. The remaining airlines' field 10s are assigned based on matching the aircraft type associated with the more frequently used route.

The analysis of cruising speed and cruising altitude showed expected, general correlation to aircraft type and length of flight. Variations in the filed cruising speeds and cruising altitudes, when observed, were minor. During the field 10 analysis it was observed that the aircraft type was closely correlated to the city pair and airline (i.e., an airline tends to use one aircraft type for all its flights between two given airports). It was therefore concluded that for ease of implementation, the cruising speeds and cruising altitudes would be accumulated according to the same factors as the field 10s. Averages of filed cruising speeds and cruising altitudes have been used to smooth out any variations due to other factors (e.g., weather, late departure, etc.).

Appendix B

Determination of Departure Ground Times

The ETMS requires a ground time estimate to project the *wheels up* time from the *scheduled* or *filed gate pushback* time (see Section 7). Some flight plans contain wheels up times and others contain gate pushback times. The ETMS applies ground time to gate pushback times to resolve that discrepancy. This appendix describes how the ETMS ground times are determined.

B.1 Summary of Method

The ETMS derives ground time data for the pacing airports from archived historical flight data, using past flight plans (FZs) and departure messages (DZs). Proposed departure times from FZ messages are subtracted from actual departure times from DZ messages to determine departure delays. Departure delays are accumulated separately for each airline operating at each pacing airport. Statistical criteria are applied to the resulting data sets to determine whether they are suitable for deriving ground times. For the resulting delay data sets, the median values are found and used as ground times by the ETMS. For airline–airport combinations whose data fails the statistical criteria or for which data are not accumulated, a default value (currently 10 minutes) is used for ground time. Median Ground Times are stored in a file accessed during flight modeling by the ETMS. The Median Ground Time file is updated weekly, using data from the immediate past three weeks.

B.2 Analysis

The empirical approach to the determination of the ground time values was selected because of the availability of the NAS message data. The decision to separate delays by airport and airline was made based on general knowledge of airport ground operations and the lack of specific gate assignment data. The delay data showed enough differences in ground times for different airlines at the same airport to justify the use of airline as a distinguishing factor.

Previously, ETMS used an estimated taxi time rather than an estimated total ground time. Taxi time was intended to represent time required to move the aircraft from the gate to takeoff position and did not take into account additional elapsed time between scheduled pushback and takeoff such as gate hold. To improve the quality of takeoff time predictions, as well as of downstream event time predictions, taxi time was replaced by total ground time in the ETMS model. The same archived departure messages data sets are used, but instead of a delay bias being extracted from each dataset, a median delay value is obtained. The median value was chosen as the most stable measure of the total ground time. Because the median is much less sensitive to individual data values than the taxi time, the statistical criteria are less stringent. The departure delay data set is now subject to the following steps:

- Minimum number of flights must equal or exceed ten. This reflects a requirement that there be a minimum sample on which to base an analysis.
- Datasets must satisfy cohesiveness. To eliminate "outliers", unrepresentative extreme values of data, "tails" of the data distribution are removed: the highest 5% and lowest 5% of the delay distribution are eliminated, as are values exceeding an unrealistic, large value. The remaining distribution must meet the condition that the fraction of empty histogram bins not exceed a limit. Because the median is far less sensitive to data characteristics than was the departure delay bias, the maximum allowed delay (above which data are excluded) is 80 minutes and cohesiveness is 60%. (By comparison, for the taxi time computation, the values were 30 minutes and 40%.) For distributions which satisfy cohesiveness, the median delay of the entire dataset is determined and added to a lookup table for use in ETMS.
- A resulting negative median value of departure delay is replaced by zero.
- The default delay, used when no statistical determination is available, is 10 minutes.

Ground times are derived from past NAS messages (FZs and DZs). The NAS messages are archived continuously. Once weekly, the current week's messages are saved and the archiving process restarted. The current week's data is appended to the previous two weeks' data and the three weeks' data used as input to a new ground time determination. This allows new airlines and new departure characteristics to be reflected, while retaining enough data to support statistical analysis.

Index

A

B

C

D

E

F

G

I

L

M

N

P

R

S