MTR 98W0000120

MITRE TECHNICAL REPORT

NAS Flight Data Model

September 1998

Ronald A. Schwarz
Catherine N. Bolczak
Dr. Nels A. Broste
Carl E. Dahlke
Sally F. Stalnaker
Long K. Truong

Sponsor:
Dept. No.:

MITRE
Center for Advanced Aviation System Development
McLean, Virginia

Abstract

Data about flight is critical to the operation of the NAS and to the development of application systems that use flight data and flight data structures. Historically, each application system has developed its own view of flight data and its structure. This document reviews a number of flight-based activities in the NAS. It then describes a flight data model, created with the ERwin relational data modeling product. The model is proposed as a common view of flight data to improve flight data processing and the sharing of flight data among application systems and among operational decision makers in the NAS.

Appendices contain the details of the data model produced from reports of the ERwin flight data model.

Executive Summary

Background. The Air Traffic Management (ATM) Integrated Product Team (IPT) (AUA-500),), now known as the Free Flight Phase 1 program, has recognized the need for data sharing from both within the ATM IPT systems and programs (extensible to NAS service providers) and with the user community (e.g., air carriers, the military, general aviation, and international aviation). There are significant cost savings and operational efficiencies to be gained if the information flow between NAS systems can be enhanced in ways that foster data standardization and reuse. In addition, the System Development organization (ASD-100) has been supporting data standardization efforts as a prerequisite to more efficient, cost-effective system development and operation.

These two FAA organizations have joined forces in establishing the NAS Information Architecture Committee (NIAC), a cross-cutting organization that is supporting the work described in this document and that is now supported by numerous additional FAA organizations. Interest in a flight data model now spans every IPT whose systems manage flight data; NAS users, who require flight data as part of their operations; and the development community that is building applications requiring a flight data structure.

The exchange of flight data and NAS status data is also of considerable interest to the NAS user community. The Collaborative Decision Making (CDM) program, with original RTCA sponsorship, is a cross-cutting effort to develop a capability to improve operational decision making through more timely, accurate, and consistent exchange of NAS data. In addition, input about flight management needs were taken from RTCA Special Committee 169 Working Group 5. In addition, the AUA-500 R&D program and the ATS/ASD operational concept both address flight planning and flight data issues. CAASD has participated in both of these activities.

Historically and naturally, every organization and application that required flight information independently created a structure for these data. That has led to the situation today in which there are many variations of a structure for flight data, each with its own tailored terms, definitions, formats, and codes.

Based on AUA-500 requirements, an effort was started in FY97 to define a common data management environment for TFM systems. Several application systems were analyzed to identify TFM data requirements. Because the focus of this activity was on data elements shared across the interface among systems, those elements internal to a program were not considered; therefore, only those data elements that are inputs to or outputs from a particular program were included. Data elements were identified through existing documentation, including NAS-MDs, system specification and requirements documents, and other technical sources.

One result of that effort is a set of proposed NAS data categories, shown in Table ES-1, of which category 1 is demand, or flight data. It is that set of flight data which is the scope of this analysis and is the basis for the flight data model described in this document. Additional categories of NAS data must also be modeled and established as system requirements.

Table ES-1. Proposed NAS Data Categories

Category	Description
Demand	Flight data, including information such as the flight itinerary, flight identification, flight planning, flight events and status, and ATM control events that affect a single flight.
Capacity	Resource data, including adaptation data, describing relatively static resources, such as airports, runways, and airways, as well as their dynamic status, e.g., configuration, current capacity, and activation
Weather	Data about weather including terminal and airborne weather observations, forecasts, and reports of weather phenomena
Traffic Management	Data that describe situations in which capacity exceeds resources, and actions taken by ATC, TFM, and NAS users to resolve these imbalances.
General Resources	Data that are not NAS-specific, such as time, geography, and geopolitical data
Performance	Data used to describe NAS operational effectiveness and their ability to meet user needs
Miscellaneous	Data elements that do not fit under the other major categories. Conceivably, new major categories could be identified and defined from these elements.

Purpose. The purpose of this document is to describe an organization for flight data in a flight data model. It is the purpose of the model to capture the essence and the breadth of flight information and invite comment. The model has the flexibility to change in scope and direction and can be easily modified to accommodate additional views and events.

It is not the purpose of this flight data model to become an absolute standard — relevant for all contexts and for all users in exactly one shape and form. Rather, the model is intended to be a common but flexible guide that should help to provide the benefits of commonality, where appropriate, while allowing extensions and customization when and where those are justified. It is expected that the flight model will change continuously over time to keep pace with new operational requirements. The use of automated tools, such as database modeling software, makes the process of updating the data model manageable. The output of the database modeling tool supplies a database management system (DBMS) with the structure it needs to implement the model in a database.

A data model is a formal representation of a related set of information. In this case, a logical data model (LDM) has been designed using the entity-relationship (ER) format in Platinum Technology's ERwin (version 3.5) database design software product that is built on the IDEF1X data modeling standard.

The flight data model has these primary goals:

- **Data sharing:** To be used as a standard view of flight data by applications as a fundamental step toward improved data sharing among systems sharing flight data and between systems and NAS users, whether flight data is passed in message format, as the result of a query, or via a database access over an intranet or the internet.

- **Application development:** To be a standard view of flight data for use in application development and system re-design. This includes establishing standards for key flight data concepts, such as coordinate position and time.

- **Information management and data analysis:** To enable better information management for flight data, including its events and corresponding status information, so that real-time flight data as well as post-event historical data can be better managed and analyzed, typically within COTS data management tools, such as database management systems (DBMSs), data warehousing products, and query tools.

Why produce a flight data model when many NAS applications already use flight data? It is exactly this issue — that these numerous applications tend to establish their own 'local' view of flight data. Invariably, each application system tends to define flight data differently from other applications. The result is that system interfaces, such as message passing, cause additional pre-processing for the receiving system, at best, and ambiguity or error in interpreting the incoming data, at worst. Such an environment also complicates data sharing across the FAA-to-NAS user boundary and is expensive to maintain and coordinate.

Flight Plan Processing for Free Flight Phase 1. With CAASD support, the RTCA has developed an operational concept for Free Flight Phase I, the FAA's near-term program to provide user benefits by 2002. This concept addresses a wide range of issues, among them are two below that involve data management and, within that, flight data management. As quoted from the FFP1 Operational Concept,

> ***"Technical and Information Interdependencies:*** This area is critical to the evolutionary development process because it highlights the arrangements, interactions, and interdependencies of the FFP1 capabilities as a common set of systems and services. This requirement is judged to present the greatest challenge for FFP1 implementation. The technical and information architectures of the

independent systems will be brought together into a cohesive array of features delivering increased enhancements and performance to the NAS."

The flight data model is a component to be used to standardize the view of flight across FFP1 and across the NAS. It contains new content and a proposed structure for flight-related data. The technical aspect of this issue is how flight information will be deployed, made accessible, protected, and used in system development.

Flight Data Model. The flight data model has its origin in recent work for the AUA-500 organization in developing a common data environment for TFM. This work lead to the creation of an inclusive set of NAS data categories, of which flight data was one. This work also included the analysis of several key TFM systems and the mapping of the data requirements to each of the data categories. In this way, all of the flight data per system was discovered and associated with a flight attribute from the flight, or demand, category. This mapping was used to create a proposed standard set of data elements per category and a data model, as in the case for flight data, to define the relationships among the data and as a requirement for the exchange of such data at the interfaces between systems.

Data Requirements. Staff from CAASD compiled the data attributes for the TFM applications and organized them by application program and by the seven data categories described above. Table ES-2 displays the element count by application program. Each of the elements referenced in the table was named and described in a Microsoft Access database. For this analysis, only the flight data elements (category 1: demand) have been addressed. It is anticipated that additional modeling efforts will cover the other categories, with a focus on capacity and NAS resources (category 2: capacity). These data were subsequently organized into common NAS data categories, as described in the next section.

Table ES-2. Application Programs Element Count

System Acronym	System / Program Name	Data Element Count
NFDC/ AIS	National Flight Data Center/ Aeronautical Information System	1159
CDM	Collaborative Decision Making	53
CTAS/ TMA	Center TRACON Automation System/ Traffic Management Advisor	81
CTAS/ pFAST	Center TRACON Automation System/ passive Final Approach Spacing Tool	97
DOTS	Dynamic Ocean Track System	101
ETMS - (hub)	Enhanced Traffic Management System (data flow to/from hub facility)	124

ETMS - (site)	Enhanced Traffic Management System (data flow to/from fielded sites)	61
FSM	Flight Schedule Monitor	5
HCS	Host Computer System	198
NFDC/ NOTAM	National Flight Data Center/ Notice To Airmen	110
OAG	Official Airline Guide	38
OCS	Operational Control Segment	3
RTCA	RTCA Special Committee (SC) on Free Flight Implementation	166
SMA	Surface Movement Advisor	15
TFM	TFM R&D Programs	73
TFM-ART	Traffic Flow Management – Architecture Requirements Team	346
URET	User Request Evaluation Tool	465
	Total Number of Data Elements	**3095**

For context, Table ES-3 displays a data attribute count per data category, showing that over 25% of the attributes are flight category-related.

Table ES-3. Element Count by Data Category

Category Number	Data Element Category Name	Data Elements in Category
1	Demand (Flight)	831
2	Capacity (Resources)	1624
3	Weather	76
4	Traffic Management	319
5	General Resource	19
6	Performance	34
7	Miscellaneous	7
—	Elements not placed in a category	185
	Total Number of Elements	**3095**

For its initial analysis, CAASD focused on demand data — pertaining to flight; that is, flight schedules, flight plans, flight progress, and related aircraft data. This focus was chosen to address the Air Traffic Service (ATS) Concept of Operations [5] for the NAS in 2005, which identifies a flight data thread in the NAS-wide information system, providing “information on each flight from the moment of push-back to wheels-up, including surveillance data in flight, touchdown time and gate assignment.”

Common Flight Data Structures. Flight data need to be shared throughout the NAS. For ATC and TFM, these data need to be shared and accessed in real-time and near-real time by both FAA service providers and Traffic Management Coordinators (TMCs) but also by AOCs and pilots. System developers require that flight data be shared in terms of their data structure and the data associated with it. FFP1 requires that the flight data model structure and relationships represent current and future operational procedures and requirements. These needs can be accommodated by the flight data model presented here.

Another way to view the flight data model is to consider its applications across various phases of flight, as shown in Figure 3-1. This view may be useful in designing applications within each phase in terms of data structure, content, data provider, data recipient, and desired capability. These phases also have correspondence to the several NAS domains (e.g., En Route, Terminal, TFM, CNS, Oceanic) in which NAS capabilities are being implemented.

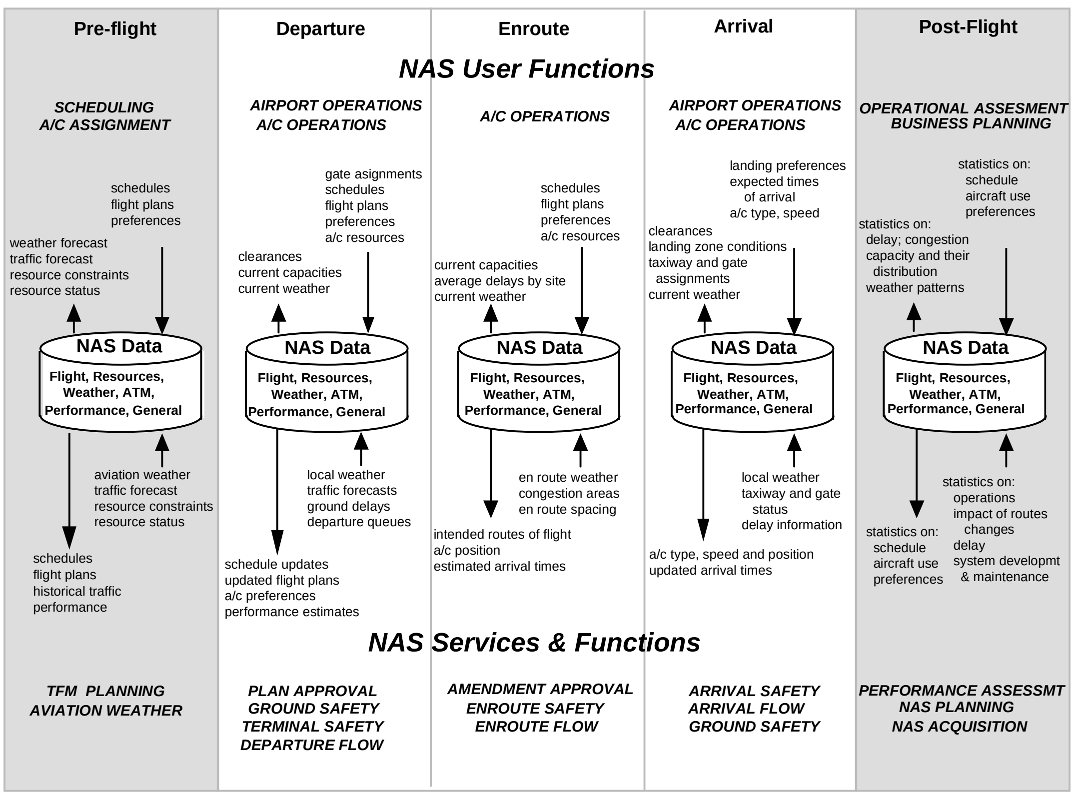


Figure ES-1. Data Support for a Phase of Flight View

The flight data model is an entity relationship model focused on flight. Its structure centers on data uniquely identified for each flight that are associated with several aspects, or event categories, about a flight. These event categories are flight planning; flight events that change the parameters of a flight, such as altitude, speed, and location; and ATM events, some of which are ATC events such as departure and arrival control events and others of which are TFM events, such as ground delays and metering to control traffic flows. In addition, the model also captures description and dynamic information about the aircraft assigned to a flight.

The model also contains links to airspace structure data that are related to flight, such as routes, fixes, NAVAIDs, and other basic aeronautical data, but the model does not attempt to address these airspace structures in detail. Rather, it recognizes that other data models already exist or will be developed to represent the detailed relationships in these areas, and that they will be integrated with the flight data model in the future. Therefore, the description of the flight data model that follows focuses only on data attributes that are directly related to the attributes in category 1, namely demand, or flight, data.

There are several concepts represented in the model, as shown in Figure ES-2. With event type as the category discriminator, these concepts are:

- Flight Plan events, starting from initial flight schedules and plans that may be modified, even as the flight is in progress
- Dynamic Aircraft events, which trace the status of a flight after wheels off with a focus on aircraft position and other parameters about the aircraft, such as speed, climb rates, and fuel usage
- Air Traffic Management (ATM) events, which capture changes in control and alert situations and in traffic flow management
- Aircraft descriptors and an operator description

The model represents many flight events. However, it does not intend to be inclusive at this time since there are some flight events that are not directly modeled in this first design.

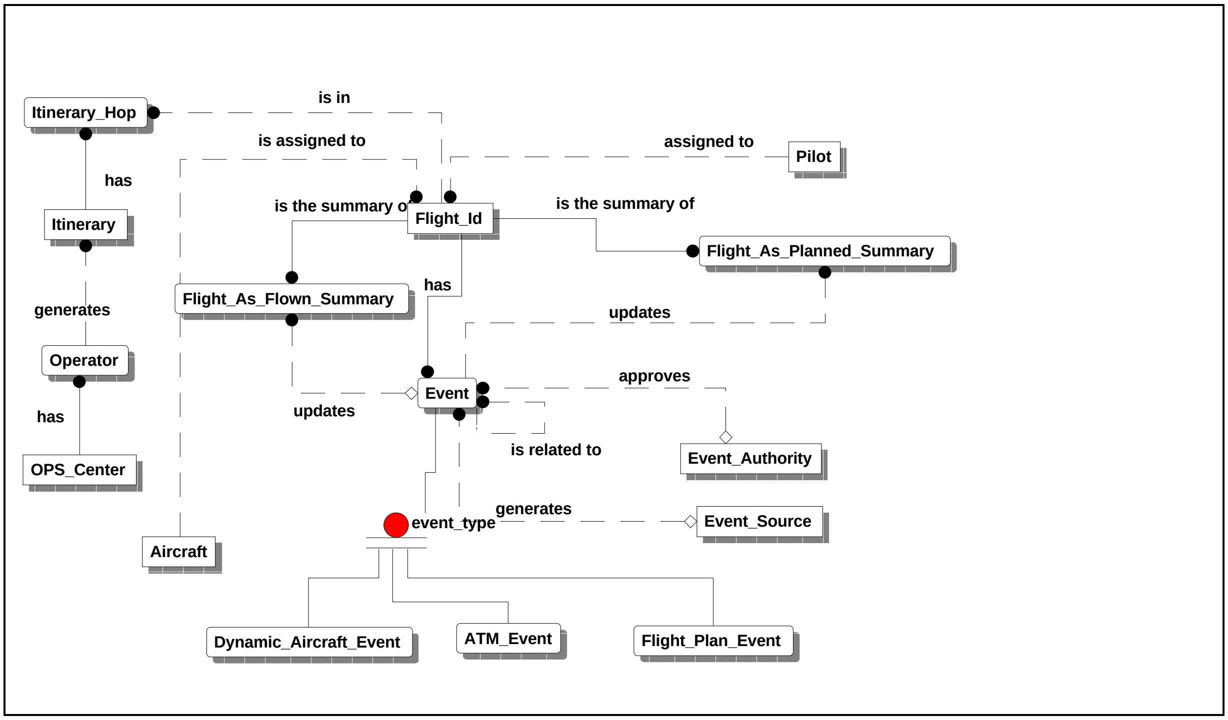


Figure ES-2. Top-level View of the Flight Data Model

Model Features. There are many views of flight that could be represented by the model. This model seeks to maintain the features of the current processing environment that are required for flight processing while supplementing it in two ways: 1) with features known to be current requirements and 2) with probable requirements of future systems. The model permits or facilitates the implementation of these features. It does not guarantee that any or all of these features will be present because that depends on how the model is implemented.

The features of the model include:

- A uniquely identified flight based on a system-generated identifier. The identifier is system-generated because there is currently no unique flight identifier in use. Also, this allows the model to be used internationally.

- A flight itinerary as a set of flights

- A 4-dimensional view of flight position, e.g., typically referred to as ‘x,y,z,t,’ that includes the three-dimensional spatial coordinates (x,y,z) and the time (t) at that position. Today’s flight plans establish information in the first two of these dimensions with limited altitude and time information.

- Numerous variations of flight plans, including proposed, active, cancelled, and trial plans for simulation, testing, and operational decision making. These variations are captured in domain [1] values rather than in the explicit model structure.

- Compatibility with an ICAO flight plan and with a free flight ‘New Age’ flight plan

- Audit trail data to allow searching on ‘who,’ ‘what,’ and ‘when’ regarding a flight. This may assist in answering ‘why’ an event happened.

- A data archive to capture flight events to enable an analysis of these events and the discovery of flight patterns that could lead to improved safety, efficiency, or both

- Descriptive and dynamic aircraft information, including equipage, fuel use, and operating characteristics

- Climb and descent profiles: actual and preferred

- Forecasted position (i.e., trajectory) information

- Flight data that are derivable if it is not stored directly, e.g., wheels off and wheels on times as a function of landing gear position and pressure

Potential outcomes. In practice, the FAA, by encouraging a common view of flight, can achieve system engineering efficiencies in several ways. Some of these are:

- There can be a dialog across organizations and across applications within the FAA about the meaning of flight data and how it will be managed structurally. This should naturally lead to discussions of how flight data itself (i.e., the instance data) should be managed.

- NAS users can participate and contribute to a common view of flight. The RTCA has already acted as a facilitator in this regard, and is, most likely, ready to continue this process.

- Consultants and development organizations can benefit from a NAS-wide view of flight and specific requirements for interoperability instead of being forced to develop their own ‘local’ view. In fact, several such organizations have already shown considerable interest in starting from a common FAA-approved view of flight to save time and money and develop to a view that is more likely to be accepted and interoperable.

[1] The term ‘domain,’ when used in a data modeling context, refers to the set of allowable values that an attribute may take.

- System maintenance organizations can streamline their work in upgrading the flight aspects of application systems by reusing code involving flight structures. Also, system interface process (e.g., message communications and processing) can be simplified by reducing the considerable pre-processing and translation that is required of messages at a receiving system that contain flight data unique to the sending system.

Implementation Issues. There are always many implementation issues in moving from a model to reality in the field. In terms of this flight model, some of the questions that naturally surface include:

- What FAA organization will 'adopt' or champion a flight model? This flight model? This raises the larger issue of the means of coordination among the numerous FAA organizations that have an interest in flight data management and their role in it. The NAS Information Architecture Committee (NIAC) has started this process.

- How will flight data be made available to decision-makers? Through existing applications or through newly-developed applications, views, and screen interfaces? What general query capability will be developed to offer *ad hoc* data access? How will policies regarding data availability include the variety of aviation data users, e.g., commercial, business jet, GA, military, air frieght, international, airport operations?

- What archival capability will be developed to store instance data for system restart and analysis of past operational patterns?

- How will the model be used by the various FAA application development organizations? Will it be used to interface with current systems, when current systems are redesigned, and/or when new systems are built?

- What is the role of the developer? To what extent will the FAA specify a structure for flight data to be used across the developer community?

- With the recent proposals for the development of common data servers at the Centers and TRACONs to offload some of the Host's processing, there is a need to agree on an implementation that simplifies, rather than complicates, the complex processing that now occurs among the applications at these facilities. The HADDS (Host Application Data Distribution System) and the TADDS (Terminal Application Data Distribution System) have been proposed as COTS-based data servers at the Centers and TRACONs, respectively. The Local Information Service (LIS) has been proposed in the NAS Architecture as a more general information service capability for Center-based application support and operational decision-making.

- What influence will NAS users have in the continuing design and development of a flight model? How will this affect implementation?
- Specific to the model, what is the number of views that will be maintained of the flight plan and the actual flight. For instance, will the several periodic snapshots captured during the flight plan process be accessible or will the system only reflect the latest view, and finally, the last view before no more updates are made to the plan?

Acknowledgements

The authors wish to thank Ted Cochrane, MITRE/CAASD, for contributing his experience regarding data management issues from the User Request Evaluation Tool (URET) team, Mike Hermes, for his thorough review of the document, which came along with numerous value-added suggestions, and Patricia Palmer and Lynn McDonald for their attention and care in editing and producing the finished version of the document.

TABLE OF CONTENTS

List of Figures

List of Tables

Section 1

1. Introduction

1.1 Background

The Air Traffic Management (ATM) Integrated Product Team (IPT) (AUA-500), now known as the Free Flight Phase 1 program, has recognized the need for data sharing from both within the ATM IPT systems and programs (extensible to NAS service providers) and with the user community (e.g., air carriers, the military, general aviation, and international aviation). There are significant cost savings and operational efficiencies to be gained if the information flow between NAS systems can be enhanced in ways that foster data standardization and reuse. In addition, the System Development organization (ASD-100) has been supporting data standardization efforts as a prerequisite to more efficient, cost-effective system development and operation.

These two FAA organizations have joined forces in establishing the NAS Information Architecture Committee (NIAC), a cross-cutting organization that is supporting the work described in this document and that is now supported by numerous additional FAA organizations.[2] Interest in a flight data model now spans every IPT whose systems manage flight data; NAS users, who require flight data as part of their operations; and the development community that is building applications requiring a flight data structure.

The exchange of flight data and NAS status data is also of considerable interest to the NAS user community. The Collaborative Decision Making (CDM) program, with original RTCA sponsorship, is a cross-cutting effort to develop a capability to improve operational decision making through more timely, accurate, and consistent exchange of NAS data. In addition, input about flight management needs were taken from RTCA Special Committee 169 Working Group 5. In addition, the AUA-500 R&D program and the ATS/ASD operational concept both address flight planning and flight data issues. CAASD has participated in both of these activities.

Historically and naturally, every organization and application that required flight information independently created a structure for these data. That has led to the situation today in which there are many variations of a structure for flight data, each with its own tailored terms, definitions, formats, and codes.

Based on AUA-500 requirements, an effort was started in FY97 to define a common data management environment for TFM systems. Several application systems were analyzed to identify TFM data requirements. Because the focus of this activity was on data

---

[2] Working with the NIAC, ASD, and AUA, these additional organizations include: (in FAA) AOP, AIT, ACT, AND, ANS, AAF, ARS and AOS; (other) CAASD, SETA, Lincoln Labs, Unitech, AUA/TAC, DMR, SEI

elements shared across the interface among systems, those elements internal to a program were not considered; therefore, only those data elements that are inputs to or outputs from a particular program were included. Data elements were identified through existing documentation, including NAS-MDs, system specification and requirements documents, and other technical sources.

One result of that effort is a set of proposed NAS data categories,[3] shown in Table 1-1, of which category 1 is demand, or flight data. It is that set of flight data which is the scope

Table 1-1. Proposed NAS Data Categories

Category	Description
Demand	Flight data, including information such as the flight itinerary, flight identification, flight planning, flight events and status, and ATM control events that affect a single flight.
Capacity	Resource data, including adaptation data, describing relatively static resources, such as airports, runways, and airways, as well as their dynamic status, e.g., configuration, current capacity, and activation
Weather	Data about weather including terminal and airborne weather observations, forecasts, and reports of weather phenomena
Traffic Management	Data that describe situations in which capacity exceeds resources, and actions taken by ATC, TFM, and NAS users to resolve these imbalances.
General Resources	Data that are not NAS-specific, such as time, geography, and geopolitical data
Performance	Data used to describe NAS operational effectiveness and their ability to meet user needs
Miscellaneous	Data elements that do not fit under the other major categories. Conceivably, new major categories could be identified and defined from these elements.

of this analysis and is the basis for the flight data model described in this document. Additional categories of NAS data must also be modeled and established as system requirements.

Before describing the details of the flight data model design, the definition of the key term for the model, "flight", needs to be defined. In ATM terms, a flight is an aircraft

[3] There have been various proposed NAS categorizations over time. The one proposed here stems from work done in the Traffic Flow Management - Architecture Review Team (TFM-ART) from 1992-1993.

departure from an airport or other takeoff zone followed by its arrival at another airport or equivalent landing zone, with no intermediate touchdown points in-between. By contrast, a flight itinerary is a set of concatenated flights, in which the departure airport for the next flight is the arrival airport of the previous flight, except for the first leg of the itinerary. A flight is defined by a flight identifier, its departure airport, and its departure time. In terms of the data model, flight represents an information domain which includes states, events, processes, activities, and data related to a flight.

1.2 Data Modeling Context

The logical flight data model described in this report is part of a larger effort to develop an effective, implementable information architecture that will improve the information management practices in the automation systems supporting the National Airspace System (NAS). This effort is especially critical to implement the benefits identified by the Federal Aviation Administration (FAA) and the RTCA for the Free Flight concept [2,5]. Free Flight requires collaboration among aviation service providers and NAS users. This collaboration, in turn, requires the exchange of information and interoperability among NAS automation systems.

These automation systems are to be developed by multiple domain Integrated Product Teams (IPTs) including Tower, Terminal, En Route, and Traffic Flow Management (TFM) domains, where there are a number of architectures associated with each systems. The NAS Architecture document developed by ASD includes broad NAS-wide perspectives on automation system architectures. It also identifies the need for further detailed analysis and implementation in such areas as information architecture and technical architecture. The document indicates that these detailed views are best worked from a domain IPT perspective while maintaining a parallel, coordinated effort with a NAS-wide perspective.

CAASD, in partnership with ASD-100, AUA-500, AUA-300, and AUA-200 has been working to develop information architectures and technical architectures from both a domain and a NAS-wide perspective over the past couple of years. Figure 1-1 illustrates a context for architecture developments for the TFM domain, the Terminal domain, and a NAS-wide domain. The domain-specific architectures provide details needed to develop and implement specific systems and programs. The NAS-wide architecture provides the framework and guidance for consistent, efficient, interoperable NAS-wide deployment and evolution of the automation systems based on common and open standards as well as interoperability in sharing flight data with non-FAA systems in the military, international aviation, and domestic NAS users.

The logical flight data model described in this document is a step in the development of an improved NAS-wide information architecture. It also draws from and contributes to the development of the TFM, Terminal and En Route domain information architectures.

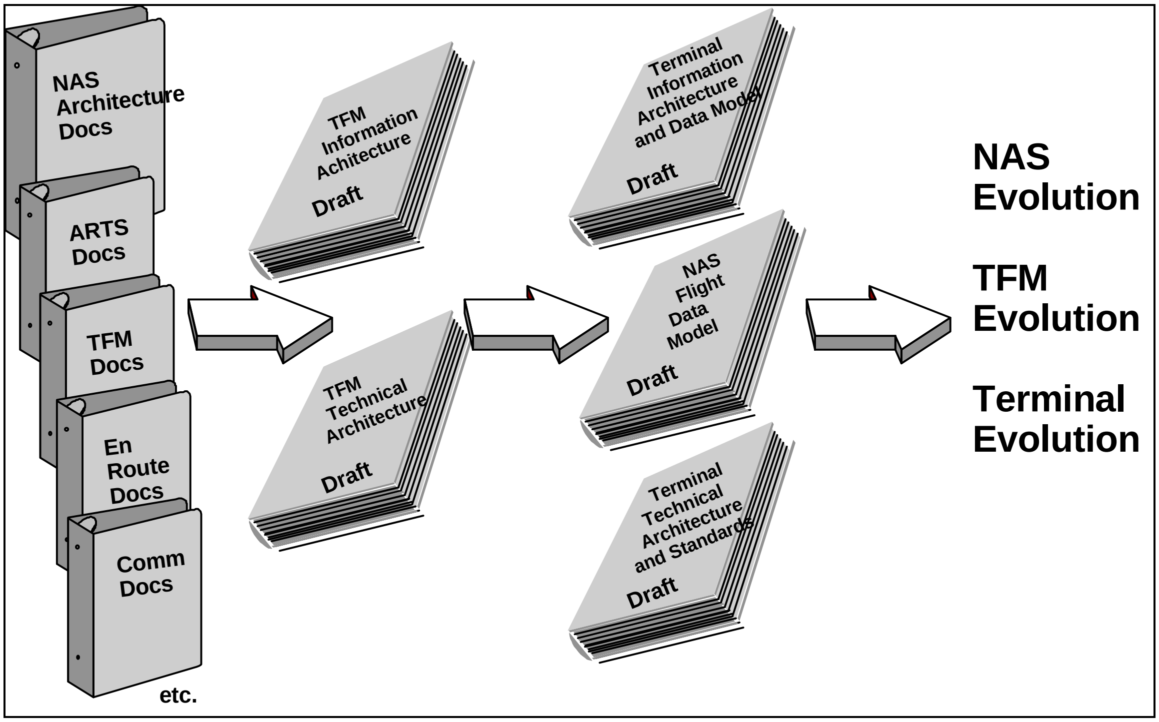


Figure 1-1. Context for Flight Data Model Development

1.3 Purpose

The purpose of this document is to describe an organization for flight data in a flight data model. It is the purpose of the model to capture the essence and the breadth of flight information and invite comment. The model has the flexibility to change in scope and direction and can be easily modified to accommodate additional views and events.

It is not the purpose of this flight data model to become an absolute standard — relevant for all contexts and for all users in exactly one shape and form. Rather, the model is intended to be a common but flexible guide that should help to provide the benefits of commonality, where appropriate, while allowing extensions and customization when and where those are justified. It is expected that the flight model will change continuously over time to keep pace with new operational requirements. The use of automated tools, such as database modeling software, makes the process of updating the data model manageable. The output of the database modeling tool supplies a database management system (DBMS) with the structure it needs to implement the model in a database.

A data model is a formal representation of a related set of information. In this case, a logical data model (LDM) has been designed using the entity-relationship (ER) format in Platinum Technology's [4] ERwin (version 3.5) database design software product that is built on the IDEF1X data modeling standard.

[4] Platinum Technology acquired LogicWorks, the developer of the ERwin product, in the spring of 1998.

The flight data model has these primary goals:

- **Data sharing:** To be used as a standard view of flight data by applications as a fundamental step toward improved data sharing among systems sharing flight data and between systems and NAS users, whether flight data is passed in message format, as the result of a query, or via a database access over an intranet or the internet.

- **Application development:** To be a standard view of flight data for use in application development and system re-design. This includes establishing standards for key flight data concepts, such as coordinate position and time.

- **Information management and data analysis:** To enable better information management for flight data, including its events and corresponding status information, so that real-time flight data as well as post-event historical data can be better managed and analyzed, typically within COTS data management tools, such as database management systems (DBMSs), data warehousing products, and query tools.

Why produce a flight data model when many NAS applications already use flight data? It is exactly this issue — that these numerous applications tend to establish their own ‘local’ view of flight data. Invariably, each application system tends to define flight data differently from other applications. The result is that system interfaces, such as message passing, cause additional pre-processing for the receiving system, at best, and ambiguity or error in interpreting the incoming data, at worst. Such an environment also complicates data sharing across the FAA-to-NAS user boundary and is expensive to maintain and coordinate.

Early versions of this flight data model were provided by CAASD to the FAA, which then released it to development contractors and other Research and Development (R&D) organizations for coordination and evaluation. The flight data model presented in this paper differs in several significant ways from that earlier version due to continued model development.

1.3.1 Uses of a Flight Data Model

A data model — whether developed for flight data, airspace design, or weather — helps to standardize the semantics and the relationships for these fundamental NAS constructs. The model represented here is a *logical* data model (LDM). It describes flight data relationships without regard to their implementation in an operational database. An implementation-dependent version of a logical data model is a *physical* data model (PDM). The latter represents a modification of the former to account for a variety of performance-related factors, including data volumes, data distribution, system connectivity and communications practices, management and control issues, the type of questions most often asked of the data, and specific functions supported.

ERwin, and standards-based data modeling products like it,[5] can be used to produce a PDM and can also produce Data Definition Language (DDL), a formal language that contains the information about data model relationships that is required by a relational database management system (RDBMS) to implement a database schema, such as the one for flight described in this document.

Over time, changes will inevitably be made in any data model to account for new understanding, new functional requirements, or expansion of the original concept. When this occurs, it is preferable to return to the logical form of the model to apply these changes because that is where fundamental data relationships are managed. The disadvantage of applying such changes to a physical model is that they may mask some underlying relationships and data dependencies for the expediency of operational performance. A fuller view of this iterative process is pictured in Figure 1-1, showing the data modeling life cycle and its relationship to application development.

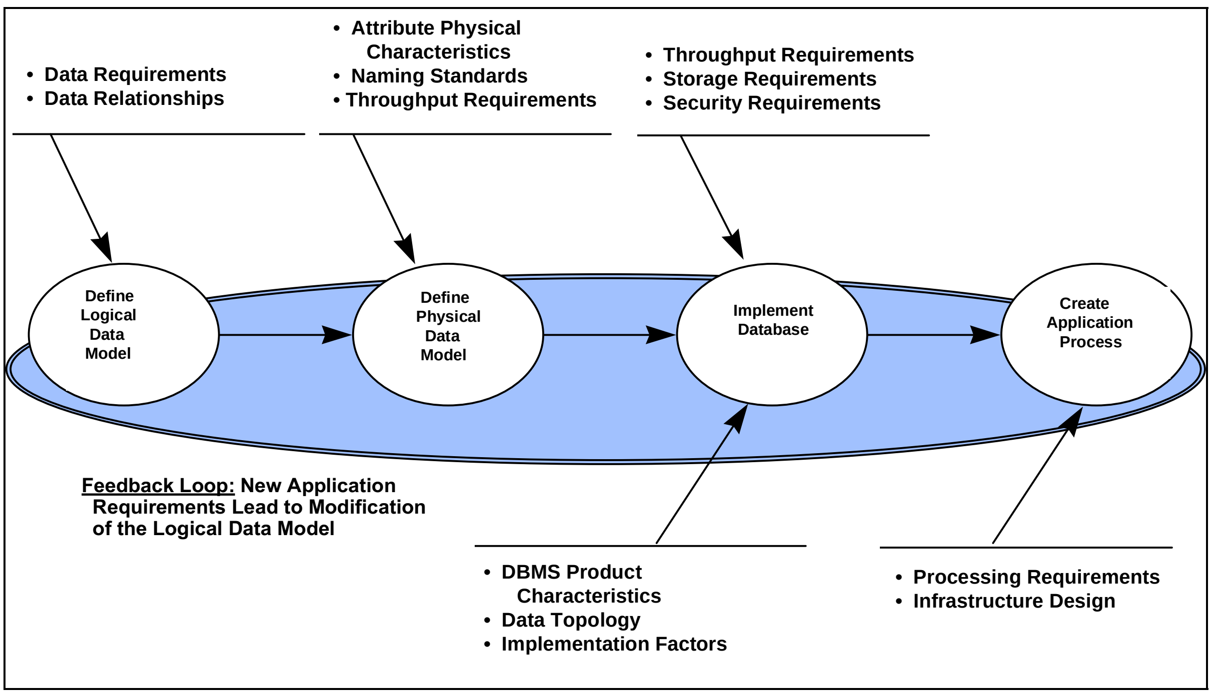


Figure 1-2. The Data Modeling Life Cycle

5 The benefit of using a standards-based product is that a model based on the IDEF1X standard built with ERwin can be imported by most any other IDEF1X-based data modeling tool, such as Oracle's Designer2000 and Sybase's S-Designor. With new interfaces now being built, it can also be imported into Unified Modeling Language (UML) products, such as Rose, from the Rational Corporation.

The flight data model was not developed with a specific target application in mind. In fact, it was designed to be useful to a wide range of flight-oriented capabilities and applications. Some of these uses are:

- Flight planning
  - Pre-flight planning
  - Flight data management (FDM): Host-based and other
  - Post-flight analysis

- Flight operations
  - Departure and arrival management
  - En route spacing and conflict probe

- Data exchange
  - Among FAA applications
  - Between the FAA and NAS users

- System acquisition, development, maintenance, and redesign
  - Application development and interoperability
  - DBMS design and implementation
  - Data visualization and presentation
  - Data warehousing, retrieval, and analysis

- Data standardization as part of NAS-wide information management

For example, pre-flight information is required by the FAA to assure that it has the resources to safely manage demand and also to meet throughput goals on minimizing delay and diversions. Real-time flight information is needed by service providers and flight operators for aircraft separation and to safely and efficiently manage individual flights as they are being flown. Post-event historical data are[6] needed by analysts to understand the behavior and dynamics of the NAS, as well as to understand how these behaviors and dynamics are influenced by capacity and procedural changes.

A common flight data model is part of establishing data standards at the NAS level. Were the FAA to develop a data warehousing capability, the use of a common flight data model throughout the NAS would streamline the data flow process into and out of the warehouse.

[6] Much common usage today ascribes the singular case to the term 'data' whereas traditionally, the word 'datum' represents the singular while 'data' represents the plural. This document adheres to the traditional use of these terms.

1.3.2 Evolution of the Current Flight Data Model

There were several inputs to this analysis and to the development of a flight data model. The primary input, however, was the collective set of flight-oriented data found in the TFM systems below. Other inputs include Host flight data processing, the Air Traffic System (ATS) Concept of Operations, RTCA guidance, and the ICAO flight plan standard.

The following TFM-related systems were the basis for this analysis:

- ETMS (Enhanced Traffic Management System)
- TMA (Traffic Management Advisor)
- pFAST (Passive Final Approach Spacing Tool)
- FSM (Flight Schedule Monitor)
- CDM (Collaborative Decision Making) and Collaborative Routing
- ODMS/AIS (Operational Data Management System/Aeronautical Information System) [7]

Other systems that were analyzed include: URET (User Request Evaluation Tool), Fast-Time What-If I, Host Computer System (HCS), Interactive Flight Planning, and the OAG (Official Airline Guide).

In addition to these numerous application programs, multiple other sources generate events for flight plans and flights. These include:

- Pilots, airline operators, and Flight Service Station (FSS) operators who submit, or file, flight plans

- Service operators who submit flight plans for trial purposes

- Service providers and the Host computer system that activate, give clearances, give cancellations, and submit amendments to flight plans

- Aircraft themselves via Data Link or Global Positioning Systems (GPSs).

[7] The AIS is the system of record in the NFDC that manages aeronautical and airspace structure data. ODMS was a development in the early 1990s to replace the AIS. That development is now called NASR (NAS Resources).

1.4 Scope of this Effort

The scope of this effort is to create a view of flight that is inclusive across multiple users of flight data that also considers future flight data management requirements, such as for flight intent. The model describes fundamental concepts for flight, e.g., a flight plan, a flight event, and the role of the aircraft on a flight. It also establishes fundamental relationships among flight entities.

Although it is somewhat indirect, an implicit part of this effort is to associate a standard vocabulary of flight with the various flight concepts and to associate these terms and meanings with the terms now used in legacy systems. This effort is not intended to be inclusive of all current systems using flight data but to examine a sufficient number of key systems to create a sound basis for this initial model development.

The effectiveness of flight data for operational decision making relies on having up-to-date capacity (e.g., NAS resource) and weather data. These categories need to be addressed in future but are not in the scope of this effort.

1.5 Approach

Since the objective of the flight data model is to capture the behavior of flights, a structured database design methodology was used to represent the components of flight (entities) and their relationships. This methodology, combining an information model, a state model, and, to some extent, a process model, defines the following three components:

- The elements that make up a flight model, including data entities, data attributes, and the relationships among them.
- The behavior of flight entities, in which each entity and relationship may have a life cycle which is a pattern of behavior. For example, a flight plan may change states from initial submitted status ('filed') to 'cleared,' and then to 'active,' and finally to 'terminated' (by cancellation or termination at the end of a flight).
- The activities and events involved in each state of a flight. For example, during the active state of a flight plan, amendments can be added to change its content. During a flight, there are various changes of event state, from events such as a departure, a cruise climb, an airport arrival, and a gate arrival.

Within each phase of a flight, related events generate a snapshot of the flight, its states, and the information about events, from which one can study a flight's behavior. Ideally, these data are captured and maintained in a way that makes them accessible at a future time. Today, data warehousing technology is the established way in which large amounts of post-event, structured data can be captured, stored, analyzed, and accessed and disseminated.

Section 2

2. Related Flight Plan Contexts

As mentioned above, organizations have tended to develop their own data structures in developing applications. The sections below describe the use of flight data in the following contexts:

- International Civil Aviation Organization (ICAO)
- RTCA Modified Flight Plan
- Host Computing System (HCS)
- Enhanced Traffic Management System (ETMS)
- En Route ATM Decision Support Toolset (ERATMDST)
- Free Flight Phase 1 (FFP1)

Standards-based organizations such as ICAO are certainly interested in promulgating common flight data standards to enhance the safety and efficiency of global airspace. The RTCA, likewise, promotes a federal-industry partnership for improved airspace operations, and more efficient management of flight data and the exchange of these data is a cornerstone of their interest.

2.1 International Civil Aviation Organization (ICAO) Flight Plan

Annex 15 of the ICAO flight plan standard describes an ICAO-standard flight plan by the fields shown in Table 2-1. The line numbers correspond to those on the ICAO form.

Table 2-1. ICAO Flight Plan Format

Line Number	Description
1	Flight ID, including air carrier, flight number, and flight type
2	Aircraft designator, navigation capability, transponder type
3	Proposed departure airport and departure time
4	Initial cruise speed and initial cruise altitude, route of flight
5	Entry and exit speed on entering the Minimum Navigation Performance Specifications (MNPS) airspace on a North Atlantic track
6	Specification of a step climb to a specified flight level
7	Destination airport, estimated time en route (ETE), and alternate destination airports
8-10	Estimated en route times for flight information regions (FIRs), Center boundaries, or significant waypoints

	along the route
11	Selective Calling System (SELCAL) code and registration information for the aircraft

This flight plan format is used by Aeronautical Operational Control (AOC) organizations and is also reformatted by the FAA into NAS plan format for use by the Host.

2.2 RTCA Proposed Modified Flight Plan

The RTCA has an interest in a wide range of aviation issues, including an interest in a common model for flight data to address ATM-to-AOC information exchange. In doing so, it has proposed a variation of the ICAO flight standard called a New Age Flight Plan.[8] This construct is "an expanded flight plan to provide better information for management of NAS resources and for more accurate prediction of demand for those resources, and identification of steps toward implementing such a flight plan." This variation introduces additional data to improve flight plan processing and gain additional operational benefits.

The RTCA proposal starts from the ICAO standard flight plan and proposes to augment it with the additional information, as described in Table 2-2. The line numbers correspond to those on the RTCA form.

Table 2-2. RTCA Proposed New Age Flight Plan Format

Line Number	Description
1-11	Same as the ICAO flight plan format (see Table 2-1 above)
12	Planned aircraft takeoff weight, intersection takeoff capability (y/n), and time to top of climb (flight level and minutes to achieve)
13	Preferred departure runway, runways capable of being used, acceptable delay for the preferred runway and route, unacceptable runways, and an alternate route if the preferred route is unavailable
14	Planned aircraft landing weight, preferred landing runway, acceptable delay waiting for the preferred runway, unacceptable landing runway, and minimum capability levels for the aircraft and crew on approach and landing
15-17	Description of the alternate route(s)
18-19	Estimated time en route for the alternate route(s)

8 DO-241, "Operational Concepts and Data Elements required to Improve Air Traffic Management (ATM) - Aeronautical Operational Control (AOC) Ground - Ground Information Exchange to Facilitate Collaborative Decision Making," 6 October 1977, RTCA, Special Committee 169; quotation taken from the recommendations, page ES-4.

2.3 Host Computer System (HCS) Flight Plan

The FAA's NAS MD-311 [3] document describes the structure and format of the HCS flight plan, shown in Table 2-3, which is defined as a message type. The purpose of the flight plan message is to establish a database for a flight plan. The entire flight plan, as accepted, is stored for use by strip printing, display, printout, and inter-facility data transfer functions. Individual fields of data are interpreted, processed, and stored for use by other program functions.

Table 2-3. Host Flight Plan Message Format

Field Number	Description
00	Source of message (entering facility)
01	Message type
02	Flight identification, including aircraft identification, departure point, beacon code, proposed departure time
03	Aircraft data, including type of aircraft and type of airborne equipment
04	Beacon code
05	Filed speed
06	Coordination Fix
07	Coordination Time
08	Assigned altitude (active flight plan)
09	Requested altitude (proposed flight plan)
10	Route Data
11	Remarks

2.3.1 Flight Data Processing

The Host system at each ARTCC operates as the primary database for flight information in the NAS. It manages data for a flight from pre-flight through arrival phases and generates data to be used in the post-flight phase. It exchanges flight data messages with other NAS applications and systems and processes data inputs.

Flight data processing includes the following activities:

- Entering flight plans into the flight database that were filed via these mechanisms:
  - By a pilot through an air information service center
  - By airline or Department of Defense air operations centers
  - By automatic entry of permanently stored, pre-filed, repetitive flight plans filed by airlines based on day of the week and other criteria
  - By air traffic control system personnel

- The flight plan describes the desired route of flight, the time of flight, various aircraft characteristics that are helpful for control and identification purposes, and miscellaneous information that might affect the control of the flight.

- The processing of algorithms to generate a four-dimensional trajectory and the posting of information that ensures that the appropriate flight information is sent to the correct automation system or user at the correct time. For example, four-dimensional trajectory processing may trigger an alert to an ATC controller if the estimated time to a fix is in question.

- The processing of NAS user and ATM-entered flight plan amendments, including updates for events such as departures, flight plan cancellations, flight progress reports, and closure messages, and the posting of this information

- The automatic processes that operate on the flight database and disseminate or otherwise control flight data information as required. These processes include changes in the airspace adaptation data base that impact individual aircraft flight data, such as a route closure or a runway configuration change

- The acceptance of and response to manual requests for information concerning a particular flight or set of flights

- Support to other automation functions that require information from the flight database

- Coordination of acknowledgments and responses to ensure that all systems and personnel are current with respect to the status of each flight that is relevant to a decision maker’s area of responsibility.

The Flight Data Model described in this document attempts to model the data needed to support these processes. It includes detailed flight plan information, adding new attributes, such as ground movement events, to support information needs identified in future concepts of operation. Flight plan events capture the creation of and changes to the flight plan, as well as the initiator of and reason for the change. Information about the aircraft executing the flight is logically separate from flight plan information, but specific information such as navigation equipment and performance characteristics that affect flight plan processing can be associated with a flight. Flight events, which are distinguished from flight plan events, reflect shifts in flight control, as well as in flight position. The Flight Data Model has links to information about airspace structure and ground resources, including adaptation data, that will be used effect changes to a flight plan.

2.3.2 Host Flight Data

The Host exchanges data with applications via messages, which are manually entered or internally generated. Each message has associated logic checks to verify that valid information has been entered and that the message meets 'eligibility' requirements; that is, entry of the message is compatible with current flight plan status and the message source is eligible to enter the message. Each message also has associated processing rules that may result in changes to the flight plan or flight status, as well as additional message output. The major message types used to maintain the flight plan database, which are documented in NAS-MD-311, include the following:

- Flight Plan: Used to establish a database for a flight plan, including the flight identifier, the type of aircraft, its beacon code, its speed, departure point and time, coordination fix and time, assigned altitude, and requested altitude. Also, the estimated time en route may be appended to the destination element of a proposed flight plan, while the estimated time of arrival may be included in an active flight plan. The entire flight plan, as accepted, is stored for use in flight strip printing, display, printout, and inter-facility data transfer functions. Individual fields of data are interpreted, processed, and stored for use by other program functions

- Flight Plan Amendment: Used to modify, add to, or delete previously filed flight data; the accepted amendment data become part of the flight database. Relevant data include flight identifier, the reference to the field to be amended, and the amendment information. Flight plan amendments can have many impacts, depending on the nature of the amendment.

- Mission Flight Plan Message: Used to establish a database for a flight plan, for the purpose of printing flight strips

- Beacon Code Modification: Used to assign or change non-discrete beacon codes, and to assign or change discrete codes not presently assigned to another aircraft

- Discrete Code Request: Used to request or change a discrete beacon code. This action cannot be input in the same message with any other action that also requires a Flight Identification.

- Departure Message: Used to activate a proposed departure or a proposed airfile flight plan — a filing in flight, e.g., a flight under visual flight rules (VFR) converting to instrument flight rules (IFR). Relevant data include the flight identifier, coordination time, and assigned altitude.

- Hold Message: Used to initiate, modify, terminate or cancel a hold action for any specified flight. The hold fix may be a converted fix along the aircraft's route of flight, the present track position, or, if the aircraft is not being tracked, the present flight plan position.

- Progress Report: Used to update the fix times and status of an active flight plan. It may also be used to release a flight plan from a prior hold action. Relevant information includes the fix name and time at the fix.
- Reported Altitude: Used to update the reported altitude in the aircraft data block.
- Assigned Altitude: Used to modify the assigned altitude or flight level for the specified flight.
- Remove Strip Message: Used to remove from the system all flight data for an entered or tentative flight plan and associated track, if any.
- ICAO Flight Plan message: Used to establish a database for a proposed international flight plan. Currently, this message is converted into a NAS Flight Plan message.
- ICAO Departure Message. Used to activate a proposed flight plan by entering a time of departure. Currently, this message is converted to a NAS Departure message.

In addition to messages that help to maintain the flight plan database, there is a group of track control action messages. These messages, described in NAS-MD-311, correspond to the flight portion of the flight information model, and include:

- Accept Handoff: Used to assume control of a track, thus completing the transfer of control. It may also be used to retract the transfer of control.
- Initiate Handoff: Used to initiate the transfer of control for a flight from one sector to another sector or facility.
- Select automatic Handoff: Used to prevent or allow automatic handoff for a sector or individual aircraft.
- Track: Used to initiate or re-initiate a Free or Flight Plan Aided Track on an identified radar trail, and to modify certain track data (e.g., heading, speed, altitude).

2.3.3 System Interfaces

The analysis of TFM applications, described in Section 1.3.2, includes the identification of data flows among those systems. Figure 2-1 shows the data exchange between the HCS and the major Air Traffic Management (ATM) systems, as defined in the

HCS/ATM Interface Requirements Document (IRD). An important system engineering issue is that each of the major systems in this environment has defined its own view of flight data. One of the goals of a common flight model is to integrate the view and management of flight data more smoothly across these systems and new systems that will follow.

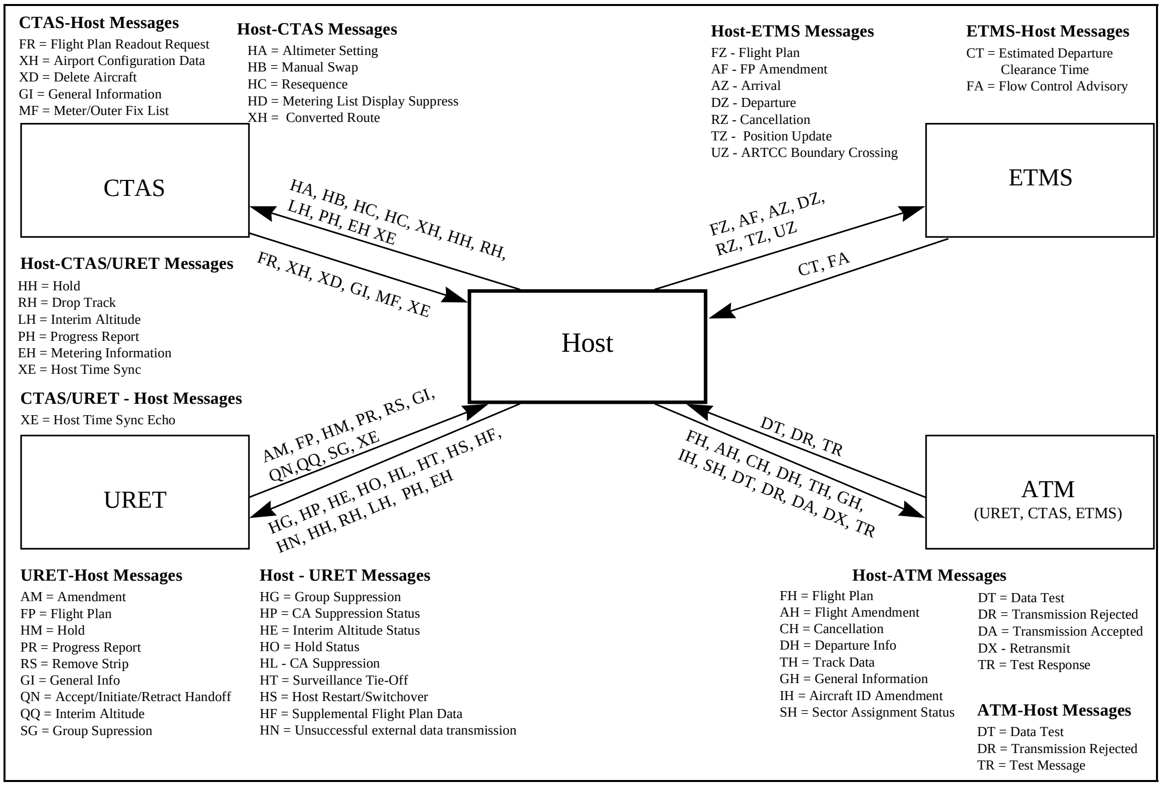


Figure 2-1. Host - Air Traffic Management (ATM) Interfaces

Table 2-4 lists the interfaces between the Host Computer System (HCS) at the Centers and other systems and facilities within the FAA and to NAS users. The table generically describes the types of information passed to and from the Host; the information is basically a superset of the messages identified in Figure 2-1. It is apparent that the Host is the central database of flight data and that there is a wide variety of consumers — and uses — of these data.

Table 2-4. Host-to-FAA External Interfaces

Facility Name	Information Type
Airline Dispatch Office	Pre-filed Flight Plan Changes Pre-filed Flight Plan Cancellations
ARINC	Position Reports
Direct User Access Terminal (DUAT) Services	Flight Data Acknowledgments ICAO IFR Flight Plan Proposals ICAO VFR Flight Plan Proposals SAR Alerts SAR Flight Plan Response
DSR (En Route) ISD (Oceanic)	Active Flight Plans Altitude Reservation Data ATC Special Lists ATC Special Lists Updates and Requests Clearances Control Coordination Control Data Updates and Requests Coordination Data Data Link ATC Messages Demand Data Emergency Information Flight Data Flight Progress Data Probe Data Probe Requests SAR Alerts and Data
Flight Service Data Processing System (FSDPS)	Flight Data SAR Alerts
Foreign ATC	Control Coordination Data Flight Plans Handoffs
Military ATC	Control Coordination Data Flight Data
Military Base Operations	Cancellation Messages Flight Plan Amendments
Mode S Sensor Surveillance	Data Link ATC Messages
NORAD	NORAD Flight Plan Data (Incoming ADIZ, CADIZ, DEWIZ)

Other Area Control Computer Complexes	Control Coordination Data Coordination Data Emergency Information Emergency Mode Messages Flight Progress Data Flight Data Search and Rescue (SAR) Alerts and Data
Tower/Approach/TRACON (ARTS/STARS)	Control Coordination Data Coordination Data Data Link ATC Messages Demand Data Emergency Information Flight Data SAR Alerts and Data
Traffic Management Processor (TMP)	Altitude Reservation Data Analysis and Evaluation Data Capacity Data Coordination Data Demand Data Emergency Information Estimated Departure Clearance Times Flight Data Flight Progress Data Traffic Management Advisories

2.4 Enhanced Traffic Management System (ETMS) Flight Data Model

The Enhanced Traffic Management System (ETMS) gathers and aggregates data for forming a national view of air traffic. Customers of ETMS are airlines and traffic management units (TMUs) at various air traffic control facilities. ETMS capabilities are managed from a hub at the Volpe National Transportation Systems Center (VNTSC) in Boston, MA that collects information about air traffic, such as aircraft position and flight plan data, from various sites. These sites include the Center Host computers, oceanic air traffic control centers, Terminal Airspace Radar Terminal System (ARTS) computers, and other sources. The information is synthesized into a model of all flights (proposed and active) in the NAS. This model is referred to as the ETMS NAS model. The NAS model is maintained in a proprietary, memory-based database management system within ETMS.

After receiving flight-based inputs from the various Centers and Terminals and after considerable processing at the hub, ETMS then ships processed flight data back to the more than 80 Centers, Terminals, and other sites. This includes the Aircraft Situation Display (ASD), the primary processed output from ETMS that displays a color-coded view of the last known position of every aircraft in the skies. It is updated regularly, every five minutes, based on radar data originally sent from the Host processors at each

Center and every one minute by data sent from ARTS computers at several Terminal locations. In the future, updates may occur every two minutes.

Construction of the NAS model requires associating the flight messages received from the hub's many data sources with individual flights.

Flight messages must be associated for two reasons. First, each of the data sources used by the ETMS hub models flights independently from one another. Second, many of the messages received at the hub from a single source, such as a Center, do not contain a convenient means of associating records, such as a unique flight identifier. The association task requires extensive heuristics because the data in the messages received at the hub are not sufficiently defined to identify a flight uniquely. For example, track data (aircraft position reports) contain a flight id. But the same flight id may be in use simultaneously by different flights and may be reused throughout the day.

Similarly, each Host computer system generates a unique identifier for every flight track. However, since the Host computers across the NAS were not architected to be interoperable, the same identifier may be in use by several Host systems simultaneously. Thus, as a workaround, geographic and time windows must be used to associate a track message with a given flight. One of the objectives of the flight data model proposed here is to create a unique flight identifier to avoid the current ambiguity in identifying flights and the increased processing now required.

An ETMS output data stream exists in two versions. The version 4 series data stream is sent to airlines and other subscribers. The version 5 series stream is distributed to ETMS sites that are typically located in traffic management units (TMUs) at air traffic control facilities, to the military, and to international air traffic facilities.

The version 4 data stream is essentially a pass-through of the various records received at the hub. Since there is no association among the messages received by the hub, there is no association among the messages in the version 4 stream. Thus, anyone using this stream to model flights must perform the same task of association that is performed at the ETMS hub, which is considerable.

The version 5 data stream consists of a subset of the information contained in the ETMS NAS model in the form of a set of Apollo Domain PASCAL records that describe flights. Generation of records in the ETMS data stream is triggered by receipt of new data at the hub. The record generation is a two step process:

- The data are incorporated into the ETMS NAS model, which includes international flights bound for U.S. destinations
- Records are derived from the model and distributed to the sites

The event that triggers the generation of a record in the stream is identified in the record as a source. For example, receipt at the ETMS hub of a departure (DZ) message from a Host computer at a Center results in the generation of a time message in the ETMS data stream where the source is set to 'DZ.'

The version 5 stream preserves the association among messages established at the hub by including a flight index that identifies the flight with which the messages are associated. Thus, users of the version 5 stream do not face the task of message association, unlike users of the version 4 stream. A flight index is unique for a given flight as long as that flight is active in the ETMS hub's NAS model. Once a flight is deleted from the NAS model, the flight index will be reused. Therefore, users of the version 5 stream must still determine when a flight index is being reused. However, this is a much less demanding task than performing the initial message association at the hub.

The new stream contains eight record types describing flights. Two of these types — the block and critical records — are used for recovery from a crash of either the stream or the site. The other six represent the data presented to the site during normal operations. The block and critical records contain no data that are not delivered in the other six records received by ETMS sites during normal operations. Thus, only these six need be considered for building a data model. These six normal operational records are:

- TZ: position information for flights with departures or destinations in the continental United States
- Position: oceanic position information
- Route: flight attributes and predicted routes
- Block altitude: second altitude level for a flight filing a block altitude
- Cancel: cancels either a flight or a ground delay program
- Time: times for departure, arrival and controlled departure times from ground delay programs that are generated every five minutes when a flight has failed to depart by its filed departure time.

2.4.1 ETMS Data Modeling and Data Access

When a data stream is received by the ETMS sites, it is placed in a proprietary database tailored to support specific ETMS applications, such as the ASD, but, due the proprietary nature of the ETMS architecture, the data so received cannot be queried by other systems or by *ad hoc* query. In fact, queries on ETMS data must be sent to the hub. For example, to find a list of flights planned to arrive at a given airport during a given hour, an ETMS site must send a query to the hub for processing even though the data are potentially available at the site. The set of queries to which the ETMS hub can respond is fixed, and implementing a new query typically currently takes six to nine months.

Such processes drive the current motivation to manage ETMS data in a DBMS to transform a closed application system into an open data service. The capture of the data stream in an open architecture would facilitate the formation of queries on the ETMS data and give users of the data control over what data are retained and for what length of time.

Staff at CAASD have produced an ETMS-based flight data model, also in ERwin, to improve flight data management in ETMS. It is designed as a basis for analysis of the NAS and for generating NAS simulation scenarios. The schema is faithful to the ETMS stream in that all of the various types of data contained in the stream are captured. The modeling approach is incorporated in a schema, shown in Figure 2-2 below. It is not intended as a standard view of flight data.

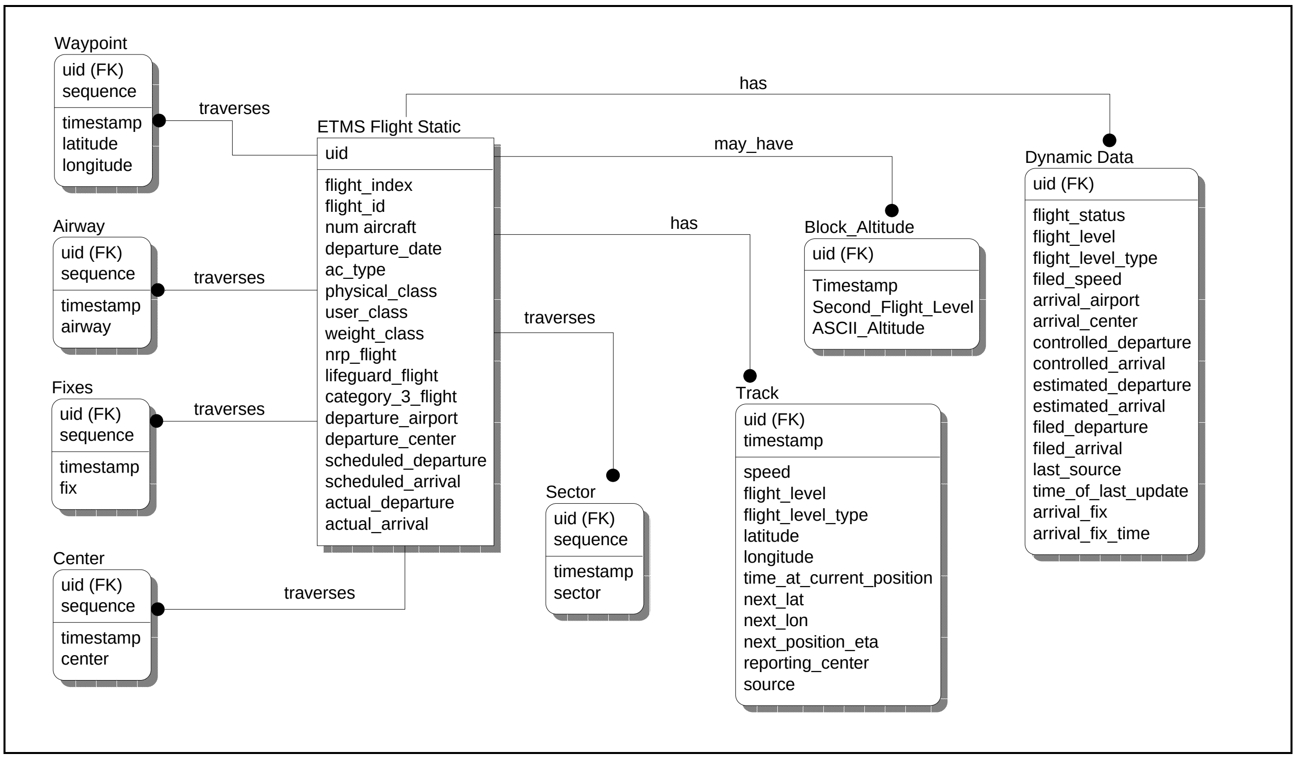


Figure 2-2. ETMS Flight Data Schema

Four transformations are applied to the data. These are:

- Separate data values packed into bit fields within integers at the hub are extracted and placed in individual fields.
- Times are uniformly represented as UNIX time in number of seconds since 1/1/1970.
- The names of the FAA Centers are expanded from one character to the standard FAA 3-characters designations, e.g., ZKC for Kansas City.
- Enumerated values, such as user classes, are converted to C type enumerations. In the data stream, these enumerations are typically represented as strings.

Aside from these transformations, the data are inserted into the schema without interpretation.

The scenario generation and analysis processes require that all types of information contained in the stream be captured, but not that all of the updates of the information be retained. The program populating the schema has different policies for four groups of information. These groups are:

- Static single-valued attributes
- Dynamic single-valued attributes
- Position data
- Route data

Static single-valued attributes cannot be changed without changing the definition of a flight. These values are captured in the ETMS_Flight_Static table. Once a record containing one of these values is received, the value is never updated. Dynamic single-valued attributes can be freely updated. The simulation generation process requires only the latest values of the dynamic data; therefore, these data are updated, with only the latest value retained. Position data are required to model the flight's 'as flown' routes. Therefore, each position report is retained in the Track table. Order between position reports is maintained by the current position time field. For simulation purposes, the route data represents an 'as filed' flight plan. Therefore, the route data for a flight are deleted and replaced with each route message received until the flight departs. Thereafter. the route data are not updated.

The unique identifier key that allows selection of all rows for a flight from each table is generated during the process of loading the schema so that flights sharing a flight index can be distinguished.

This ETMS flight data model makes the following assumptions about various data items and model conventions:

- Units of Measure: All altitudes are in flight levels. Times are UNIX times (number of seconds since Jan 1, 1970), latitudes and longitudes are decimal degrees. Speed is measured in knots.

- Speed Type: The best current guess is that ETMS preserves the NAS convention that track speeds are ground speeds and all other speeds are true air speeds. But since this is undocumented in the ETMS system specification, one would have to examine the source code to determine whether NAS speeds are passed through.

- Enumerations: In the ETMS_Flight_Static table, the columns lifeguard_flight, category_3_flight, and nrp_flight have the values 'TRUE' or 'FALSE.' A category 3 flight is equipped for ILS category 3 landings. An NRP flight participates in the National Route Program (NRP). Physical class has the values 0 through 3 representing 'piston,' 'turbo,' 'jet' and 'helicopter,' respectively. User class has the values 0 through 5 representing 'other,' 'air taxi,' 'cargo,' 'commercial,' 'general aviation' and 'military,' respectively.

- Times: Several types of time measurement are associated with each flight. Static times appear in the ETMS_Flight_Static table and are the scheduled and actual departure and arrival times. Scheduled times are an ETMS estimate about flights that will fly for a given day and correspond to Official Airline Guide (OAG) times supplemented with ETMS historical information. A complete set of scheduled times is loaded into the stream as much as twelve hours before a flight's scheduled departure. Actual arrival and departure times are derived from Host DZ and AZ messages.

- Dynamic times appear in the Dynamic_Data table and are proposed, estimated and controlled departure and arrival times. Proposed times are derived from Host FZ messages, controlled times are derived from Expected Departure Control Time (EDCT) messages, and estimated times are an ETMS 'best guess.'

- This version of the schema is non-archiving; that is, new information overlays current data (e.g., route data) with the exception of track data, which are archived.

- The following tables are always updated as a group since a complete set of rows for a flight is included in each route message: sector, center, fix, waypoint, airway.

Many NAS systems, including ETMS, contain algorithms that perform trajectory modeling. ETMS generates route data by calculating the cells of a geographic grid that will be traversed by a flight. Route estimations are based on the route of flight field in the flight plan (FP) message and the previous history of specific flight identifiers. The lists of route elements are then generated by extracting the elements contained in each cell of the grid traversed by the route. Grid data are stratified by altitude. Only the entities associated with a flight's predicted altitude in each cell are extracted from the geographic database and added to the route message. Thus routes are very altitude sensitive. Once a flight departs, the route description provided for a flight may change repeatedly simply because the altitude changes without any change of intent for the flight's two dimensional route. This information has implications for how route information is managed, and, therefore, how flight data are managed in ETMS.

2.5 En Route ATM Decision Support Toolset (ERATMDST)

The En Route ATM Decision Support Toolset (ERATMDST) project in AUA-200 is another area that requires consistent flight plan structure and processes. It is designed to incorporate Traffic Management Advisor (TMA), and perhaps Descent Advisor (DA), functionality with Initial Conflict Probe (ICP) functionality into a system using a common system views and resources, such as a common trajectory modeler and common databases. It is proposed to be implemented about 2005.

Among the data sets that ERATMDST is examining are those for NAS adaptation and flight modeling. It is currently developing a specification for a flight model based on a 4-dimensional flight model, i.e., the current 2-dimensional plane, plus altitude and time.

The ERATMDST activity is a related activity but is not part of the development of the flight data model described here. There is opportunity to coordinate the flight specification in ERATMDST with the flight model.

2.6 Flight Plan Processing for Free Flight Phase 1 (FFP1)

With CAASD support, the RTCA has developed an operational concept for Free Flight Phase I, the FAA's near-term program to provide user benefits by 2002.[9] This concept addresses a wide range of issues, among them are two below that involve data management and, within that, flight data management. As quoted from the FFP1 Operational Concept,

> ***"Technical and Information Interdependencies:*** This area is critical to the evolutionary development process because it highlights the arrangements, interactions, and interdependencies of the FFP1 capabilities as a common set of systems and services. This requirement is judged to present the greatest challenge for FFP1 implementation. The technical and information architectures of the independent systems will be brought together into a cohesive array of features delivering increased enhancements and performance to the NAS."

The flight data model is a component to be used to standardize the view of flight across FFP1 and across the NAS. It contains new content and a proposed structure for flight-

[9] *Government/Industry Operational Concept for NAS Modernization, 1998-2002: Free Flight Phase 1, Volume I,* RTCA, June, 1998.

related data. The technical aspect of this issue is how flight information will be deployed, made accessible, protected, and used in system development.

> ***"Adaptation:*** This area is concerned with site-specific adaptation of FFP1 capabilities. Much of the new functionality delivered in the FFP1 capabilities is complex and very data dependent. Some of the site-specific airspace data are inconsistent across centers and across systems. During this period, an airspace review process will be on-going, potentially altering some site data. Adaptation across centers for a single system and across systems installed at the same site are concerns for FFP1 implementation."

Adaptation is not directly a flight data issue. It is a capacity, or resource, issue, and as such it does interplay directly with NAS system performance and flight. Nevertheless, NAS users need to be aware of airspace structure, the unconstrained capacities of NAS resources, and the current capacity of those resources based on near-term constraints, such as demand and weather. In addition to flight data, FFP1 will require a consistent view of such adaptation data to improve interoperability and the exchange of airspace information.

The management of flight information is an important component in the implementation of the technical and information architecture changes that will be required for FFP1 and other NAS activities.

The FFP1 Operational Concept describes a number of uses for a coordinated flight plan in the context of collaborative decision making. The text below is also taken from that source: [10]

> "During FFP1, flight planning is enhanced by collaborative decision making, capabilities used by Airline [11] Operational Control (AOC) and Air Traffic Management (ATM). These capabilities enable information sharing on a variety of NAS status data and improve the implementation of the ground delay program, when necessary. The capabilities support improved flight planning and associated services, resulting in increased collaboration between users and service providers.
>
> ***Exchange of Real-Time Information***
>
> Improvements to flight planning are provided by automation capabilities that increase the sharing of near real-time information between users and service providers regarding airspace or airport flow restrictions. A variety of information regarding the state of the NAS operations and infrastructure is available via electronic means to NAS users. This information can be used by users to prepare flight plans that result in a reduction in the number of in-flight

---

10 *FFP1 Operational Concept*, volume 1, The MITRE Corporation, 1998

11 The term 'Airline' has been replaced by the term 'Aeronautical' in the acronym AOC.

reroutes due to misinformation, reducing workload for all parties, and significantly improving the chances that air traffic control (ATC) will approve a user-preferred flight plan. As the user generates a flight plan, information regarding current and predicted weather conditions, traffic density, restrictions and status of Special Use Airspace (SUA) is available. Prepared routes can be checked against these conditions and any potential problems can then be reconciled by the user before the flight plan is filed.

Improved NAS Demand Data

The management of the NAS is influenced by the availability of runways, the critical resource that limits system throughput in 2002. In most instances today, demand at the major airports is defined by the Official Airline Guide (OAG) schedule, not the actual daily schedules planned by the users. During FFP1, the Air Traffic Control System Command Center (ATCSCC) and AOCs use improved decision support tools to revise the status of active and proposed flights to reflect more realistic schedule times (i.e., the latest planned departure times), resulting in more accurate predictions of traffic load, and increased flexibility due to the imposition of fewer flow restrictions.

Collaborative Flight Planning

AOCs are participants in the resolution process when certain traffic flow initiatives are required. When ATCSCC must implement a ground delay program or exercise the use of alternate arrival and departure routes due to severe weather problems, AOCs are consulted for their inputs. For example, as the ATCSCC polls the participating AOCs for flight schedule reductions to alleviate an airport traffic flow problem, they might obtain sufficient voluntary reduction such that no further action is required. Otherwise, an enhanced ground delay program, or capacity management program, is implemented using an approach called “ration by schedule” (RBS). This approach uses the OAG schedule to allocate arrival slots for the air carriers at the affected airport and is commonly referred to as “control by time of arrival (CTA).” Use of the OAG schedule allows the airlines to receive credit for their voluntary actions to reduce the demand at the airport. For scheduled air carriers, this approach preserves the desired arrival order and reduces bank disruptions at hub airports, giving the AOC greater control over their operations. For users without published schedules, flights are treated in 2002 in the same manner that they are today.

Availability of this flight planning information, coupled with NAS status information, facilitates more effective collaborative decision making between the AOC and ATM. This increased collaboration and information exchange provides a more accurate baseline for estimating system demand.”

These anticipated operational improvements can only be achieved through the use of commonly defined data that are augmented by a common flight data model.

Section 3

3. Flight Data Model

The flight data model has its origin in recent work for the AUA-500 organization in developing a common data environment for TFM. This work lead to the creation of an inclusive set of NAS data categories, of which flight data was one (see Table 1-1). This work also included the analysis of several key TFM systems and the mapping of the data requirements to each of the data categories. In this way, all of the flight data per system was discovered and associated with a flight attribute from the flight, or demand, category. This mapping was used to create a proposed standard set of data elements per category and a data model, as in the case for flight data, to define the relationships among the data and as a requirement for the exchange of such data at the interfaces between systems.

3.1 Data Requirements

Staff from CAASD compiled the data attributes for the TFM applications listed in Section 1.3.2 above and organized them by application program and by the seven data categories described above. Table 3-1 displays the element count by application program. Each of the elements referenced in the table was named and described in a Microsoft Access database. For this analysis, only the flight data elements (category 1: demand) have been addressed. It is anticipated that additional modeling efforts will cover the other categories, with a focus on capacity and NAS resources (category 2: capacity). These data were subsequently organized into common NAS data categories, as described in the next section.

Table 3-1. Application Programs Element Count

System Acronym	System / Program Name	Data Element Count
NFDC/ AIS	National Flight Data Center/ Aeronautical Information System	1159
CDM	Collaborative Decision Making	53
CTAS/ TMA	Center TRACON Automation System/ Traffic Management Advisor	81
CTAS/ pFAST	Center TRACON Automation System/ passive Final Approach Spacing Tool	97
DOTS	Dynamic Ocean Track System	101
ETMS - (hub)	Enhanced Traffic Management System (data flow to/from hub facility)	124
ETMS - (site)	Enhanced Traffic Management System (data flow to/from fielded sites)	61

FSM	Flight Schedule Monitor	5
HCS	Host Computer System	198
NFDC/ NOTAM	National Flight Data Center/ Notice To Airmen	110
OAG	Official Airline Guide	38
OCS	Operational Control Segment	3
RTCA	RTCA Special Committee (SC) on Free Flight Implementation	166
SMA	Surface Movement Advisor	15
TFM	TFM R&D Programs	73
TFM-ART	Traffic Flow Management – Architecture Requirements Team	346
URET	User Request Evaluation Tool	465
	Total Number of Data Elements	**3095**

3.2 Data Organization

A set of common NAS data categories were developed to provide a framework for organizing flight-related data elements from current TFM systems that were analyzed as a foundation for developing a flight data model. By organizing data in this fashion, CAASD decomposed NAS data into sets of standard data elements by logical groupings. These categories, shown below, are open to modification as analysis of data requirements proceeds.[12] There are currently seven primary NAS data categories, each of which has several sub-categories defined to the entity level. These categories are:

- **Demand,** or flight data, includes information such as the flight itinerary, flight identification, flight planning, flight events and status (including position reports), and ATM control events that affect a single flight.

- **Capacity,** or resource data, describes static resources, such as airports, runways, and airspace, as well as their dynamic status, such as configuration, current capacities, and activation.

- **Weather** data include ground, satellite, and airborne weather observations, forecasts, and reports of weather phenomena.

- **Traffic management** data describe situations in which capacity exceeds resources, and actions taken by ATC, TFM, and users to resolve these imbalances.

- **General resources** are data that are not NAS-specific, such as time, geography, and geopolitical data.

[12] The detailed categorization of flight data from the flight data model is found as Table 3-3. The categories themselves are derived from the cross-agency TFM-ART activity.

- **Performance** data are those used to describe NAS operational effectiveness and its ability to meet user needs.

- **Miscellaneous** data elements are those that do not fit under the other major categories. Conceivably, new major categories could be identified and defined from this set.

For context, Table 3-2 displays a data attribute count per data category, showing that over 25% of the attributes are flight category-related.

Table 3-2. Element Count by Data Category

Category Number	**Data Element Category Name**	**Data Elements in Category**
1	Demand (Flight)	831
2	Capacity (Resources)	1624
3	Weather	76
4	Traffic Management	319
5	General Resource	19
6	Performance	34
7	Miscellaneous	7
—	Elements not placed in a category	185
	Total Number of Elements	**3095**

For its initial analysis, CAASD focused on demand data — pertaining to flight; that is, flight schedules, flight plans, flight progress, and related aircraft data. This focus was chosen to address the Air Traffic Service (ATS) Concept of Operations [5] for the NAS in 2005, which identifies a flight data thread in the NAS-wide information system, providing "information on each flight from the moment of push-back to wheels-up, including surveillance data in flight, touchdown time and gate assignment."[13] Table 3-3 provides additional detail about flight data subcategories proposed by CAASD.

[13] *ATS Concept of Operations*, Federal Aviation Administration, 1997

Table 3-3. TFM Demand Data Categorization

Category Number	Category Name	Description
1.	**Demand (Flight)**	Data required by NAS users and operators to describe, manage, and control the safe movement of aircraft in the NAS. Much of this data is associated with one flight, in which a flight normally includes one take-off and a subsequent landing.
1.1.	Flight Itinerary	Description of an aircraft operation involving multiple takeoffs and landings.
1.2.	Flight Operator	The person(s) or organization(s) responsible for the operation of a flight.
1.3.	Flight Identification	Unique flight identifier, unique NAS-wide
1.4.	Flight as Planned	Data needed to describe a flight to be made at some future time. May include alternatives and preference from many sources (e.g., users, ATC, TFM, automation, policy and procedures).
1.4.1.	Route Preference	Preferred route description from take-off to landing
1.4.2.	Departure Preferences	Preflight and departure operational preferences
1.4.3.	Arrival Preferences	Arrival operational preferences
1.4.4.	En Route Preferences	En route operational preferences
1.4.5.	Descent Preferences	Descent profile preferences
1.4.6.	Diversion Preferences	Alternative airport preferences for IFR flights
1.4.7.	Flight Plan Amendments	Record of changes made to an approved flight plan
1.5.	Flight as Flown	Description of the flight as flown, including measured parameters and actual times of events
1.5.1.	Flight position report	Measured or estimated position of an aircraft from a surveillance system
1.5.2.	Flight event	Flight activity associated with a specific event
1.5.2.1.	Flight path event	Description and timing of an event associated with aircraft movements, (e.g. wheels up).
1.5.2.2.	ATM control events	Description and timing of an air traffic management action taken that affects flights.
1.5.2.2.1.	Single flight control	Description and timing of an air traffic management action (e.g. controller action) that affects a single flight.

1.5.2.2.2.	Traffic management strategy	Description and timing of an air traffic management action (e.g. TFM constraint) that affects multiple flights.
1.5.3.	Flight status	Description of dynamic flight parameters or state variables (e.g. velocity, altitude, position, assigned beacon code, etc.).
1.5.4.	User fee	Data needed to assign user fees
1.6.	Flight as Forecast	Predicted flight parameters and predicted times for events
1.6.1.	Flight position predicted	Data describing the predicted location of an aircraft.
1.6.2.	Flight event predicted	Predicted time of a flight event
1.7.	Aircraft	Description of the aircraft used for a flight
1.7.1.	Aircraft description	Static data identifying and describing an aircraft
1.7.2.	Aircraft status	Dynamic data (e.g., fuel on board) affecting how a flight is flown
1.7.3.	Aircraft equipage	Description of equipment (e.g. navigation systems) that affects how and where a flight is flown

Table 3-4 contains the distribution of data elements within the flight category from the set of applications examined, specified by the first sublevel in that category. Again, these counts are only from the selected TFM systems described above. These counts will be higher if additional systems are included. It is these attributes, after being mapped to a

Table 3-4. Flight Data Element Distribution

Category Number	Data Element Category Name	Data Elements in Category
1.0	Demand (Flight)	24
1.1	Flight Itinerary	26
1.2	Flight Operator	22
1.3	Flight Identification	105
1.4	Flight as Planned	216
1.5	Flight as Flown	238
1.6	Flight as Forecast	125
1.7	Aircraft	74
	Total Number of Flight Elements	**831**

more compact standard reference set, that can be most closely associated with the attributes in the flight data model.[14] This is an activity proposed to follow discussion of the flight model and agreement about its use.

3.3 Common Flight Data Structures

Many NAS applications use flight information. Increasingly, to develop the ATS Operational Concept, the Free Flight Phase 1 Operational Concept, and to build systems faster and cheaper and with greater interoperability, NAS systems will be required to be built from commonly accepted data structures. This does not prevent them from customizing the these data or the view of these data for application-specific needs. However, common data structures will make many information management tasks easier, such as data exchange and application maintenance. The flight data model described in this document is a structured way to represent flight data to assist with the management of such data in any context in which it is required.

The flight data model has been in direct support of TFM. Its design is generic and is applicable to any system requiring flight data. Most of these data, such as flight schedules, flight intent, aircraft and operator descriptions, are generated by the air carriers. Other demand data, such as data about flights in progress and forecast flight data, are generated by the ATC system via the Host Computer System (HCS) and applications such as URET, TMA, and FAST.

Flight data need to be shared throughout the NAS. For ATC and TFM, these data need to be shared and accessed in real-time and near-real time by both FAA service providers and Traffic Management Coordinators (TMCs) but also by AOCs and pilots. System developers require that flight data be shared in terms of their data structure and the data associated with it. FFP1 requires that the flight data model structure and relationships represent current and future operational procedures and requirements. These needs can be accommodated by the flight data model presented here.

Another way to view the flight data model is to consider its applications across various phases of flight, as shown in Figure 3-1. This view may be useful in designing applications within each phase in terms of data structure, content, data provider, data recipient, and desired capability. These phases also have correspondence to the several NAS domains (e.g., En Route, Terminal, TFM, CNS, Oceanic) in which NAS capabilities are being implemented.

---

[14] This initial flight data model does not necessarily use the exact names of the legacy flight data elements or of the more standardized flight data attributes to which the legacy attributes were mapped. That process is ongoing and is part of completing a data naming standard.

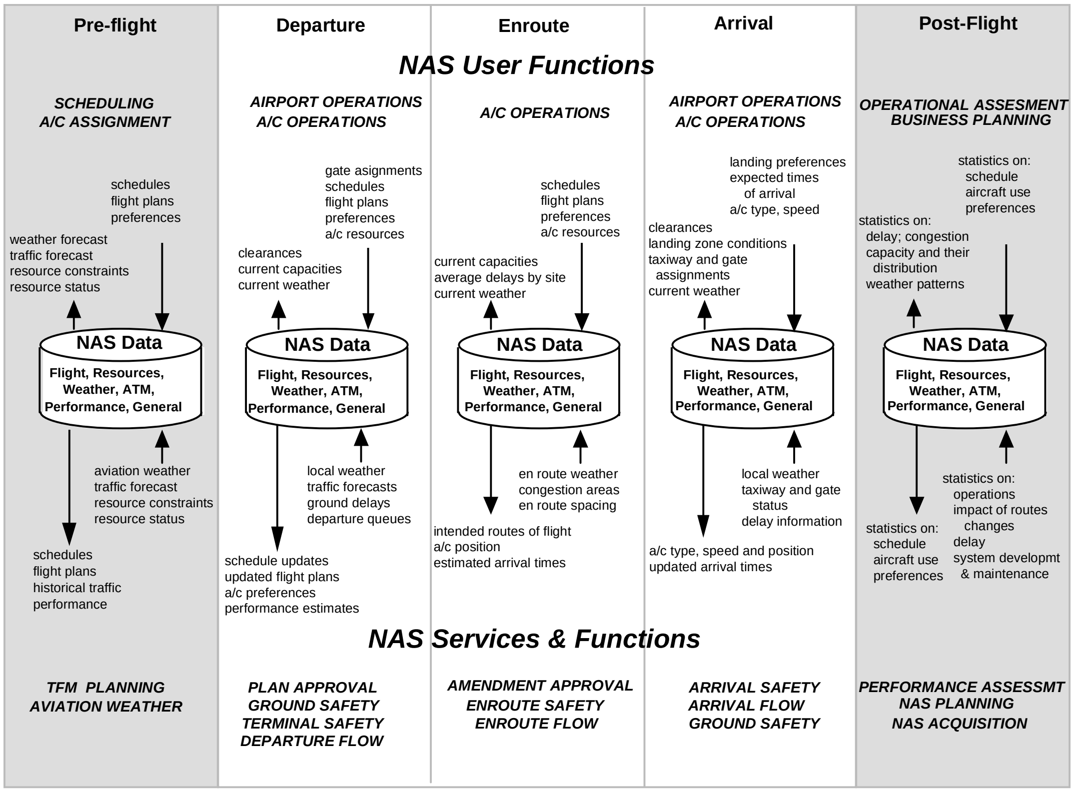


Figure 3-1. Data Support for a Phase of Flight View

3.4 Overview of Flight Data Flow

Generally, NAS users initiate information about flights and airspace demand (category 1) but with strong collaboration with the FAA while the FAA manages information about NAS resources and capacity (category 2). Figure 3-1 depicts a high level view of the flow of these types of information between NAS users and the FAA. This view, which represents both current practice and proposed future enhancements, is the basis for the information represented in the flight data model.

Figure 3-2 is a variation of the phase of flight view above in that it takes a time-based process view of the management of flight data.[15]

15 Based on a discussion in RTCA, DO-241 cited above.

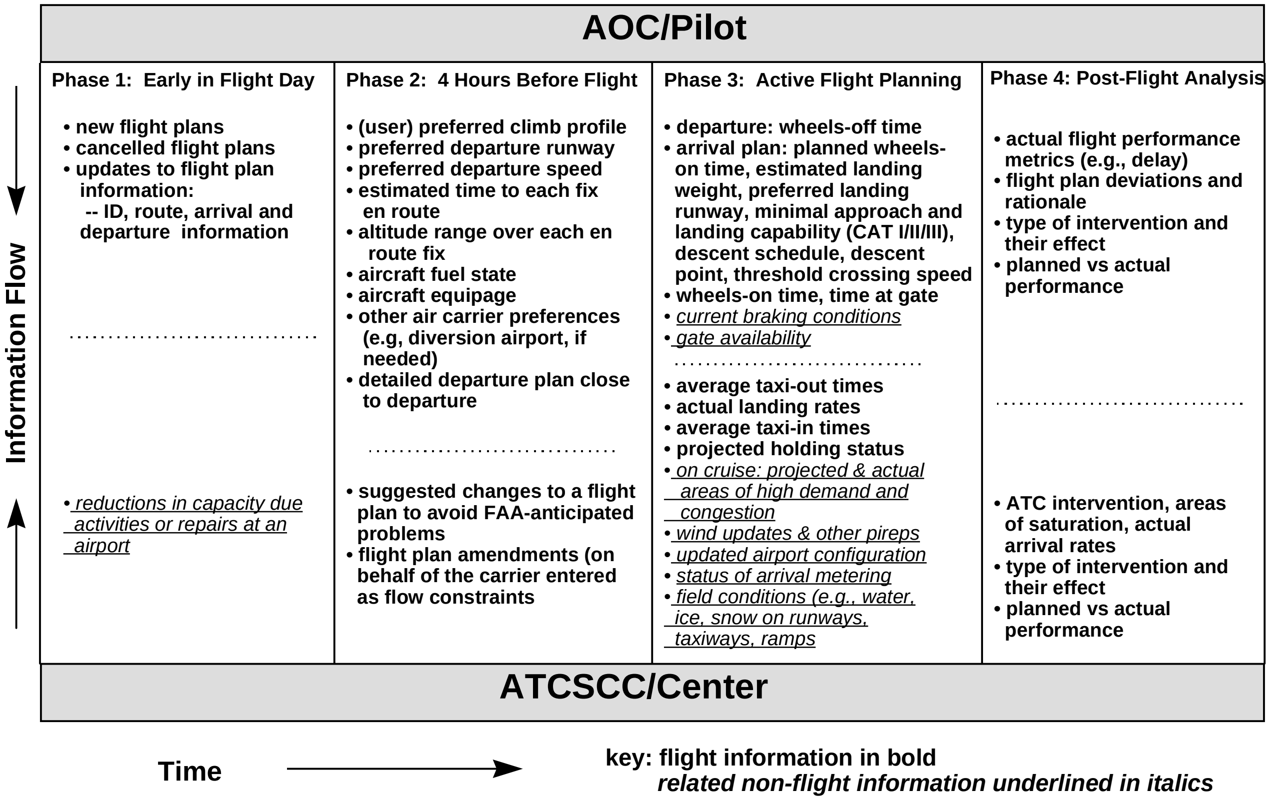


Figure 3-2. Process-based Overview of Flight Data Management

3.5 Flight Data Model Structure

The flight data model is an entity relationship model focused on flight. Its structure centers on data uniquely identified for each flight that are associated with several aspects, or event categories, about a flight. These event categories are flight planning; flight events that change the parameters of a flight, such as altitude, speed, and location; and ATM events, some of which are ATC events such as departure and arrival control events and others of which are TFM events, such as ground delays and metering to control traffic flows. In addition, the model also captures description and dynamic information about the aircraft assigned to a flight.

The model also contains links to airspace structure data that are related to flight, such as routes, fixes, NAVAIDs, and other basic aeronautical data, but the model does not attempt to address these airspace structures in detail. Rather, it recognizes that other data models already exist or will be developed to represent the detailed relationships in these areas, and that they will be integrated with the flight data model in the future. Therefore, the description of the flight data model that follows focuses only on data attributes that are directly related to the attributes in category 1, namely demand, or flight, data.

In summary, there are several concepts represented in the model, as shown in Figure 3-3. Event type is the category discriminator. These are:

- Flight Plan events, starting from initial flight schedules and plans that may be modified, even as the flight is in progress
- Dynamic Aircraft events, which trace the status of a flight after wheels off with a focus on aircraft position and other parameters about the aircraft, such as speed, climb rates, and fuel usage
- Air Traffic Management (ATM) events, which capture changes in control and alert situations and in traffic flow management
- Aircraft descriptors and an operator description

The model represents many flight events. However, it does not intend to be inclusive since there are some flight events that are not directly modeled in this first design.

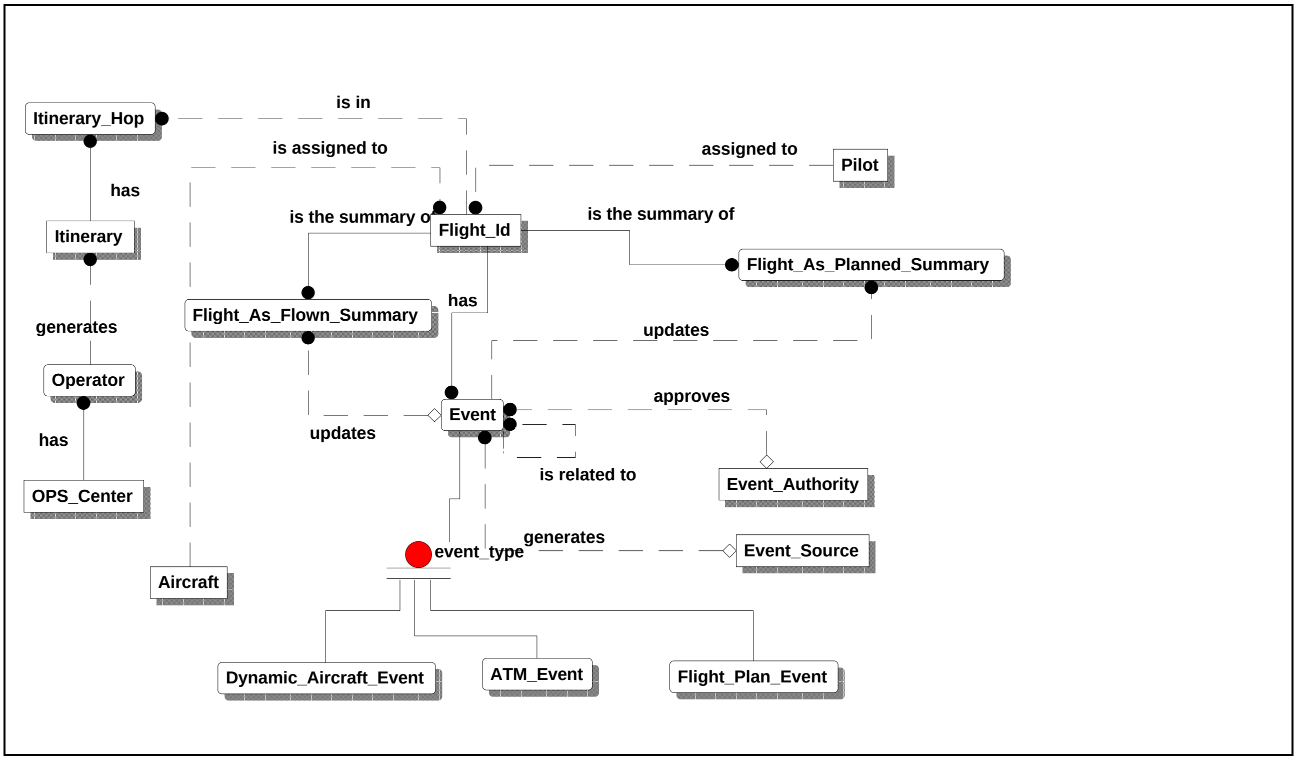


Figure 3-3. Top-level View of the Flight Data Model

3.5.1 Model Features

There are many views of flight that could be represented by the model. This model seeks to maintain the features of the current processing environment that are required for flight processing while supplementing it in two ways: 1) with features known to be current requirements and 2) with probable requirements of future systems. The model permits or facilitates the implementation of these features. It does not guarantee that any or all of these features will be present because that depends on how the model is implemented.

The features of the model include:

- A uniquely identified flight based on a system-generated identifier. The identifier is system-generated because there is currently no unique flight identifier in use. Also, this allows the model to be used internationally.

- A flight itinerary as a set of flights

- A 4-dimensional view of flight position, e.g., typically referred to as 'x,y,z,t,' that includes the three-dimensional spatial coordinates (x,y,z) and the time (t) at that position. Today's flight plans establish information in the first two of these dimensions with limited altitude and time information.

- Numerous variations of flight plans, including proposed, active, cancelled, and trial plans for simulation, testing, and operational decision making. These variations are captured in domain [16] values rather than in the explicit model structure.

- Compatibility with an ICAO flight plan and with a free flight 'New Age' flight plan

- Audit trail data to allow searching on 'who,' 'what,' and 'when' regarding a flight. This may assist in answering 'why' an event happened.

- A data archive to capture flight events to enable an analysis of these events and the discovery of flight patterns that could lead to improved safety, efficiency, or both

- Descriptive and dynamic aircraft information, including equipage, fuel use, and operating characteristics

- Climb and descent profiles: actual and preferred

- Forecasted position (i.e., trajectory) information

- Flight data that are derivable if it is not stored directly, e.g., wheels off and wheels on times as a function of landing gear position and pressure

[16] The term 'domain,' when used in a data modeling context, refers to the set of allowable values that an attribute may take.

One of the pragmatic issues involving the data model is the extent to which the data it describes are available. This issue wasn't addressed directly. It is understood that some of the data in the model are not acquired at present. Some of these data may, in fact, be very difficult or expensive to capture or may depend on future components of the NAS, such as extensive air-ground data link. The model considers this issue to some extent, but it is intended that a review of the model will include the feasibility of including any particular data item.

3.5.2 Flight Identifier

A fundamental capability in data management is the ability to identify a particular instance of data unambiguously. This identifier is called a key. For instance, one's social security number uniquely identifies a person in the United States. However, that identifier would not suffice in developing a global data model since social security number is not used outside of this country. Even within the U.S., various system account numbers may duplicate an existing social security number.

Short of assigning everyone on the planet a unique identifier, another identifier — perhaps a wholly fictitious one — could be defined if a naturally unique identifier were not available. For example, one option would be to string together several identifiers, which, taken as a whole, would uniquely identify someone. In theory, a set consisting of {last name, first name, country of birth, birth date, mother's maiden name, favorite color} might well point to only one individual, if all of this information were available. It would be somewhat unwieldy, however, from a system management viewpoint.

Another option would be to generate an arbitrary, but unique, value using the automated system. In the case of flight, where there is a great deal of ambiguity in defining a unique flight, such an option was considered, together with naturally occurring flight identifiers.

One's first reaction for selecting a unique flight identifier might be to select the OAG flight number. However, there are several drawbacks with this choice. First, the same flight number may be in use by two flights simultaneously.[17] Second, flight number are reused day after day, so that, at a minimum, date is also required to achieve uniqueness. Third, an OAG flight is typically a flight itinerary rather than a single flight.[18] Fourth, the OAG has flights that never fly. Finally, the OAG is published only as a guide and contains only scheduled flights in the United States but not international, general aviation (GA), military, air taxi or air freight flights. Another complication is that some flights are known to the system only via track information. How should these be identified?

Another candidate key might be {aircraft tail number, departure airport, departure date, scheduled departure time}. This key could apply since it covers all aircraft. However,

[17] For example, while a flight can be en route and reporting Track Update messages, a Flight Plan message can be entered for the next leg of the flight (which shares the same flight number).

[18] A flight itinerary is a concatenated set of flights, or flight segments, in which a flight consists of one takeoff followed by a landing

tail number is sometimes not available in advance and may change when aircraft are substituted during the course of a flight itinerary.

3.5.2.1 Options for a Unique Flight Identifier

A system-generated flight identifier would be unique, controllable (by specifying the algorithm that creates the identifiers), always available, and universal, i.e., covers all flight types. A drawback is that such an arbitrary identifier contains no natural data and thus conveys no information about a particular flight.

Considering only flights by scheduled air carriers for the moment, an alternative, naturally-occurring flight identifier might consist of the following set of identifiers: {air carrier identifier, departure airport, arrival airport, departure date [in mmddyyyy (month/day/year) format], scheduled departure time [in hhmmss (hours/minutes/seconds) format]}. An instance of this format might appear as:

DAL||STL||ATL||03211998||074300

Arrival airport is included because this identifier might not be unique without this added descriptor since an air carrier may have several flights scheduled for the same departure time at an airport but with different destinations. This assumes that time is *scheduled* departure time. Were time defined as *actual* departure time, with accuracy in seconds, then this compound identifier would not require arrival airport. However, it would require a more refined data capture process that could acquire actual departure time in real-time to create a database record based on this data item**.**

Another modification to the sample flight id above would be to include the flight number, as follows:

DAL||123||STL||ATL||03211998||074300

In this case, it is likely that one or more of the fields in this string could be eliminated, e.g., destination airport, date and time, if it is safe to assume that no two flights between the same city pairs on the same air carrier would have the same flight number. It is not clear, however, that this is the case. In addition, a flight number may apply to an itinerary, i.e., a set of flights (takeoffs/landings), rather than to just one flight and, therefore, could not be used in the unique flight id.

The Host associates Center identifier (CID) (e.g., ZKC: Kansas City Center) with a flight. It would be part of the key were airport identifier not unique in the CONUS. Since airports are uniquely identified, CID can become a descriptor, or attribute, of a flight rather than part of its key. This is preferable since CID changes when an aircraft crosses into an adjacent Center. In theory, the value of a key should not change once it is established and having CID as part of the key would introduce such a complication. In addition, were CID part of the key, it would make the association of parts of the same flight more difficult as it crossed one or more Center boundaries.

Generalizing now to all types of flight, the sample composite key above would not apply to general aviation, military, business jet, air freight, and other non-scheduled flight types. To cover all of these types, the key would have to contain information such as pilot name or identifier, date and time of the flight plan filing, type of aircraft, and so on in addition to the universal information about departure and arrival airports or location. Uniquely identifier every type of flight is complicated, and there does not seem to be a convenient and pragmatic key available.

For this data model, therefore, a system-generated key was selected and is called the flight_unique_id.

3.5.2.2 Time

The management of time is a fundamental aspect of NAS operations and data modeling. In the flight data model, time is used in two basic ways. First, time represents when an event is scheduled to occur, such as a scheduled flight departure (e.g., fp_departure_time_gate in the Flight_as_Planned_Summary entity) or when it actually occurred (e.g., departure_time_gate in the Flight_as_Flown_Summary entity). The difference between a proposed time and an actual time is context dependent. The attribute ‘timestamp_event’ in the Event entity contains this information.

The second way in which time is used in the model is for audit trail purposes — to record when data were entered, i.e., when an event was acknowledged. These data are important to reconstruct a flow of events, if that becomes necessary. For all events, the attribute ‘timestamp_recorded’ in the Event entity captures the time when an event is recorded in the database. The Event_Authority and Event_Source entities manage additional audit trail detail.

A standard for representing time is UTC (Universal Time Coordinate). This is the time implicit in the timestamp attributes and in those attributes that reference time, such as ‘departure_gate_time.’ UTC is considered as local time. In practice in the NAS today, however, the use of time is not so straightforward since time may be local or Zulu (referenced on the Greenwich time standard, which is equivalent to UTC).

Within some ARTCCs, there may be as many as three variants of local time, e.g., in the Indianapolis Center, Eastern Standard Time (EST), Eastern Daylight Time (EDT), and Central Daylight Time (CDT) all co-exist at the same time at different locations within the Center’s boundary. In addition, most NAS systems have internal time generators that are not synchronized. Knowing which time is appropriate and converting among these variants is a significant operational issue.

There is presently no recognized standard for representing time in the NAS. One goal of this model activity is to formulate and propose such a standard. Establishing reliable clock synchronization is also a primary system requirement.

3.5.2.3 Position

Another important type of data in the model is position, or location. This applies primarily to aircraft position (using the four dimensions of latitude, longitude, altitude, and time) but also to 2-dimensional locations for ground-based facilities, such as airports and NAVAIDs.

Table 3-5 illustrates the variety of formats now in use to represent a 4-dimensional aircraft position. In addition, geographic coordinate projections for latitude and longitude are not consistently represented. It is not difficult to appreciate the many drawbacks of this current practice. Primarily, safety may be compromised by a misinterpretation or incorrect translation of position data from one system to another. Also, the time and expense to maintain system interfaces among the many systems exchanging position information is considerable.

Just as with time, it is a goal of this model to propose a common, internationally referenced standard for the description of position data in the NAS. With any standard, accommodation must be made to each of the existing formats to be sure that the underlying data is available wherever they are needed, notwithstanding the need to offer translation services among the differing format implementations. With the existence of a standard, variations in meaning (semantics) and format will be reduced over time.

Table 3-5. Format Variants in NAS Applications for Aircraft Position

System	Definition of Location (x,y,z,t)	Example
ETMS	Time ::= HHMMSS Latitude ::= dddd[N|S] Longitude ::= ddddd[E|W] Altitude ::= ddd (100 ft) (reported) (d)dd (100 ft) | [d]ddB[d]dd (altitude/flt level block) | ABV/[d]dd (above specified altitude) | VFR/[d]dd (VFR + altitude) | other fmt	020845 4258N 08626W 070 or 70 or 70B190 or ABV/125 or VFR/125
ARINC (varies by airline)	Time ::= YYYYMMDD.HHMMSS000 Latitude ::= [N|S]dddd Longitude ::= [E|W]dddd Altitude ::= ddd (ft)	19960917.020845161 N4258 W08626 070
ARTS IIIA	Time ::= HH:MM:SS.SSS Latitude ::+ [-]dd.dd (nmi) Longitude ::= [-]dd.dd (nmi) Altitude ::= dddd (ft)	02:08:45.161 12.65 -8.56 0752
ARTS IIIE 6.04 (TRACON)	Time ::+ HH:MM:SS:SSS Latitude :: [+|-]dd.dd (nmi) Longitude ::= [+|-]dd.dd (nmi) Altitude ::= ddddd (ft)	02:08:45:161 +12.65 -8.56 00752
SAR (translated)	Time ::= hh:mm:ss Latitude ::= ddd.ddddddd Longitude ::= ddd.ddddddd Altitude ::= ?	02:08:45 012.6500000 120.2494306
NIIS	Location ::=Geographical Coordinates GeographicCordinates ::= {CoordinatesType type, GeographicCoordinatesType, CoordinatesTypeGraphicString}. GeographicCoordinatesType ::+ 0 (v-h-coordinates) | (latitude-longitude) | 2 (npa-nxx) | 3 (country-city)	{n, x....x}

3.5.3 Flight Data Model Events

The flight data model is structured around three classes of events that follow from the unique identification of a flight — flight plan events, ATM events, and dynamic aircraft events. ATM events are further decomposed into ATC events and TFM events.

The distinctions among these three classes are fairly clear but not absolute in that different persons might reasonably classify the same event in different event classes. More important, however, is that the model clearly explain which events are classified together and that these data are important for managing flight-related applications. In any case, the model is centered on the identification of a unique flight and a unique event, which is also identified by a system-generated event identifier, named event_id.

Three event classes are modeled as subcategories of an event, with event_type as the subcategory discriminator. Each event is time-stamped. Audit data are also present for event traceability and include the authority for the event and its source. These attributes are present to answer the questions: 'what,' 'who,' 'where,' 'when,', and 'why' about each flight-related event.

Two additional entities are linked as child entities, or sub-entities,[19] to the Flight_Id entity. They are treated as summary records for flight plan events and ATM events and are named Flight_as_Planned_Summary and Flight_as_Flown_Summary. Conceptually, they are updated dynamically whenever related attributes in a Dynamic_Aircraft_Event, an ATM_Event, or a Flight_Plan_Event entity changes. In implementation, this may be done by a process or a trigger rather than directly by the model. At the completion of a flight, they would contain key summary data about the flight plan, i.e., its last view, and about the flight itself, as it was flown. The audit process can capture any and all intermediate data about each of these views. Additional attributes can be added to either entity, as needed, to enlarge the scope of information about either the flight planning process or the actual flight.

3.5.3.1 Flight Plan Events

This section describes flight plan events, as shown in Figure 3-4. These events involve only the flight plan process, starting with the filing of a flight plan, and include the many variations and changes of a flight plan. Using the entity names in the flight data model, these events are:

- **FP_Filing:** A flight plan filing. The filing contains basic flight information, especially origin and destination airports for the flight, plus cruising airspeed, time en route, actual departure and arrival times, and gate information. In addition, the flight_status attribute tracks the flight plan status. The value can be selected from the set ('filed', 'approved', 'activated', 'cancelled').

- **FP_Amendment:** Amending a flight plan, e.g., to change the route, part of the route, or various parameters about the route, such as altitude or cruise speed. All amendments are associated with the base flight plan using the unique flight identifier, called flight_unique_id.

  Within this view, there may be several types of amendments. The Route_Amendment entity covers the case where all or part of a route is changed. The Other_Amendment entity covers all non-route amendments, such as changes to filed airspeeds and altitudes.

- **FP_Substitution:** Substitution of a new flight for a currently scheduled flight

The model has links to route information but does not intend to cover such airspace structure data in detail. That will be covered by modeling NAS resources and capacity.

[19] In data model, a child entity is one that is dependent on another entity i.e., a parent entity.

However, the flight model shows references to key airspace structure entities to support route data, such as NAVAIDs, fixes, and other waypoints.

The attribute fp_event_type is the category discriminator for the Flight_Plan_Event entity. The attribute waypoint_type is the category discriminator for the Segment_Waypoint entity.

In addition to the FP_Filing entity, three additional entities contain supporting details. These are: 1) Flight_as_Planned_Summary, which contains the overview of a flight and the most recent changes to a plan, 2) Route_Structure, which contains a high level view of a flight's route, e.g., route_structure_id, route_type, and number_of_segments, and 3) Route_Segment, which contains the detail-level view of the route in terms of each segment and waypoint. Finally, Segment_Waypoint is the parent entity for entities that contain detailed waypoint information, e.g., about fixes and NAVAIDs.

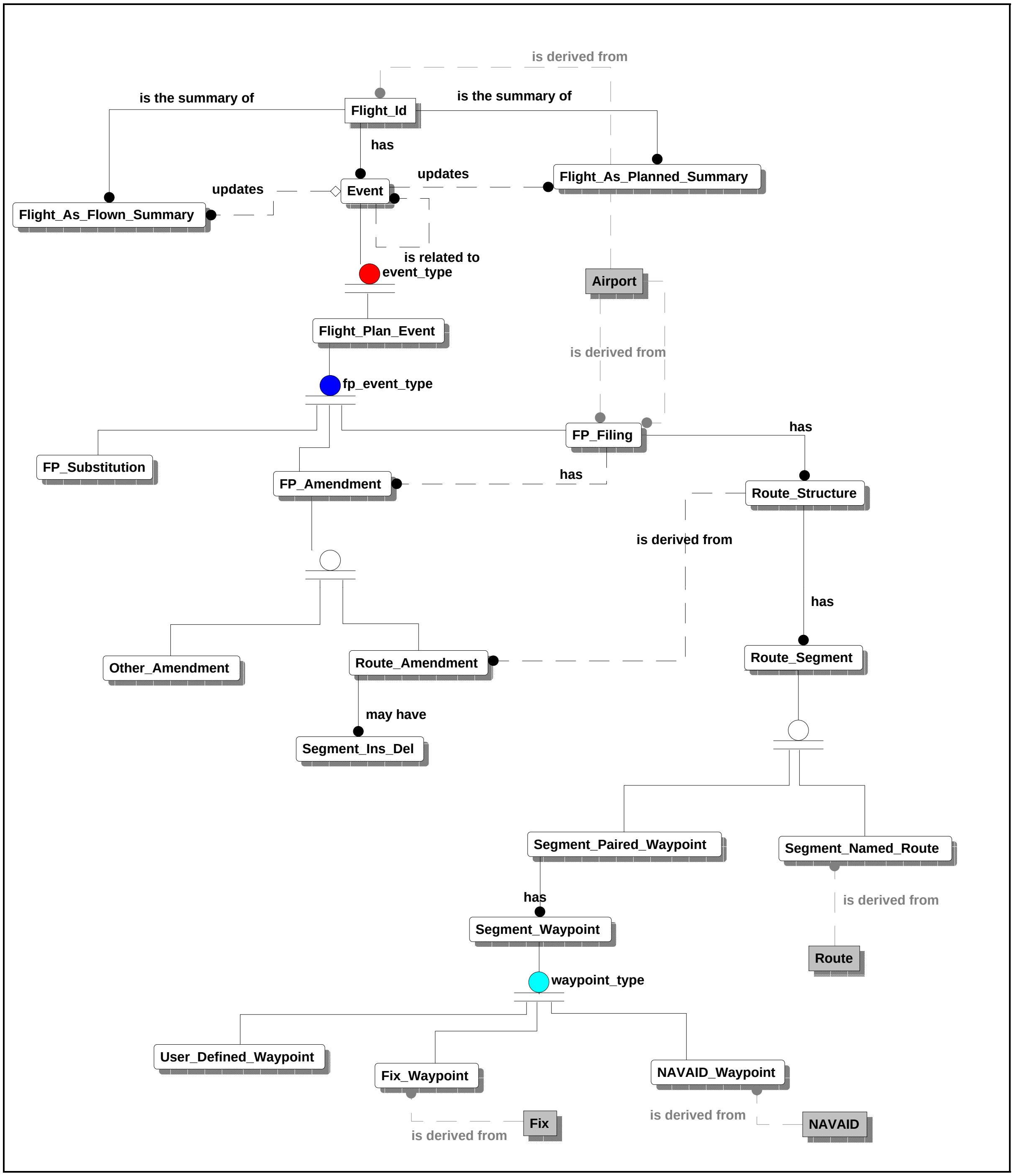


Figure 3-4. High-level View of Flight Plan Events

3.5.3.2 Dynamic Aircraft Events

A dynamic aircraft event is an event involving the aircraft's status, the on_board weather status, or an aircraft alert. Other events that could be considered as flight events, are defined in the model as ATM events, and are described in the following section.

As shown in Figure 3-5, dynamic aircraft event entities are:

- **Aircraft_Dynamics_1:** Constantly updated information about the aircraft's airspeed, position, pitch, yaw, roll, fuel burn rate, and weight. Updated aircraft position data can be obtained from radar readings from ARTS, STARS, ETMS or other system, typically en route.
- **Aircraft_Dynamics_2:** Additional data about the aircraft, such as phase of flight position, and data about the status of the landing gear, beacon, and mode C
- **Onboard_Weather:** Data taken from onboard weather sensors to measure wind speed and direction, the outside air temperature, and the barometric pressure
- **Aircraft_Alert:** Data about any of several aircraft alerts, including those generated by automation, such as TCAS, or by controller action

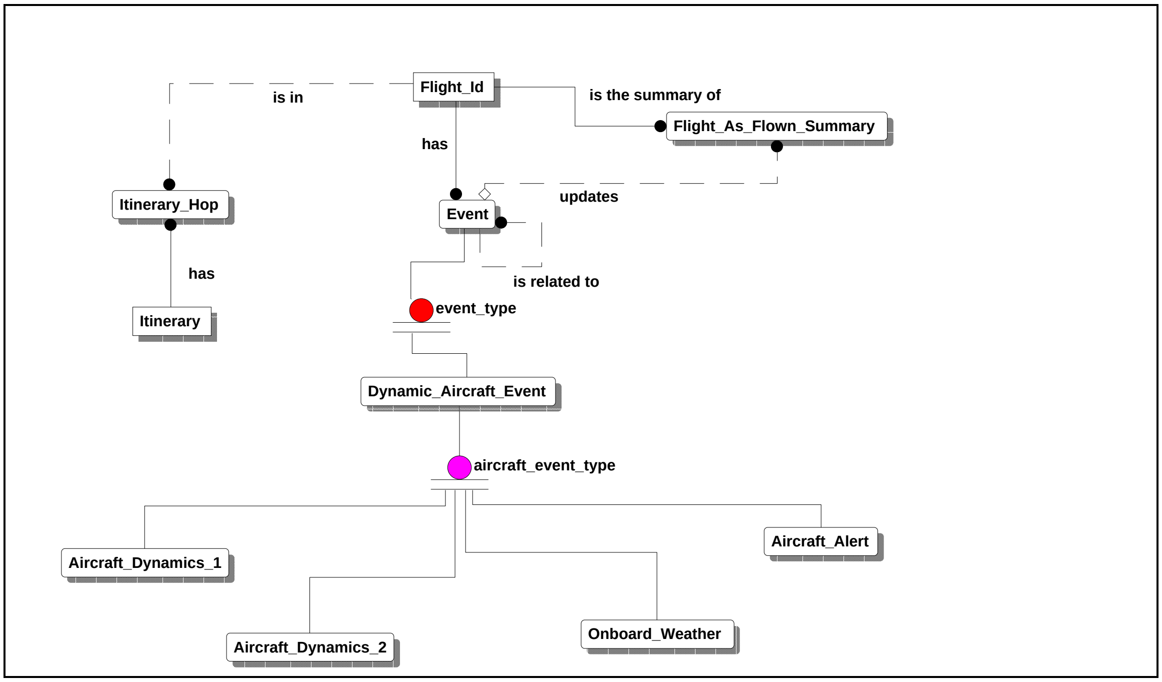


Figure 3-5. High-level View of Dynamic Aircraft Events

3.5.3.3 ATM Events

An ATM event is an event that is also associated with a flight, and could, therefore, be part of a flight event but an event that could also be classified in one of two other ways: Related to 1) aircraft separation and safety activity, e.g., Air Traffic Control, or ATC, or 2) a TFM event involving delay and traffic flow.

ATM events are subdivided into these two subevent types: ATC and TFM. Within each are a number of entities representing such events. The ATM category discriminator is atm_type while the subcategory discriminators are atc_event_type and tfm_event_type, respectively.

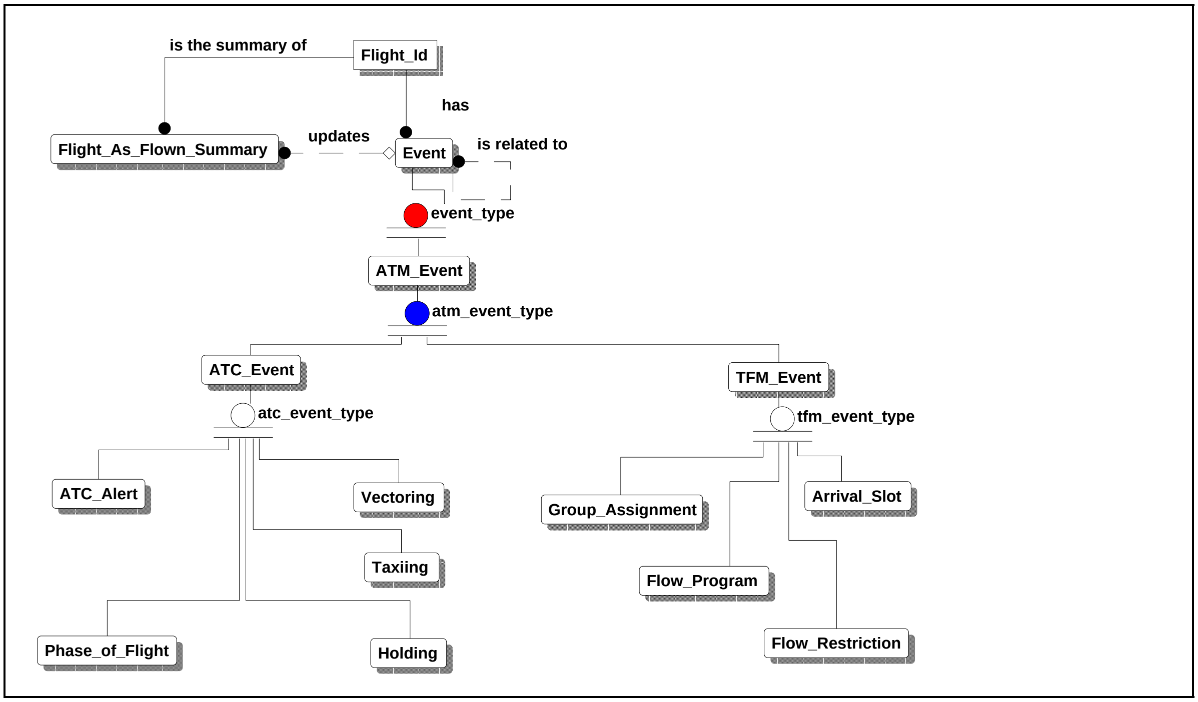


Figure 3-6. High-level View of ATM Events

These events involve events of a flight, from ground-based events prior to pushback and takeoff, through en route airspace, to arrival at a gate at a destination airport. They can be viewed in terms of a phase-of-flight view. The flight events are:

- ATC event entities, which consist of:
  - **ATC_Alert:** Data about the varieties of ATC alerts, such as a TCAS alert
  - **Change_in_Control:** Data about changes in aircraft control, primarily Center-to-Center handoffs
  - **Phase_of_Flight:** Data about phase of flight activity, including time of push-back, wheels off, wheels on, and gate arrival plus taxiing time at the origin and destination airports

- **Vectoring:** The assignment of a new heading to the flight
- **Taxiway:** Data about taxiway time and name and gate information at the departure and arrival airports
- **Holding:** Data about the assignment of an airborne hold, including location and holding time

- TFM event entities, which consist of:

  - **Group_Assignment:** The association of a number of flights as a group, e.g., as a bank, for a managed arrival reservoir (MAR), for purposes of managing severe weather, or for military flights
  - **Arrival_Slot:** The time assigned to a flight at which it is intended to arrive at its destination airport
  - **Flow_Program:** Identification of a TFM program, such as a ground delay or ground stop, to manage traffic
  - **Flow_Restriction:** Identification of a TFM restriction for a flight, such as metering, miles-in-trail, weather, and special use airspace restrictions

3.5.4 Aircraft Data

Another class of data in the model is aircraft data. Unlike the dynamic aircraft events, these data are fairly static. This class includes not only data about the particular aircraft (Aircraft entity) on a flight, but also data about the operation of the aircraft. The Aircraft entity contain the basic descriptors of the aircraft and is keyed on aircraft_tail_number, which is considered a unique identifier. If, in practice, this is not the case, a unique system-generated key can be produced, as is done for the flight id. This file also contains the aircraft's manufacturer and serial number, an alternate primary key pair. It also contains other attributes, such as aircraft type, weight class, engine class, registration, endurance, take-off weight, payload weight, and cruise speed.

Additional entities offer more specialized structures for managing data about the aircraft. As depicted in Figure 3-7, these entities are:

- **Manufacturer:** Coded and descriptive information about the aircraft's manufacturer
- **Engine_Class:** Coded and descriptive information about the aircraft's engine
- **Weight_Class:** Coded and descriptive information about the aircraft's weight class
- **Aircraft_Type:** Coded and descriptive information about the aircraft's type
- **Aircraft_Category:** Coded and descriptive information about the aircraft's category

- **Aircraft_Nav_Equipment:** Descriptive information about the aircraft's navigation equipment
- **Aircraft_Comm_Equipment:** Descriptive information about the aircraft's communications equipment
- **Aircraft_Emergency_Equipment:** Descriptive information about the aircraft's emergency equipment

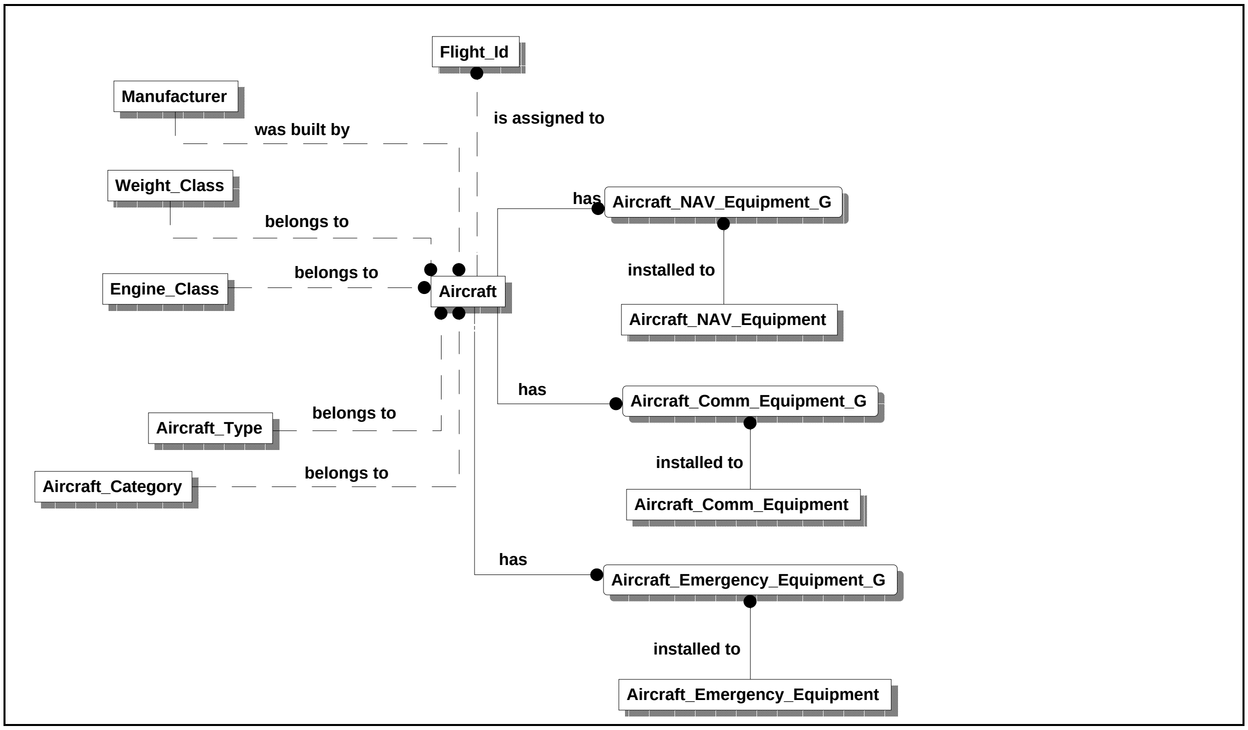


Figure 3-7. High-level View of Aircraft Data

3.5.5 Related Data Structures

The flight data model represents flight data, but, in expanding the model, one invariably links to information beyond flight data (category 1). For example, NAS resources and capacity (category 2) contains detailed data about NAS resources such as airports, routes, fixes, NAVAIDs, holding patterns, and such. They are represented in this model view on the periphery of the flight data model. In fact, they will also be modeled (many of these areas have already been modeled) to provide a complete view of NAS data.

In the way it is constructed, a model will tend to support some aspects of data processing while not supporting other aspects. A mitigating factor is that, as a conceptual model moves to its physical representation and then to implementation, some of these aspects can be diminished or enhanced, depending on what view the model has taken.

Since this flight data model is intended to be more universal than specific to a particular application, its constructions to support particular types of flight processes may or may not be present. This need not be a problem since the model can be modified at the logical level if a more specific target application requires customization. However, the definition and representation of certain data attributes are expected to become established requirements for data exchange across the NAS to ensure system interoperability.

Some of the considerations that went into the development of the current version of the flight data model are that it:

- Is anchored on a unique flight identifier. In the NAS today, there is no single unique flight identifier.

- Is based on sets of flight events, each of which is uniquely identified and retrievable.

- May be easily expanded to hold additional event categories and additional events within each category.

- Assumes that flight histories are desired and, thus, using a timestamp, allows the capture of a series of data that change over the course of a flight, e.g., accumulating position data, rather than simply overlaying new data over the last instance of that data.

- Assumes that audit trails are desired and builds in a basic set of audit trail elements, such as who modified flight information and where and when they were modified.

- Uses official NAS identifiers, such as fix, NAVAID, route, and airport id, as references for flight components. These identifiers may be found in the FAA's LOC-ID publication, which is updated every 56 days. Some of these identifiers may be modified to conform to ICAO standards.

- Uses entity and attribute names not related to any specific application. Ideally, after discussion, these names will be considered as a set of standard names to be used across the NAS. [A larger issue is establishing an office or function to manage issues such as these, an aspect of information management called Data Administration. In addition, an office or function is also needed to manage information security issues.]

Section 4

4. Uses of the Model and Implementation Issues

4.1 Flight Data Use

Two primary areas of impact for consistent and NAS-wide management of flight data are in 1) numerous areas of operational decision making and performance, and 2) system engineering and acquisition activities of system design, development, procurement, implementation, integration, and maintenance. These two areas are discussed below.

4.1.1 Operational Performance Impact

In a recent publication [20], the RTCA describes a set of operational scenarios that require a robust set of quality flight data. That description, based on overall mission planning, is incorporated in Figure 4-1 below and supplemented with additional aspects of flight data management.

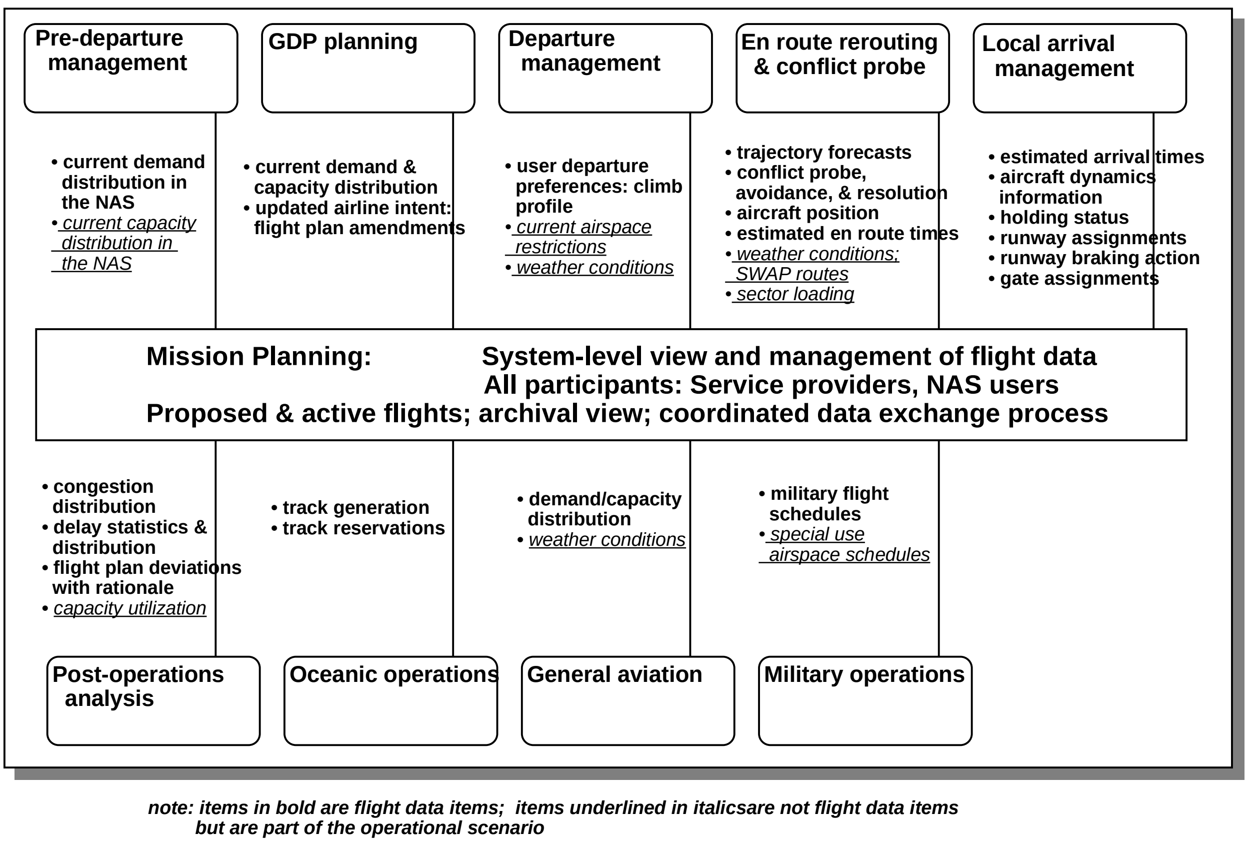


Figure 4-1. Set of Integrated Operational Scenarios Requiring Flight Data

[20] Op. cit.

The flight data model is proposed as a unifying structure across the applications that use flight data. In effect, each of these legacy systems has implemented its equivalent of what is proposed here, whether this was done with a formal data model and a COTS DBMS or simply by using various flight information structures as input to computer code.

The figure above implies the need for a near-real time and archival data resource (e.g., database, warehouse, repository) that is distributed across many users (and user classes), sites (land-based and airborne), organizations (FAA and industry), and time zones. A system is needed that can institute a fluid process of data capture and exchange and that can manage the control and distribution of flight information according to new rules that will be required to protect the confidentiality of these data.

The New Age Flight Plan envisions greater collaboration between the FAA and the industry based on the sharing of additional data previously known to one party but unavailable to the other. Examples include industry flight intent data, such as preferred climb and descent profiles, and NAS configuration data, such as airport acceptance rates, sector loading, and capacity constraint information.

The implementation of improvements to NAS flight processing requires more than just an upgrade to the set of flight information as captured by the flight data model described in this document. It also requires a number of modifications to the NAS. Although a detailed description of these modifications is not in the scope of this document, some of them are listed here:

- The roles and responsibilities of the various NAS decision makers, from traffic management coordinators to airline dispatchers

- The processes by which information is shared within the FAA and between the FAA and NAS users, such as the assignment of responsibility for the management and control of various subsets of flight data to people and organizations, and the synchronization of flight data updates

- The interoperability of the infrastructure that connects people and systems to allow a smooth flow of data across the NAS, e.g., single points of interface for access and delivery of flight data

- The sensitivity with which flight data is handled by all parties and relevant security issues involving access authorization, data integrity, and reliability of service.

- The internal structure of application systems (i.e., the automation system) that process flight data

4.2 The Data Model and Its Use

A model is an abstraction of reality. It represents reality in the way the data modelers see it. This is both a strength and a weakness of a model. This flight model, therefore, has its strengths and weaknesses as well, although different viewers might not agree on which are its strengths and which are its weaknesses.

First, this model does not represent any particular application system's view of flight. Second, the model incorporates views of flight from some major systems, such as Host and ETMS, but also accommodates views of flight not yet found in any application system, such as flight preferences and flight intent more closely associated with free flight. The model represents a number of flight events but does not claim to represent all flight events. The model is fairly detailed in the data it describes, but, given the complexity of the subject, does not represent that it has fairly captured every concept, every attribute, nor every business rule regarding flight.

This model has achieved, and to our knowledge for the first time, a synthesis of a large variety of flight data from many legacy applications, operational and developmental. It has also addressed flight standards and proposals for new flight data.

To reiterate some of the purposes stated at the beginning of this document:

- to suggest a unifying view of flight in which an essential structure and key characteristics can be agreed upon

- to consider such a view a basis for future system development, whether that involves modifying an existing system, re-designing an existing system, or developing a new system.

Another important purpose of the model is to be a focal point for discussion and to reflect views that have not been considered. In this sense, the model is a living entity ready to accept modification, enhancement, or a change in scope. Its value is to be used across the system development life cycle so that participants that conduct requirements analysis and software design could have the same understanding and use the same structure as organizations that perform software maintenance, performance tuning, and system interface implementation.

The model is in Entity Relationship (ER) format, which emphasizes a data-centered view of its subject and is a well-established standard for data modeling. As an ER model, data definition language (DDL) code can be generated routinely and easily that creates the flight model structure in a relational COTS DBMS. However, the way the model is constructed also permits translation to an object-oriented view as well.[21]

[21] A test was done to import the relational model from LogicWorks' ERwin product to Rational's Rose object-oriented modeling product using a software translator recently developed by the two

One way to manage the overall NAS data environment is to establish a NAS-level System Engineering Database in the Data Administrator area to manage the NAS metadata system-wide and for each system. Part of the metadata would include, in addition to the basics of attribute definition and characteristics (i.e., type and length), the source of the attribute, the set of all systems in which it is used, its criticality, its reliability, and other parameters. Such a database would support system acquisition and development as well as system maintenance. It would also be an input to the investment analysis process, helping to answer questions about the cost to acquire and maintain various sets of data, especially those purchased from third parties.

Regarding data availability, some of the data identified in the model either may not be available or may be available but proprietary. Until fairly recently, actual wheels off and wheels on data were only estimated from NAS radar tracking data. Now many such data (but not all) are available from the Aircraft Communications and Addressing Reporting System (ACARS), which records Out, Off, On, In (OOOI) data from aircraft sensors. An example of proprietary data is actual rate of fuel burn per flight, known by the airlines. Although manufacturer specifications would indicate approximate fuel burn rates per aircraft model type, the actual rates are deemed proprietary by the carriers and must be managed as sensitive data.

In addition, as more ground-based and airborne systems become more interoperable to achieve free flight, the opportunities for errors in data exchange at system interfaces will increase significantly. Without common data models there are no effective controls on the increased complexity in multi-system operations. The RTCA has stated that aviation safety will be improved by improvements in data sharing [1], but this assumes effective and error-free data exchange.

Potential outcomes. In practice, the FAA, by encouraging a common view of flight, can achieve system engineering efficiencies in several ways. Some of these are:

- There can be a dialog across organizations and across applications within the FAA about the meaning of flight data and how it will be managed structurally. This should naturally lead to discussions of how flight data itself (i.e., the instance data) should be managed.

- NAS users can participate and contribute to a common view of flight. The RTCA has already acted as a facilitator in this regard, and is, most likely, ready to continue this process.

- Consultants and development organizations can benefit from a NAS-wide view of flight and specific requirements for interoperability instead of being forced to

companies. The import was successful in that the basic structures were created in Rose but some effort was still needed to adjust the object-oriented view that Rose created. Additional effort is needed to sort out the pros and cons on this translation process.

develop their own 'local' view. In fact, several such organizations have already shown considerable interest in starting from a common FAA-approved view of flight to save time and money and develop to a view that is more likely to be accepted and interoperable.

- System maintenance organizations can streamline their work in upgrading the flight aspects of application systems by reusing code involving flight structures. Also, system interface process (e.g., message communications and processing) can be simplified by reducing the considerable pre-processing and translation that is required of messages at a receiving system that contain flight data unique to the sending system.

Implementation Issues. There are always many implementation issues in moving from a model to reality in the field. In terms of this flight model, some of the questions that naturally surface include:

- What FAA organization will 'adopt' or champion a flight model? This flight model? This raises the larger issue of the means of coordination among the numerous FAA organizations that have an interest in flight data management and their role in it. The NAS Information Architecture Committee (NIAC) has started this process.

- How will flight data be made available to decision-makers? Through existing applications or through newly-developed applications, views, and screen interfaces? What general query capability will be developed to offer *ad hoc* data access? How will policies regarding data availability include the variety of aviation data users, e.g., commercial, business jet, GA, military, air frieght, international, airport operations?

- What archival capability will be developed to store instance data for system restart and analysis of past operational patterns?

- How will the model be used by the various FAA application development organizations? Will it be used to interface with current systems, when current systems are redesigned, and/or when new systems are built?

- What is the role of the developer? To what extent will the FAA specify a structure for flight data to be used across the developer community?

- With the recent proposals for the development of common data servers at the Centers and TRACONs to offload some of the Host's processing, there is a need to agree on an implementation that simplifies, rather than complicates, the complex processing that now occurs among the applications at these facilities. The HADDS (Host Application Data Distribution System) and the TADDS (Terminal Application Data Distribution System) have been proposed as COTS-based data servers at the Centers and TRACONs, respectively. The Local Information Service (LIS) has been proposed in the NAS Architecture as a more general information service capability for Center-based application support and operational decision-making.

- What influence will NAS users have in the continuing design and development of a flight model? How will this affect implementation?

- Specific to the model, what is the number of views that will be maintained of the flight plan and the actual flight. For instance, will the several periodic snapshots captured during the flight plan process be accessible or will the system only reflect the latest view, and finally, the last view before no more updates are made to the plan?

Recommendations. The following are recommendations for developing and implementing the flight data model.

- The model, as published here, should be discussed with interested parties in the FAA and with NAS users to receive feedback about the model contents, structure, and impact.
- The model should be extended to cover additional events. Attribute values should be completely specified, and a set of business rules should be established.
- The model, as extended and enhanced, should be presented as a candidate standard for applications requiring flight data. This requires extending this logical model, in part, into a physical data model by specifying candidate formats (type and length) for each attribute.
- The use of the model should be explored as an interoperability aid for multiple systems that share, or will share, flight data. That is, its structure should be considered in messages that are exchanged between systems and as the basis for a database schema in applications that use a DBMS for data management.
- The model should considered as a core structure in the evolution of NAS flight data processing (FDP).

4.2.1.1 0.0.0.1 Adaptation Data

Major enhancements are needed to provide even a limited model for adaptation data. It may be possible to build upon the on-going work of the NAS Information Architecture Committee (NIAC) working group investigating opportunities for common adaptation data services for multiple domains and systems.

4.2.1.2 0.0.0.2 ARTS Data

The terminal domain data model strawman was built from an analysis of STARS data elements. But the ARTS data elements need to be equally well documented and analyzed in order to support the transition of terminal systems in an operational environment where both ARTS and STARS will exist together.

4.2.1.3 0.0.0.3 Terminal Trajectory and Route Data

A longer term data model development effort should be done to examine options for data structures and relationships needed to support the exchange of detailed route and detailed flight trajectories in the terminal environment.

4.2.2 0.0.1 Terminal Data Configuration Management

The terminal domain data model is just one small step toward the development and implementation of an effective, implementable, cost effective information architecture for the terminal domain. A long range goal of an information architecture is the use of common data standards. But even as we work toward this goal, the realities are that there is now, and probably always will be, a certain amount of diversity in data definition and usage. A set of procedures and tools is needed that can provide comprehensive, up-to-date information about terminal domain data and its usage. Other offices in the FAA are working to develop their own form of a comprehensive data configuration management system which may be a starting point for a terminal domain system. For example, ASY has developed a Data Encyclopedia System (DES) using commercial data modeling and data management tools for a flight safety data system. The NIAC is also sponsoring initiatives in the development of common data and metadata management tools.

APPENDIX A: ERwin Data Modeling Tool Conventions

ERwin/ERX is an Entity-Relationship (ER) logical data modeling product that runs under Windows. It supports the following objects which are important to information modeling: primary and foreign key identification; referential integrity constraints; entity, attribute, and relationship definition; business rules; foreign and alternate key definition; inversion entries. It can transform a logical data model into a physical schema and can generate Structured Query Language (SQL) Data Definition Language (DDL) scripts for many commercial relational DBMS products. In addition, it can also reverse engineer existing SQL DDL scripts into ER diagrams. Definitions for these terms can be found in the glossary.

The purpose of this appendix is to provide the reader with a basic understanding of the ER diagrams presented in this paper with examples from an aeronautical data context.

ERwin offers two choices of diagramming conventions that are widely recognized. One is Information Engineering (IE) notation and the other is called Chen, after the person who first proposed it. The flight data model currently uses Chen notation.

In an ERwin diagram, an entity is represented by a box with the name of the entity on top and all of the attributes of the entity inside the box as shown in figure A-1. The horizontal line in the entity box divides the attributes into two distinct sets: the area above the line is called the key area, and the area below the line is called the data area. The attribute or set of attributes that uniquely identify each instance of an entity is called the primary key.   Primary keys are placed above the line in the key area. All attributes in the data area are referred to as non-key attributes. Foreign key attributes are defined as primary key attributes of a parent entity contributed to a child across the relationship as shown in figure A-2. These attributes are designated in the diagram by an (FK) following the attribute name.

Figure A-1. Sample ERwin Entity

There are two types of foreign key relationships possible: identifying and non-identifying. In an identifying relationship, the foreign keys migrate to the key area. The relationship is called identifying because the keys from the parent form part of the identity of the child (i.e., the child is dependent on the parent for its identity). An example of such a relationship is illustrated in figure A-2. Identifying relationships are indicated by a solid line connecting the entities. The 'crow's feet' at the child entity indicates that the parent may create more than one child entity.

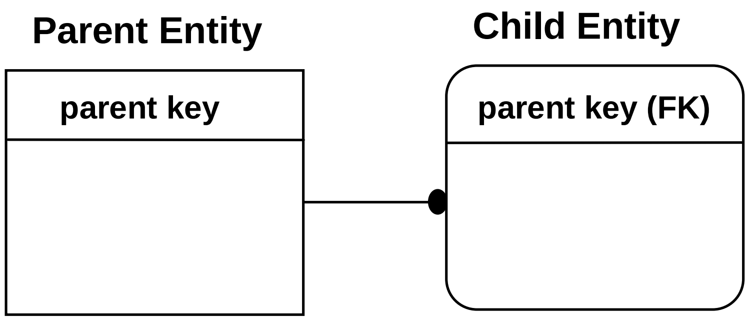


Figure A-2. Identifying Relationship

In a non-identifying relationship, the foreign keys migrate to the data area. Since some (or all) of the migrated keys are not part of the primary key of the child, the child is not identified by the parent. An example of a non-identifying relationship is shown in figure A-3. Non-identifying relationships are indicated by a dashed line. The diamond indicates that the relationship is optional, i.e., it may not exist.

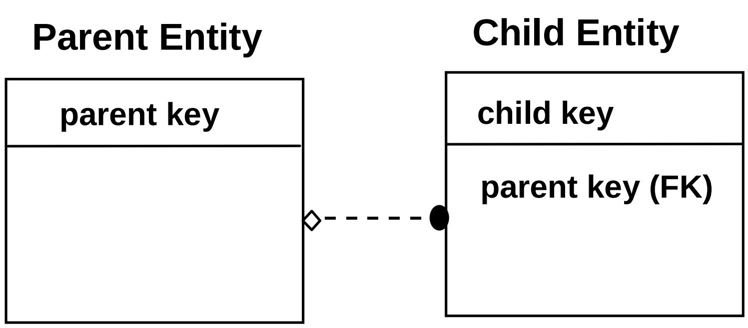


Figure A-3. Non-Identifying Relationship

Entities are designated as either independent (square boxes) entities, or dependent entities (round edges) as a function of how they acquire their keys. An independent entity does not depend on any other entity in a model for its identification, whereas a dependent entity depends on one or more entities in a model for both its existence and identification. An example is figure A-4, which illustrates both an independent and dependent entity.

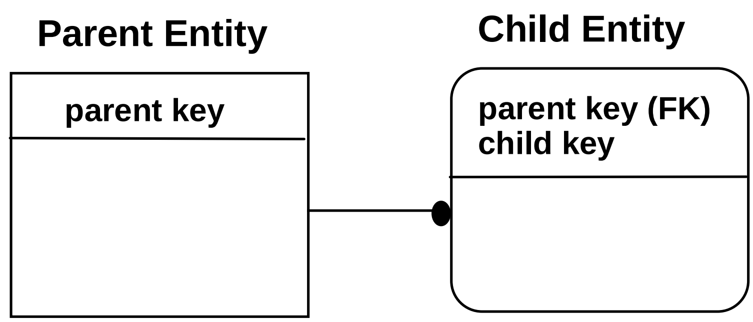


Figure A-4. Independent and Dependent Entities

There are two types of dependent entities. An existence dependent entity is one which cannot exist unless the parent does. An identification dependent entity cannot be identified without using the primary key of the parent. Identifying relationships always result in both existence and identification dependency. This type of relationship is prevalent in the flight data model. The airport entity and fuel type entity illustrate this concept in figure A-5 below.

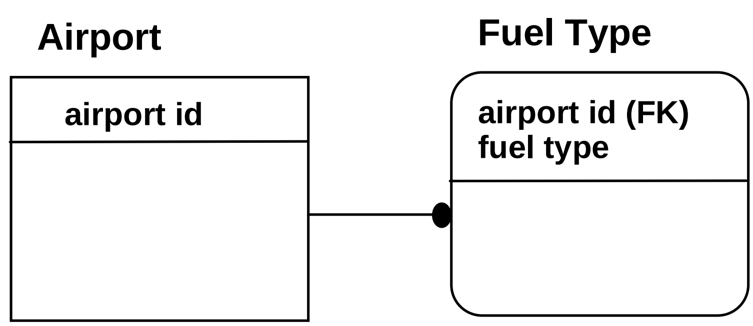
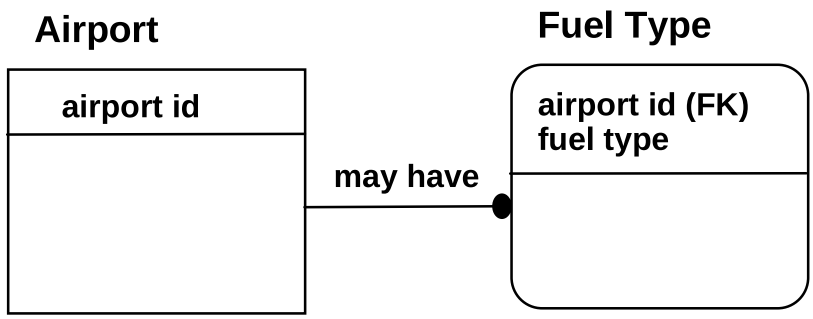
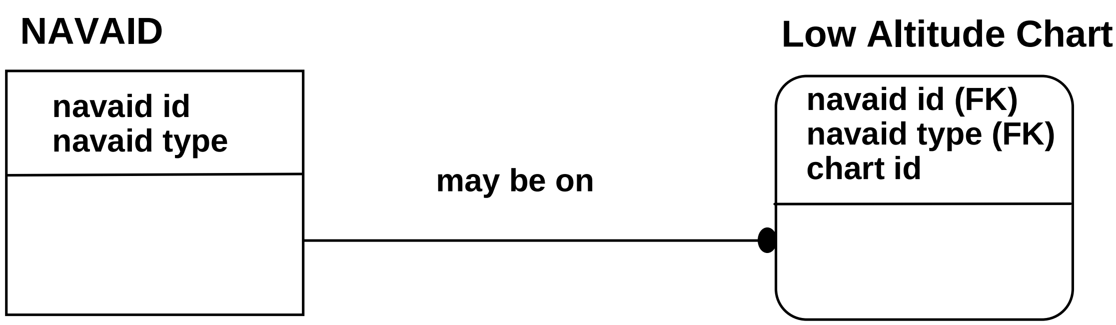


Figure A-5. Existence and Identification Dependency

Relationships are much more complex than they seem on the surface. They are the heart of the information model because they carry a lot of information. They describe the business rules and the referential integrity constraints. Relationships are displayed as a line connecting two entities, with a dot on one end and a verb phrase written along the line. If the verb phrase has been chosen correctly, one should be able to read a relationship from the parent to the child entity using the verb phrase. Figure A-6 illustrates two examples with verb phrases and describes how they are read.



An Airport <may have> many fuel types.



A NAVAID <may be on> many low altitude charts.

Figure A-6. Verb Phrases

Relationships have a property called cardinality which defines how many of each participating entity may, or must, participate. Cardinality statements are used to indicate how many instances of the parent entity are connected to how many instances of the child entity. For the identifying relationships below, the parent of a relationship can be related to the child in one of four ways:

1. Each parent is connected to Zero, One, or More instances of the child.

2. Each parent is connected to One or More instances of the child.

3. Each parent is connected to Zero or One instance of the child.

4. Each parent is connected to Exactly n instances of the child where the value n is known.

The cardinality of the relationship is depicted graphically in the ER diagram by a symbol that is located near the end of the relationship line. This is illustrated in figure A-7.

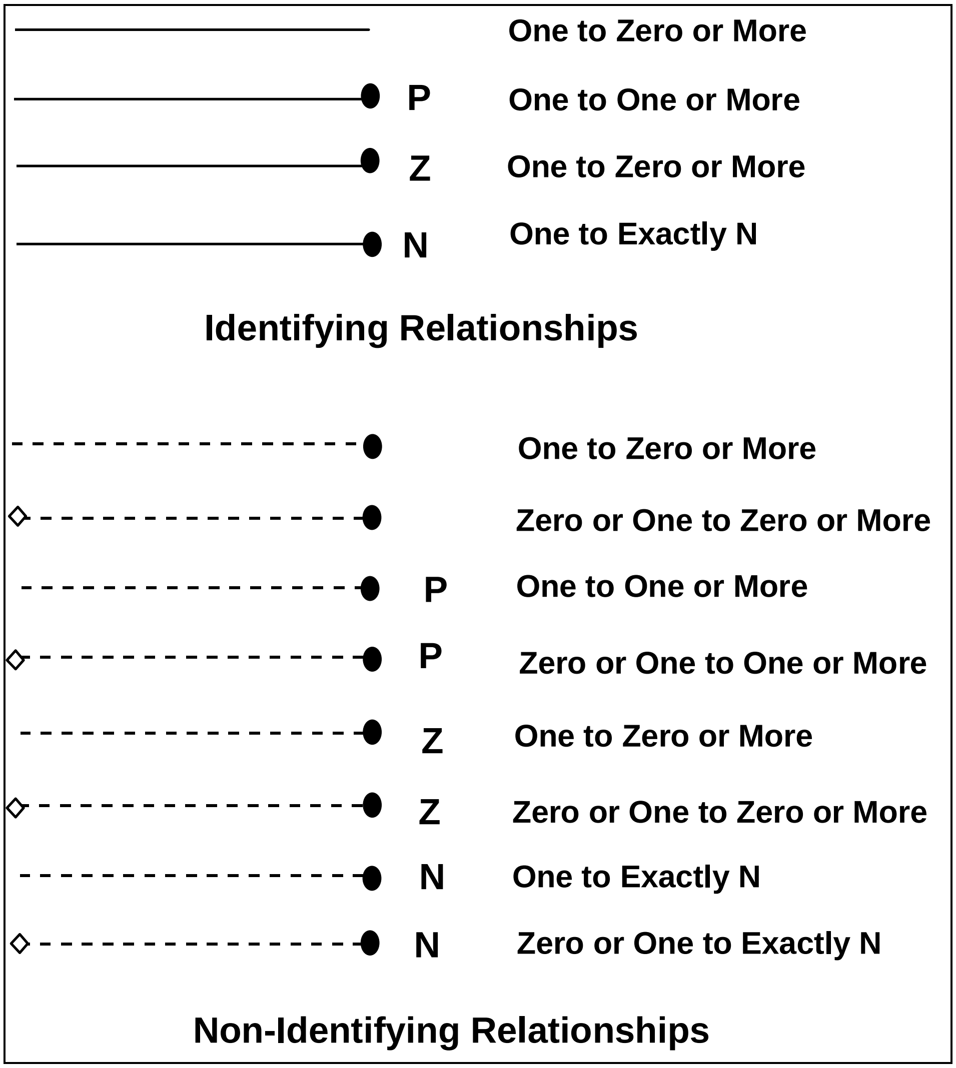


FIGURE A-7. Cardinality Statements

Another type of special relationship which exists in the model is a recursive relationship. A recursive relationship is one in which the parent entity and child entity for the relationship are the same. This type of relationship occurs frequently in the real world. Two examples are: companies own companies and employees manage employees. All recursive relationships are non-identifying. The flight data model has one such relationships in the event entity. For each event, there may exists the following recursive relationship: for any flight event, there may be another event related to it. This is illustrated below in Figure A-8.

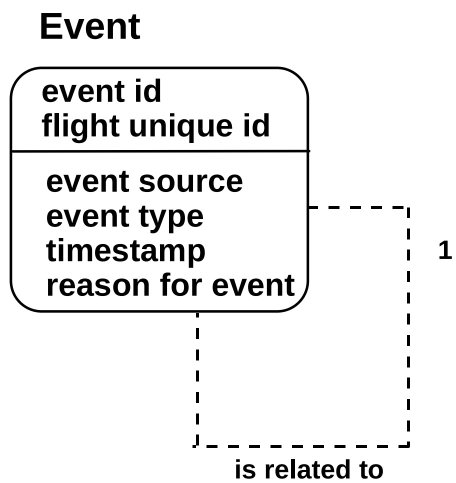


Figure A-8. Event Recursive Relationship

It is often the case that several attributes in a logical data model are defined over the same domain. They have the same allowed values, but not the same definitions. This is generally the case when foreign keys are contributed to an entity by a relationship. In these instances, role names are used to allow different occurrences of the same underlying attribute to play different roles. A role name is a new name that is assigned to a foreign key attribute and defines the role that it plays in the entity.

The syntax for defining and displaying a role name is (role-name.base-name). The part before the period is the role name. The remaining part contains the original name of the foreign key and is referred to as the base name. The flight data model several such relationships where role names are used. The airport id (base name) is a foreign key in which a role name is used to provide the necessary context. Figure A-9 illustrates the airport id attribute and the role it plays in the Flight ID entity to distinguish the departure airport (role name).

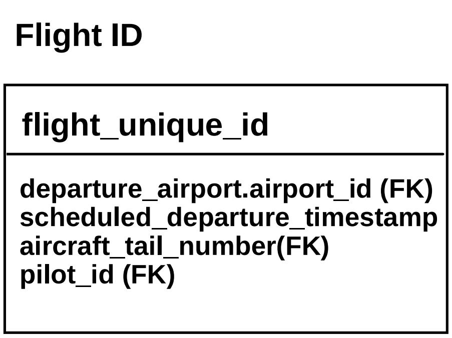


Figure A-9. Airport Id Role Name

Another type of entity that appears in the model is known as a generalization or inheritance hierarchy and may also be referred to as a sub-type or category hierarchy. A generalization hierarchy is a grouping of entities that share a set of common characteristics and consists of a generalization entity (generic parent), category entities and a category discriminator. A generalization entity is located at the top of any level of a generalization hierarchy. The category entity is a subset of the instances of a generalization entity that share common attributes or relationships that are distinct from other subsets. In addition, a category discriminator is an attribute that determines to which category a generic parent instance belongs.

The logical data model has several areas where the use of generalization hierarchies are used. These include Flight Plan Event, Dynamic Aircraft Event, and ATM Event. Within ATM Event are ATC Event and TFM Event. Figure A-10 illustrates how a generalization hierarchy was incorporated into the ATC Event family. Each ATC Event has its own characteristics, but it also shares a common set of characteristics. This common set of characteristics is maintained in the ATC Event entity (i.e., the generic parent) and the characteristics that are specific to each subset are maintained in the ATC Alert, Vectoring, Holding, and Change in Control entities, respectively.

Generalization hierarchies also contain the category relationship. This relationship connects the generic parent to the category entity. From the perspective of the parent, this is a one to zero or one relationship with the implicit verb phrase *is a*. Each instance of the generic parent either is an instance of the category or it is not. From the perspective of the category entity, each instance of the category entity is an instance of the generic parent. The category discriminator in this case is the atc event type attribute.

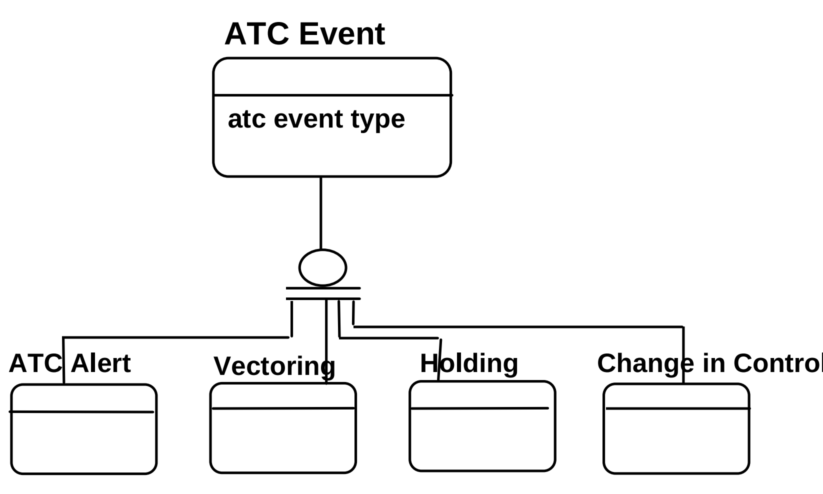


Figure A-10. ATC Event Generalization Hierarchy.

APPENDIX B: Relational Data Model Fundamentals

The following sections describe the fundamental features of the relational model.

Data Integrity. Ensuring data integrity is an essential aspect of data modeling, especially when data may be shared in a distributed processing system, a likely outcome for flight data with the NAS system architecture. Rules to maintain data integrity can be enforced at the database level (i.e., physical structure) and at the application level. The advantage of enforcing constraints at the database level are many, the primary one being that the constraints are automatically enforced in spite of a user's inadvertent attempts to modify the data incorrectly. In addition, the constraints are transparent to application programs. Additional constraints can be added on the data without affecting the application programs that process the data. Some disadvantages are that overhead is required to check constraints when inserting, updating, and deleting records, and additional indices must be maintained.

In the relational model, there are three types of data integrity rules. These are:

1. Domain integrity -- defines allowable values, or check conditions, for each data attribute

2. Entity integrity -- uniquely defines a relation and restricts the use of null values in a primary key

3. Referential integrity -- defines foreign key (attributes which are logically linked to the primary key of another table) relationships. During data maintenance, these relationships ensure that modifications to the data are properly implemented.

Rules 2 and 3 are general in that they apply to every database which conforms to the relational model. To implement domain integrity, a relational database must specify rules to enforce data accuracy and consistency.

The logical data model maximizes the number of data integrity constraints at the database level. These include check conditions[22] and validation tables[23] to enforce domain integrity constraints, primary keys to enforce entity integrity, and foreign keys to enforce referential integrity. Validation tables are used at the application level to validate data, ensuring data consistency of the domains of the attributes in the system.

There are numerous advantages in using validation tables to enforce domain integrity constraints at the application level. Application program maintenance is greatly reduced.

---

[22] A check condition is a domain integrity constraint that validates data in a table or view by requiring that a condition be true each time a record is inserted or updated.

[23] A validation entity stores the valid domain from which one or more attributes draw their values.

If the domain of an attribute changes, the modifications of the domain are applied to the validation table. Even though a validation table may be used in multiple applications, the modification of the domain is done in one place and is transparent to all applications. If validation rules are built into an application and modifications to the attribute become necessary, then each application affected must be modified.

Primary Keys. Each table in a relational database must have a primary key. A primary key is a single attribute or combination of attributes whose values uniquely identify each row in a table. That is, while the values of other attributes may be repeated in the table, primary keys must be unique. In addition, no attribute participating in the primary key of a table shall be allowed to accept null values. This rule is known as entity integrity.

Foreign Keys. Foreign keys are used to establish relationships between tables in the relational model. This can be viewed as establishing a parent/child relationship among tables. A foreign key is a single attribute or combination of attributes in one table whose values are required to match those of the primary key of some other table. In addition, the foreign key should be defined on the same logical domain as the corresponding primary key. Because foreign keys reference primary keys, it is essential that a primary key value exist somewhere in the database for every foreign key value. This rule is known as referential integrity. Any proposed design for the flight data model must designate foreign key relationships throughout the logical data model.

Enforcement of referential integrity is implemented as follows: When an insert or update is performed on a child table, the domain of values placed in the referencing attributes must match the domain of values in the parent table. If an effort is made to delete or update a row in the parent table which contains values matching those in the child table, one of three options will be applied.[24] The choice of update and delete constraints will be established by the developer.

Indices. An index is a data structure whose primary purpose is to speed retrieval of the records in a table. In addition, it may also be used to enforce certain uniqueness constraints on the data. While indices speed data retrieval, they also require storage. The tradeoff between space and retrieval speed may be important for large databases. Also, other system resources are needed to maintain indices. Therefore, there is also a tradeoff between delete, update and retrieval operations, which benefit from the use of indices, and the overhead needed to maintain an index.

---

[24] The cascade option will delete all dependent records from the child entity and then delete the parent entity.
The restrict option will prevent records from being deleted from the parent entity if dependent records exist.
The set null option will set null each nullable attribute in the foreign key of each dependent record and will delete the corresponding parent record.

APPENDIX C: Flight Data Model Detailed View

The content in this appendix is a detailed, attribute-level ERwin model view. It typically takes 4 sheets of paper. For final publication, the view will be professionally produced as a single oversized foldout. The printout is available from flight35.er1 by selecting the subject view 'Flight Object Model' and printing.

APPENDIX D: Flight Data Model Entity Definitions

The table below contains the definitions of the entities in the flight data model, as taken from the ERwin model.

Table D-1. Flight Data Model Entity Definitions

Entity Name	Entity Definition
Aircraft	Characteristics of the physical aircraft associated with a flight id
Aircraft_Alert	One of a number of alerts involving the aircraft in flight
Aircraft_Category	Coded value and additional information about the association of the aircraft with a standard category code based on aircraft characteristics
Aircraft_Comm_Equipment	Details about each item of communications equipment on an aircraft
Aircraft_Comm_Equipment_G	General information about the communications equipment on the aircraft
Aircraft_Dynamics_1	Dynamic (performance) information about the aircraft assigned to a flight
Aircraft_Dynamics_2	Additional dynamic aircraft status information
Aircraft_Emergency_Equipment	Details about each item of emergency equipment on the aircraft
Aircraft_Emergency_Equipment_G	General information about emergency equipment on the aircraft
Aircraft_NAV_Equipment	Details about each item of navigation equipment
Aircraft_NAV_Equipment_G	General information about navigation equipment on the aircraft
Aircraft_Type	Coded value based on the type of aircraft
Airport	Related file containing data about a departure or arrival airport; based on a unique airport id
Airport_Gate	Related information describing the airport gate assignment for a flight
Airport_Runway	Related information describing basic runway information associated with an airport
Arrival_Slot	The time window assigned to a flight at which it is intended to arrive at its destination airport
ATC_Alert	Data about various ATC alerts
ATC_Event	One of number of ATC events, such as ATC alerts, change in control, phase of flight information (pushback, wheels off), vectoring, taxiing, and holding
ATM_Event	Either an ATC or a TFM event
Change_in_Control	Data about changes in aircraft control, primarily Center-to-Center handoffs, including various track control actions (see NAS MD 311)
Dynamic_Aircraft_Event	A set of events that track changes in the status of the aircraft, such as position, fuel, and speed
Engine_Class	Engine class
Event	Base entity describing any activity involving the flight or the aircraft, as defined by the model
Event_Authority	The authorizing person or organization for an event
Event_Source	The source, whether person or organization, of an event
Fix	Related information about fixes along the route
Fix_Waypoint	A waypoint defined by a fix
Flight_As_Flown_Summary	Current information about the flight
Flight_As_Planned_Summary	Current flight plan information for a flight
Flight_Id	Base entity that identifies a unique flight
Flight_Plan_Event	Any flight plan event, such as a flight plan filing, amendment, or substitution
Flow_Program	Identification of TFM programs, such as ground delays or ground stops, to

	manage traffic
Flow_Restriction	Identification of TFM restrictions, such as metering and miles-in-trail, for the flight
FP_Amendment	Amendments to filed flight plans
FP_Filing	Information about filed flight plans
FP_Substitution	Information about substituted flight plans
Group_Assignment	Association of a number of flights by a single group designator
Holding	Data about the assignment of an airborne hold, including location and holding time
Itinerary	A full trip from initial departure to final destination, consisting of one or more flights, or hops, each consisting of a departure and arrival.
Itinerary_Hop	One 'leg' of a series of 'legs' or flights. Each hop is considered a flight.
Manufacturer	The manufacturer of an aircraft
NAVAID	Related information about the NAVAIDs along a route
NAVAID_Waypoint	A waypoint defined by a NAVAID
Onboard_Weather	Readings about weather in the area of the aircraft taken from the cockpit instruments
Operator	The company whose aircraft is flying on the route of flight
OPS_Center	The operational center of the (airline) operator
Other_Amendment	Amendments to a flight plan other than route amendments
Phase_of_Flight	An ATC event described on of several possible phases of flight
Pilot	Information about the pilot of the aircraft on a flight
Route	Related information about the route of flight, e.g., fixes and NAVAIDs along a series of flight segments
Route_Amendment	Flight plan amendments that modify the current route of flight
Route_Segment	
Route_Segment	A portion of a route bounded by a starting and an ending waypoint
Route_Structure	The definition of a route, consisting of a series of route segments
Segment_Ins_Del	Information about an insertion or deletion of a segment on a route structure
Segment_Named_Route	Segment as part of a named route
Segment_Paired_Waypoint	A route segment as defined by two waypoints
Segment_Waypoint	Information about one of the waypoints that defines a route segment
Taxiing	Data about taxiway time, taxiway name and gate information at the departure and arrival airports
TFM_Event	Include events generated by TFM which can be generated by various organizations: FAA, AOC, and the military
User_Defined_Waypoint	A user-defined waypoint specified by position information alone
Vectoring	The assignment of a new heading to a flight
Weight_Class	The weight class of a an aircraft

APPENDIX E: Flight Data Model Attribute Definitions

The table below contains the definitions of the attributes in the flight data model, as taken from the ERwin model. The organization of the table is alphabetically by entity and then alphabetically by attribute name within entity but with primary key attributes listed first.

Table E-1. Flight Data Model Attribute Definitions

Entity name	Attribute Name	Entity Definition	Key
Aircraft	aircraft_tail_number	Aircraft tail number	Yes
	aircraft_category_id	Aircraft category id	No
	aircraft_serial_number	Aircraft serial number	No
	aircraft_type_id	Id of the type of aircraft	No
	cruise_speed_max	Maximum airspeed of the aircraft, short of impacting structural integrity	No
	cruise_speed_min	Minimal airspeed to keep the aircraft aloft	No
	dry_weight	Aircraft weight in pounds when unloaded and unfueled	No
	endurance_speed_maximum	Maximum endurance speed	No
	engine_class_id	Engine class id	No
	engine_number	Engine serial number	No
	fuel_hours_on_board	Time in hours (and fraction of an hour) that the aircraft can fly based on amount of fuel on board at departure	No
	fuel_weight	Maximum capacity of fuel, in pounds	No
	landing_distance_minimum	Minimum landing distance, in feet	No
	manufacturer_id	Manufacturer id	No
	national_flag	Country of registration of the aircraft	No
	number_of_engines	Number of engines on the aircraft	No
	payload_weight	Weight in pounds of the aircraft fully loaded	No
	registration	Aircraft registration information	No
	seats	Number of passenger seats on the aircraft	No
	takeoff_distance_minimum	Minimum takeoff distance, in feet	No
	takeoff_weight_maximum	Maximum takeoff weight, in pounds	No
	weight_class_id	Weight class id	No
Aircraft_Alert	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	alert_description	Description of the alert situation	No
	alert_resolution	Manner in which the alert was resolved	No
	alert_status	Current status of the alert situation	No
	alert_type	Coded value for type of alert	No
Aircraft_Category	aircraft_category_id	Aircraft category id	Yes
	code	Code of the aircraft category	No
	description	Description of the aircraft category	No
Aircraft_Comm_Equipment	comm_equipment_id	Id of the aircraft's communications equipment	Yes
	code	Code of the aircraft's communications equipment	No

	description	Description of the aircraft's communications equipment	No
	name	Name of the aircraft's communications equipment	No
	serial_number	Serial number of the aircraft's communications equipment	No
Aircraft_Comm_Equipment_G	aircraft_tail_number	Aircraft tail number	Yes
	comm_equipment_id	Id of the aircraft's communications equipment	Yes
	channel	Broadcasting channel of the communications equipment	No
	frequency	Broadcasting frequency of the communications equipment	No
Aircraft_Dynamics_1	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	airspeed_calibrated	Calibrated airspeed	No
	airspeed_ground	Airspeed relative to the ground projected from the actual flight path velocity vector	No
	airspeed_indicator	Value shown by the airspeed indicator	No
	airspeed_true	Actual airspeed	No
	altitude	Current altitude: flying height in feet above sea level	No
	altitude_rate	Aircraft climb (+) or descent (-) rate	No
	fuel_burn_rate	Nominal fuel burn rate, in gallons per minute	No
	fuel_weight	Weight in pounds of fuel at this point in the flight	No
	latitude	Current latitude: distance north or south of the equator measured in degrees	No
	longitude	Current longitude: distance east or west of Greenwich measured in degrees	No
	pitch	Aircraft pitch	No
	roll	Aircraft roll	No
	sequence_number	System-generated unique number to identify this record at a point in time based on event timestamp	No
	takeoff_weight	Weight in pounds at takeoff	No
	thrust	Aircraft thrust	No
	yaw	Aircraft yaw	No
Aircraft_Dynamics_2	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	beacon_code	Beacon code assigned to the aircraft	No
	comm_status	Status of the communication system: on/off	No
	flight_mode	Mode of flight: auto-pilot, manual, ILS,....	No
	flight_phase	Phase of flight: possible states are: at the gate engine off, at the gate repairing, at the gate engine on, taxi to runway, waiting for traffic, at the runway, waiting for take-off, take-off, emergency take-off, aborting take-off, climbing, leaving the terminal, en route, cross sector boundary, in the holding pattern, transition to new route, entering terminal area, entering intercept point, entering approach gate, entering final approach fix, entering glide slope, on glide slope, touch down, taxi to gate, waiting to enter gate	No

	gear_status	Status of the landing gear systems: up, down, up left jammed, up right jammed, up front jammed, down left jammed, down right jammed, down front jammed	No
	mode_c_status	Status of the mode C altitude encoder, the component on the aircraft that transmits altitude information to the ground-based radar system (Nolan)	No
	priority_ac_status	Whether the aircraft has a priority status: y/n	No
	weight_on_wheels	Weight-on-wheels (WonW) flag: indicates whether the aircraft is on the ground or not.	No
Aircraft_Emergency_Equipment	emergency_equipment_id	Unique ID of the item of emergency equipment	Yes
	code	Equipment code	No
	manufacturer	Manufacturer of the item of emergency equipment	No
	name	Name of the item of emergency equipment	No
Aircraft_Emergency_Equipment_G	aircraft_tail_number	Aircraft tail number	Yes
	emergency_equipment_id	Unique ID of the item of emergency equipment	Yes
	description	Description of the specific item of emergency equipment	No
Aircraft_NAV_Equipment	nav_equipment_id	Id of the aircraft's navigation equipment	Yes
	code	Code of the NAVAID equipment	No
	description	Description of the NAVAID equipment	No
	name	Name of the NAVAID equipment	No
	serial_number	Serial number of the NAVAID equipment	No
Aircraft_NAV_Equipment_G	aircraft_tail_number	Aircraft tail number	Yes
	nav_equipment_id	Id of the aircraft's navigation equipment	Yes
	channel	Broadcasting channel of the navigation equipment	No
	frequency	Radio frequency of the navigation equipment	No
Aircraft_Type	aircraft_type_id	Id of the type of aircraft	Yes
	code	Code for the aircaft type	No
	description	Description of the aircraft	No
	model	Model of the aircraft	No
	name	Name of the type of aircraft	No
Airport	airport_id	Unique airport id	Yes
	elevation	Airport elevation in feet above mean sea level (msl)	No
	latitude	Airport latitude	No
	longitude	Airport longitude	No
	name	Official airport name	No
	other_element	Many other attributes exist and have already been modeled; (to be added when modeling NAS resources)	No
Airport_Gate	airport_id	Unique airport id	Yes
	gate_id	Unique gate id for the associated airport	Yes
	carrier	Name of the carrier operating the gate	No
	description	Description of the gate environment: location, characteristics, ...	No
Airport_Runway	airport_id	Unique airport id	Yes
	runway_id	Unique runway id for the associated airport	Yes
	altitude	Runway altitude	No

	code	Code for runway weight bearing capcity or pavement classification number (pcn)	No
	heading	Runway heading	No
	latitude	Runway latitude, based on the center line	No
	length	Runway length, in feet	No
	longitude	Runway longitude, based on the center line	No
Arrival_Slot	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	slot_duration	The length of time that the arrival time 'window' is open	No
	slot_start_time	The earliest time that the flight is due to arrive at its destination airport	No
ATC_Alert	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	atc_alert_description	Description of the alert event	No
	atc_alert_type	ATC alert type, e.g., TCAS alarm, conflict probe warning, crash, hijack, depressurization, diversion	No
ATC_Event	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	atc_event_type	Code for type of ATC event	No
	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	atm_event_type	Type of ATM event, consisting of ATC events and TFM events	No
Change_in_Control	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	cc_seq_num	Command Center-generated sequence number in the case of multiple border crossings within the same handoff	No
	change_control_status	Change in control status: one of a number of track control events: 'a' accept handoff, 'i' initiate handoff 's' select automatic handoff, 't' track (? track control only)	No
	description	Description of the change in control process	No
	fac_id_new	Facility id acquiring control of a flight	No
	fac_id_old	Facility id relinquishing control of a flight	No
	loc_id	Location where the change takes place, i.e., fix_id	No
Dynamic_Aircraft_Event	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	aircraft_event_type	Type of aircraft event, e.g., aircraft_data_1, aircraft_data_2, onboard_weather, or aircraft_alert	No
Engine_Class	engine_class_id	Engine class id	Yes
	code	Code for the engine class	No
	description	Description of the engine class	No
	name	Name of the engine class, if any	No
Event	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	event_authority_id	Unique id for the authority for an event	No
	event_source_id	Unique id for source of a flight event	No
	event_type	Type of event: flight plan, ATC, dynamic aircraft	No

	reason_for_event	Reason for the event	No
	related	System-generated unique flight id	No
	remark	Descriptive remarks	No
	timestamp_event	Date and time when an event occurred or is scheduled to occur	No
	timestamp_recorded	Date and time when an event is recorded in the database for audit and traceability purposes	No
Event_Authority	event_authority_id	Unique id for the authority for an event	Yes
	address	Address of the authority that has approved the event	No
	facility_id	Facility id of the event authority	No
	facility_name	Facility name of the event authority	No
	name	Name of the authority that has approved the event	No
	organization	Organization of the authority that has approved the event	No
	phone_number	Phone number of the authority that has approved the event	No
Event_Source	event_source_id	Unique id for source of a flight event	Yes
	address	Address of the originator that generated the event	No
	facility_id	Facility id of the originating event source	No
	facility_name	Facility name of the event source	No
	name	Name of the originator that generated the event	No
	organization	Organization that originated the event	No
	phone_number	Phone number of the originator that generated the event	No
Fix	fix_id	Unique fix id	Yes
	fix_name	Fix name	No
	fix_type	Fix type	No
Fix_Waypoint	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	route_segment_id	Id of the segment within the route structure	Yes
	route_structure_id	Id of the route structure	Yes
	segment_waypoint_id	Id of the segment waypoint	Yes
	fix_id	Unique fix id	No
Flight_As_Flown_Summary	flight_unique_id	System-generated unique flight id	Yes
	arrival_taxi_time_amt	Time, in minutes, of taxi time from the arrival runway to the arrival gate	No
	arrival_time_gate	Time when arrived at the arrival gate	No
	arrival_time_runway	Time of arrival at the destination airport runway	No
	departure_taxi_time_amt	Time, in minutes, of taxi time from the departure gate to the runway	No
	departure_time_gate	Time of departure from the gate	No
	departure_time_runway	Identity of the departure runway	No
	event_id	Unique system-generated event id	No
	flight_status	Status of the flight: at the departure gate engine off, at the departure gate engine on, taxi to departure runway, at the departure runway, takeoff, climbing, en route, descent, ...	No
	persons_on_board	Number of persons, including pilot(s), onboard	No
	time_enroute	Actual time en route	No

Flight_As_Planned_Summary	flight_unique_id	System-generated unique flight id	Yes
	event_id	Unique system-generated event id	No
	fp_alternate_dest_airport	Alternate destination airport	No
	fp_altitude_cruise	En route cruise altitude of the flight	No
	fp_arrival_airport	Scheduled arrival airport	No
	fp_arrival_country	Arrival country of the flight	No
	fp_arrival_gate	Scheduled arrival gate	No
	fp_arrival_runway	Scheduled arrival runway	No
	fp_arrival_time_gate	Scheduled gate arrival time	No
	fp_arrival_time_runway	Scheduled wheels-on time at arrival runway	No
	fp_departure_airport	Departure airport	No
	fp_departure_country	Departure country where the flight originated	No
	fp_departure_gate	Departure gate	No
	fp_departure_runway	Scheduled departure runway	No
	fp_departure_time_gate	Scheduled time of departure from the gate (pushback)	No
	fp_departure_time_runway	Scheduled wheels-off time	No
	fp_endurance	Estimated flying time for the aircraft within refueling	No
	fp_enroute_time	Estimated duration of the flight	No
	fp_flight_plan_status	Status of the flight plan: inactive, active, clear, cancel, closed	No
	fp_flight_route	Route of the flight (may have more than one or more segments)	No
	fp_speed_cruise	Enroute cruise speed of the flight	No
	fp_turn_around_time	Estimated time interval between the arrival time for this flight and the departure time for the next flight for the same aircraft	No
Flight_Id	flight_unique_id	System-generated unique flight id	Yes
	aircraft_tail_number	Aircraft tail number	No
	departure_airport	Unique airport id	No
	pilot_id	Unique pilot id	No
	scheduled_departure_timestamp	Time departure scheduled	No
Flight_Plan_Event	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	fp_event_type	Type of flight plan: e.g., i=initial filing/proposed, a=amended, c=cancelled, t=activated, c=accepted, r=rejected	No
Flow_Program	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	authority	Person or organization authorizing the program	No
	estimated_departure_clearance_time	EDCT: time to be given by the ATCSCC that a flight is scheduled to depart, based on a ground delay program in effect	No
	estimated_duration	Estimated time (in minutes) that the program is estimated to remain in effect	No
	location	Site where the program will occur; typically an airport_id or other location id	No
	remark	Contextual remark about why the program was needed, its outcome, or special characteristics about the program	No

	start_time	Proposed time that the program is to start	No
	type_program	Type of program: ground delay or ground stop	No
Flow_Restriction	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	estimated_duration	Time (in minutes) for the duration of the restriction, if any	No
	location_id	Location at or near which the program is based	No
	meter_fix_id	Id of the fix on which the meter fix program is based	No
	miles_in_trail	Distance assigned for miles-in-trail (MIT) program	No
	remark	Contextual remark about why the restriction was needed, its outcome, or special characteristics about the restriction	No
	start_time	Time at which the restriction will take effect	No
	time_at_fix	Time assigned to the flight when it is due at the meter fix	No
	type_restriction	Type of TFM restriction: milies-in-trail, metering	No
FP_Amendment	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	amend_status	Status of an amendment to a flight plan: e.g., inactive, active, and cancel.	No
	amend_type	Type of amendment; from the Host, actions are: 'a' assigned altitude action, 'c' code modification action, 'q' qualifier modification action, 's' coast track action, 't' track action	No
	filing	System-generated unique flight id	No
FP_Filing	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	airspeed_cruise	Cruise speed of the flight plan	No
	airspeed_cruise_alternate	Alternate cruise speed of the flight plan	No
	altitude_cruise	Planned altitude at cruise speed (in feet above sea level)	No
	altitude_cruise_alternate	Alternate planned altitude at cruise speed (in feet above sea level)	No
	arrival_airport_id	Unique airport id	No
	arrival_time_gate	Estimated arrival time at destination gate	No
	departure_airport_id	Unique airport id	No
	departure_time_gate	Filed time of departure from the gate (planned pushback); equivalent to the EDCT (Expected Departure Clearance Time)	No
	destination_country_id	Code of country of destination of the flight; default to 'us'	No
	enroute_time	Planned time en route	No
	enroute_time_alternate	Alternate planned time en route	No
	flight_type	Type of flight as shown in flight plan: v: civilian vfr, mv: military vfr, i: civilian ifr, mi: military ifr, o: other, g: ga	No
	flightplan_status	Status of a flight plan: e.g., inactive, active, approved, cleared, cancelled, closed	No
	flightplan_type	Type of flight plan: e.g., trial, scheduled, planned, ICAO, preferred ATC, preferred AOC	No
	origin_country_id	Code of country of origin of the flight; default to 'us'	No

FP_Substitution	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	new_flight_id	Unique flight id of the of the flight which will substitute for the current flight	No
	reason_for_substitution	Reason why a new flight plan was substituted	No
Group_Assignment	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	group_id	Identifier of the group of flights	No
	group_type	Type of the assigned group: bank, managed arrival reservoir (MAR), severe weather (SWAP), military	No
Holding	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	estimated_holding_time	Estimated amount of time to be in the holding pattern	No
	holding_altitude	Holding altitude	No
	holding_pattern	Assigned pattern for the holding: e.g., circle, race track, other	No
	holding_waypoint_id	Id of the holding waypoint	No
	holding_waypoint_type	Type of waypoint at which holding is occurring, e.g., fix, NAVAID, ils	No
Itinerary	itinerary_id	Unique system-generated itinerary identifier	Yes
	center_id	Unique identifier of an Air Route Traffic Control Center (ARTCC), i.e., CID	No
	itinerary_type	Type of itinerary: 'OAG', 'military', 'GA', 'air taxi', 'intl', 'other'	No
	number_of_hops	Number of hops, or flights, in the itinerary	No
	operator_id	Unique operator id, e.g., UAL	No
Itinerary_Hop	hop_sequence_number	The position in the itinerary that the flight represents, e.g., from 1 to n	Yes
	itinerary_id	Unique system-generated itinerary identifier	Yes
	flight_unique_id	System-generated unique flight id	No
Manufacturer	manufacturer_id	Unique manufacturer id	Yes
	code	Manufacturer code	No
	country_id	Unique country id of manufacturer	No
	description	Description of the manufacturer	No
	name	Manufacturer name	No
NAVAID	navaid_id	NAVAID id	Yes
	navaid_type	NAVAID type	Yes
	navaid_name	NAVAID name	No
NAVAID_Waypoint	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	route_segment_id	Id of the segment within the route structure	Yes
	route_structure_id	Id of the route structure	Yes
	segment_waypoint_id	Id of the segment waypoint	Yes
	navaid_id	NAVAID id	No
	navaid_type	NAVAID type	No
Onboard_Weather	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	air_temperature	Temperature of the surrounding air	No
	barometric_pressure	Barometric pressure at the aircraft altitude	No
	wind_direction	Direction of the winds aloft	No

	wind_speed	Speed of the winds aloft	No
Operator	center_id	Unique identifier of an Air Route Traffic Control Center (ARTCC), i.e., cid	Yes
	operator_id	Unique operator id, e.g., UAL	Yes
	email_address	Email address of the operator	No
	mailing_address	Operator mailing address	No
	name	Operator name	No
	phone_number	Operator phone number	No
	type	Operator type: domestic scheduled air carrier, air freight, military, international, general aviation	No
OPS_Center	center_id	ID of the Center	Yes
	center_name	Name of the Center	No
	center_type	Type of Center	No
Other_Amendment	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	reason_for_change	Reason for the amendment	No
Phase_of_Flight	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	flight_phase	Type of phase of flight: push-back, wheels off, wheels on, gate arrival	No
	source	Source of the timestamp for this phase of flight	No
Pilot	pilot_id	Unique pilot id	Yes
	address	Pilot address	No
	air_carrier	Air carrier for whom pilot flies, if applicable	No
	name	Pilot name	No
	phone	Pilot phone	No
Route	route_id	Unique route id	Yes
	route_name	Route name	No
	route_type	Route type	No
Route_Amendment	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	reason_for change	Reason for the flight plan amendment to change the route	No
	route_structure	Id of the route structure	No
Route_Segment	event_id	Unique system-generated event id	Yes
	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	route_segment_id	Id of the segment within the route structure	Yes
	route_structure_id	Id of the route structure	Yes
	route_structure_id	Id of the route structure	Yes
	altitude_rate	Assigned altitude rate for the segment	No
	est_time_seg	Estimated time for the flight to traverse the segment	No
	mea	Minimum en route altitude	No
	moca	Minimum obstruction clearance altitude	No
	segment_remark	Remark about the segment	No
	sequence_number	Sequence number of a route segment which represents the order of the segment within a complete route between the departure and arrival airports	No

	speed	Assigned speed for the segment	No
Route_Structure	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	route_structure_id	Id of the route structure	Yes
	number_of_segments	Number of segments on the route	No
	route_remark	Remark about the route	No
	route_status	Status of the route: filed, active, cancel, ...	No
	route_type	Type of route: static flight trajectory, sampled flight trajectory, optimal flight trajectory, NPR trajectory, International flight trajectory, oceanic flight trajectory	No
Segment_Ins_Del	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	action_type	Type of change to the segment structure: m: modified, i: inserted, d: deleted	No
	segment_sequence_number	System-generated ordinal associated with the segment	No
Segment_Named_Route	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	route_segment_id	Id of the segment within the route structure	Yes
	route_structure_id	Id of the route structure	Yes
	segment_named_route_id	Name of the segment on the route	Yes
	route_id	Unique route id	No
Segment_Paired_Waypoint	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	route_segment_id	Id of the segment within the route structure	Yes
	route_structure_id	Id of the route structure	Yes
Segment_Waypoint	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	route_segment_id	Id of the segment within the route structure	Yes
	route_structure_id	Id of the route structure	Yes
	segment_waypoint_id	Id of the segment waypoint	Yes
	waypoint_altitude	Altitude of the waypoint	No
	waypoint_eta	Estimated arrival time at the waypoint	No
	waypoint_starting_flag	Flag to define if the waypoint is at the beginning of the segment	No
	waypoint_type	Type of waypoint: normal, metering fix, coordinate fix, arrival fix, departure fix	No
Taxiing	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	gate_id_arrival	Id of the arrival gate	No
	gate_id_departure	Id of the departure gate	No
	queue_time_arrival	Amount of time in queue waiting for a slot at the gate	No
	queue_time_departure	Amount of time in takeoff queue	No
	taxiway_id_arrival	Id of arrival taxiway (assumes only one allowed)	No
	taxiway_id_departure	Id of departure taxiway (assumes only one allowed)	No
	taxiway_time_arrival	Minutes of time on the taxiway on arrival	No
	taxiway_time_departure	Minutes of time on the taxiway on departure	No
TFM_Event	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes

	tfm_event_type	Code for the type of TFM event	No
User_Defined_Waypoint	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	route_segment_id	Id of the segment within the route structure	Yes
	route_structure_id	Id of the route structure	Yes
	segment_waypoint_id	Id of the segment waypoint	Yes
	user_defined_waypoint_id	Id of a user-defined waypoint	Yes
	latitude	Latitude of the user-defined waypoint	No
	longitude	Longitude of the user-defined waypoint	No
Vectoring	event_id	Unique system-generated event id	Yes
	flight_unique_id	System-generated unique flight id	Yes
	heading	New heading/direction given by ATC	No
Weight_Class	weight_class_id	Weight class id	Yes
	code	Weight class code, if it exists	No
	description	Description of the weight class	No
	weight_clean	Unloaded weight of the aircraft	No

LIST OF REFERENCES

[1] RTCA Special Committee 169, "Operational Concepts and Data Elements required to Improve Air Traffic Management (ATM) - Aeronautical Operational Control (AOC) Ground - Ground Information Exchange to Facilitate Collaborative Decision Making," DO-241, 6 October 1977, internet address: www.rtca.org

[2] RTCA, "Government/Industry Operational Concept for NAS Modernization, 1998-2002: Free Flight Phase 1, Volume I," June, 1998.

[3] NAS-MD-311: "Message Entry and Checking," NAS En Route Configuration Management Document, Computer Program Functional Specifications, Model A4e1.3, 15 August 1995

[4] Solomos, G., et al., "Air Traffic Management Modeling Capabilities: A Detailed Template for Looking at Models Available to CAASD," Draft Working Note, July 1997, MITRE/CAASD

[5] "Air Traffic Service (ATS) Concept of Operations," Federal Aviation Administration, 1997

[6] "Free Flight Phase 1 Operational Concept," The MITRE Corporation, McLean, VA, 1998

GLOSSARY OF ACRONYMS

AIS	Aeronautical Information System
ACARS	Aircraft Communications Addressing and Reporting System
AOC	Aeronautical Operational Control
ATC	Air Traffic Control
ATM	Air Traffic Management
ARTS	Airspace Radar Terminal System [??]
ASD	System Development Organization
ATS	Air Traffic Service
AZ	(ETMS) Arrival Message
CDM	Collaborative Decision Making
CNS	Communications, Navigation, Surveillance
COTS	Commercial Off the Shelf
CTA	Control Time of Arrival
CTAS	Center TRACON Automation System
DBMS	Database Management System
DOTS	Dynamic Ocean Track System
DZ	(ETMS) Departure Message
ER	Entity Relationship
ERATMDST	En Route Air Traffic Management Decision Support Toolset
ETE	Estimated Time En Route
ETMS	Enhanced Traffic Management System
FAA	Federal Aviation Administration
FAST	Final Approach Spacing Tool
FFF	Flight Data Model
FDP	Flight Data Processing
FIR	Flight Information Region
FSM	Flight Schedule Monitor
GA	General Aviation
GPS	Global Positioning System
HADDS	Host Application Data Distribution System
HCS	Host Computer System
ICAO	International Civil Aviation Organization
IDEF1X	Integrated Computer Aided Manufacturing Definition 1X
IFR	Instrument Flight Rules
IPT	Integrated Product Team

LDM	Logical Data Model
LOC-ID	Location Identifier
MNPS	Minimum Navigation Performance Specification
NAS	National Airspace System
NASR	NAS Resources
NAVAID	Navigational Aid
NFDC	National Flight Data Center
NIAC	NAS Information Architecture Committee
OAG	Official Airline Guide
ODMS	Operational Data Management System
PDM	Physical Data Model
pFAST	Passive Final Approach Spacing Tool
RBS	Ration By Schedule
R&D	Research and Development
RTCA	Radio Telecommunications Communications Association
SAR	System Analysis and Recording
SC	Special Committee
TADDS	Terminal Application Data Distribution System
TFM	Traffic Flow Management
TMA	Traffic Management Advisor
TMC	Traffic Management Coordinator
TMU	Traffic Management Unit
TZ	(ETMS) Track Message
URET	User Request Evaluation Tool
UTC	Universal Time Coordinate
VFR	Visual Flight Rules

[1] Working with the NIAC, these organizations include: (in FAA) AOP, AIT, AUA, ACT, AND, ANS, AAF; (other) CAASD, SETA, Lincoln Labs, Unitech, AUA/TAC, DMR, SEI

[1] There have various proposed NAS categorizations over time. The one proposed here stems from work done in the TFM-ART from 1992-3.

[1] Platinum Technology acquired LogicWorks, the developer of the ERwin product, in the spring of 1998.

[1] The benefit of using a standards-based product is that a model based on the IDEF1 standard built with ERwin can be imported by most any other IDEF1-based data modeling tool, such as Oracle's Designer2000 and Sybase's S-Designor. With new interfaces now being built, it can also be imported into Unified Modeling Language (UML) products, such as Rose, from the Rational Corporation.

[1] DO-241, "Operational Concepts and Data Elements required to Improve Air Traffic Management (ATM) - Aeronautical Operational Control (AOC) Ground - Ground Information Exchange to Facilitate Collaborative Decision Making," 6 October 1977, RTCA, Special Committee 169; quotation taken from the recommendations, page ES-4.

[1] *Government/Industry Operational Concept for NAS Modernization, 1998-2002: Free Flight Phase 1, Volume I*, RTCA, June, 1998.

[1] FFP1, vol 1

[1] Op. cit.

[1] The detailed categorization is found as Table A-1 in Appendix A.

[1] ATS Concept of Operations **[give complete reference]**

[1] This initial flight data model does not necessarily use the exact names of the legacy flight data elements or of the more standardized flight data attributes to which the legacy attributes were mapped. That process is ongoing and is part of completing a data naming standard.

[1] Based on a discussion in RTCA, DO-241 cited above.

[1] The term 'domain,' when used in a data modeling context, refers to the set of allowable values that an attribute may take.

[1] For example, ……

[1] A flight itinerary is a concatenated set of flights in which a flight consists of one takeoff followed by a landing

[1] In data model, a child entity is one that is dependent on another, i.e., parent, entity.

[1] A test was done to import the relational model from LogicWorks' ERwin product to Rational's Rose object-oriented modeling product using new a software translator recently developed by the two companies. The import was successful in that the basic structures were created in Rose but some effort was still needed to adjust the view that Rose created. Additional effort is needed to sort out the pros and cons on this translation process.