Title: An Object-Oriented Analysis of Air Traffic Control

Author(s): Celesta G. Ball and Rebecca L. Kim

Dept.: F048  Project No.: 1763L
Date: August 1991  Contract No.: DTFA01-89-C-00001
Issued at: Washington  Sponsor: Federal Aviation Administration

ABSTRACT:

An object-oriented analysis (OOA) of air traffic control was performed to support Integration and Interaction Laboratory (I-Lab) experimental studies concerning the integration of National Airspace System (NAS) subsystems, procedures and airspace. This document describes a domain analysis of the air traffic control (ATC) system, using graphical images of classes, objects, structures, and views, and class and object specifications. An experiment and the requisite legacy (existing) software are described in terms of the ATC domain model, leading to a comparison of needed experiment capabilities with available legacy software capabilities. It is assumed that the readers of this document have some familiarity with ATC.

Suggested Keywords: air traffic control (ATC); object-oriented analysis (OOA); experimentation; legacy software
ACKNOWLEDGMENTS

Many individuals contributed to this report, by offering advice in refining concepts, by sharing ATC expertise, and by reviewing and commenting on draft versions of particular sections.

The authors appreciate the lively discussions with Anne C. Deslattes (an original member of the OOA Phase 2 team) and have benefitted from her experiences in systems analysis and design.

Many graciously responded to our requests for ATC and legacy software knowledge, including Richard A. Rucker (on ATC in general and on many detailed topics); Anand D. Mundra (on the terminal airspace environment); Robert M. Tarakan and Shaula E. Doyle (on TASF); John A. Summers, Niamh M. Burke, and Barbara C. Zimmerman (on the AERA 2 CHI Prototype); and many others. Duane W. Small was the source of information for the original (December 1990) definition of the experiment.

We gratefully acknowledge those who took time to comment on draft versions and to work with us on revisions, including Patsy Ryberg (on the methodology); Duane W. Small (on the methodology, the ATC model, and the experiment in particular, and on the entire document in general); Shaula E. Doyle (on TASF); Norma J. Taber (on the methodology and the ATC model); and Pamela A. Dargan (on the methodology).

Most especially, we thank Richard K. Sciambi, our peer reviewer, for his insightful and timely comments. Reviewing a large document which applies new techniques to a complex and evolving subject area was surely tedious and time-consuming, and we appreciate his diligence.
Table Of Contents

INTRODUCTION

• 1.1 BACKGROUND
• 1.2 OBJECTIVES AND SCOPE
• 1.3 DOCUMENT ORGANIZATION

METHODOLOGY

• 2.1 THE OBJECT-ORIENTED ANALYSIS TASK
• 2.2 MODELING CONCEPTS
  • 2.2.1 Classes and Objects
  • 2.2.2 Instance Connections and Message Connections
  • 2.2.3 Structures
  • 2.2.4 Subjects
  • 2.2.5 Views
  • 2.2.6 Model Specification
• 2.3 MODEL USAGE
  • 2.3.1 Reference Model
  • 2.3.2 Evaluation of Experiment vs. Legacy Software
  • 2.3.3 Extensibility and Reusability
• 2.4 IMPLICATIONS FOR OBJECT-ORIENTED DESIGN

DOMAIN ANALYSIS

• 3.1 ATC
  • 3.1.1 Scope of the ATC Model
  • 3.1.2 Evolution of ATC from Air Traffic
• 3.2 THE ATC MODEL
  • 3.2.1 The User Subject
    • 3.2.1.1 The Aircraft Class
    • 3.2.1.2 The Ground Vehicle Class
    • 3.2.1.3 The Flight and Vehicle Classes
    • 3.2.1.4 The Clearance Class
    • 3.2.1.5 The Air Traffic, Ground Traffic, and Traffic Classes
  • 3.2.2 The Resource Subject
    • 3.2.2.1 The Airspace/Ground Resource Class
    • 3.2.2.2 The Surveillance Resource Class
    • 3.2.2.3 The Navigation Resource Class
    • 3.2.2.4 The Communications Resource Class
    • 3.2.2.5 The Aviation Weather Resource Class
  • 3.2.3 The Manager Subject
  • 3.2.4 INSTANCE CONNECTIONS
    • 3.2.4.1 User-Manager Instance Connections
    • 3.2.4.2 Manager-Resource Instance Connections
    • 3.2.4.3 User-Resource Instance Connections
    • 3.2.4.4 User-User Instance Connections
  • 3.2.5 Message Connections
  • 3.2.6 Views
    • 3.2.6.1 Clearance to Enter Airport Movement Area Event
    • 3.2.6.2 Handoff to Flight Manager (Flight State Change) Event
    • 3.2.6.3 Problem Detection and Resolution Events
    • 3.2.6.4 Airspace/Ground Resource Saturation Detection and Resolution Event
    • 3.2.6.5 Clearance Delivery (Communications) Event
    • 3.2.6.6 NAVAID Use (Navigation) Event
    • 3.2.6.7 Tracking (Surveillance) Event
    • 3.2.6.8 Weather Events
  • 3.2.7 Model Specification

© 2000 The MITRE Corporation. ALL RIGHTS RESERVED.
PROBLEM ANALYSIS

4.1 THE EXPERIMENT
   • 4.1.1 Analysis of the Experiment vs. the ATC Model
   • 4.1.2 The Experiment Model
      ■ 4.1.2.1 The User Subject
      ■ 4.1.2.2 The Resource Subject
      ■ 4.1.2.3 The Manager Subject
      ■ 4.1.2.4 Event View: Handoff of a Flight from En Route Controller to Terminal Controller

4.2 MAPPING THE LEGACY SOFTWARE ONTO THE ATC MODEL
   • 4.2.1 TASF
      ■ 4.2.1.1 Analysis of TASF vs. the ATC Model
      ■ 4.2.1.2 The TASF Model
   • 4.2.2 AERA 2
      ■ 4.2.2.1 Analysis of AERA 2 vs. the ATC Model
      ■ 4.2.2.2 The AERA 2 Model

4.3 AN ASSESSMENT OF NEEDED CAPABILITIES

Appendix A - ATC MODEL SPECIFICATIONS
Appendix B - EXPERIMENT MODEL SPECIFICATIONS
Appendix C - TASF MODEL SPECIFICATIONS
Appendix D - AERA 2 MODEL SPECIFICATIONS

LIST OF REFERENCES
BIBLIOGRAPHY
GLOSSARY
SECTION 1

INTRODUCTION

1.1 BACKGROUND

The Integration and Interaction Laboratory (I-Lab) project is a development program undertaken by the Center for Advanced Aviation System Development of The MITRE Corporation and sponsored by the Federal Aviation Administration (FAA) Operations Research Service, AOR-1. A first step towards development of an Air Traffic Control (ATC) National Simulation Laboratory (NSL), the I-Lab project has been established by the FAA as a means of incrementally developing the operational and architectural concepts and the requirements for the NSL. The software and architecture of the I-Lab testbed will be used as the basis for establishment of the initial baseline system at the NSL.

As part of the development process for the I-Lab architecture, an object-oriented analysis of the ATC domain was carried out, producing a model of the ATC system. In order to demonstrate the practicality of the ATC model, an experiment (i.e., an integration study involving individual National Airspace System [NAS] subsystems, procedures, and airspace) and two pieces of legacy (existing) software were defined in terms of the ATC model. The experiment model identified needed capabilities and the legacy software models identified existing capabilities. Comparisons of the experiment model with the legacy software models revealed software capabilities still not available.

This document describes the method of analysis, the models constructed, and an example of the practical application of the models.

1.2 OBJECTIVES AND SCOPE

The objectives of the analysis included the following:

- To provide a reference model for software definition - The analysis produced a reference model for defining I-Lab software. It was verified that legacy software can be defined in terms of the ATC model. Object-oriented designs of new I-Lab software can be based upon the ATC model. The ATC model also provides a reference model for the definition of non-I-Lab legacy software or new software which must be integrated with I-Lab software.

- To provide a reference model for experiment definition - The object-oriented model of ATC also provides a reference model for defining experiments. It was verified that an experiment can be defined in terms of the ATC model

- To identify I-Lab functionality needed for the initial application - The initial application is defined as a set of experiments. By comparing an experiment model with legacy software models, new functionality (i.e., not provided by legacy software) needed to conduct the
experiment can be identified. Existing software which might be used can also be identified for further evaluation.

- To provide a model for the NSL - An object-oriented analysis performed for the I-Lab can serve as a model for the NSL. No existing object-oriented analysis methodology by itself was sufficient for the task; the analysis resulted in a more useful methodology.

An analysis of ATC can also identify potential areas for automation, beyond those documented by the National Airspace System (NAS) Plan or the Federal Aviation Administration Plan for Research, Engineering and Development (the "RE&D Plan"). This outcome was considered a side benefit, rather than a goal, of the task.

The steps that were followed in meeting the analysis objectives - the test of the methodology - are described in more detail in section 2.1.

The broad scope of study was ATC in the Automated En Route Air Traffic Control Phase 2 (AERA 2) timeframe, i.e., the year 1999 and approximately 10 years beyond. The scope was limited by the following:

- The amount of resources (time and staffing) available - For example, the depth of study in any one area was limited.

- A (subjective) consideration of I-Lab objectives - For example, horizontal (across application) details were emphasized over vertical (within application) details, and applications which were unlikely to be the subject of experiments within I-Lab were omitted.

1.3 DOCUMENT ORGANIZATION

Section 2 of this document describes the methodology developed for the object-oriented analysis. The documentation of the ATC domain model is contained in section 3, which concentrates on the ATC classes, objects, structures and views, and in appendix A, which contains specifications for the classes and objects. Section 4 provides models for an experiment and for two pieces of legacy software, and identifies needed capabilities to support the experiment. As in section 3, section 4 concentrates on classes, objects, structures and views, while appendices B, C, and D provide specifications.
SECTION 2

METHODOLOGY

2.1 THE OBJECT-ORIENTED ANALYSIS TASK

Object-oriented analysis (OOA) identifies the data, control, and process of a domain, with that order of emphasis. (In contrast, the order of emphasis in functional decomposition is process - control - data, and in event-response analysis is control - process - data. Functional decomposition and event-response analysis are forms of structured analysis.) (Winblad, Edwards, and King, 1990). This paper is concerned with two special forms of OOA: domain analysis and problem analysis. Domain analysis is OOA which includes the classes and objects common to all systems within a given domain. Problem analysis is OOA restricted to the classes and objects of a particular problem or application system. [Throughout this paper, the acronym OOA is used only to represent object-oriented analysis as a methodology, and is not used to refer to a particular application of OOA.]

A task was devised to demonstrate that the objectives of the object-oriented analysis (stated in section 1.2) could be met. (An earlier task had shown that OOA was appropriate for use as an analysis tool in the present task, because object-oriented techniques [1] emphasize the integration of data and operations, rather than algorithms, as fundamental system building blocks and [2] embody an incremental, iterative software development process rather than the waterfall life cycle. The methodology particular to Peter Coad [Coad and Yourdon, 1991] was studied; it was decided that his methodology, perhaps modified and expanded, contained the essential elements of OOA and could be used as a starting point.) The objectives themselves were selected with the goal of showing that an object-oriented model of air traffic control has a use consistent with the I-Lab goals of (1) providing a capability to perform integration studies among individual NAS subsystems, procedures, and airspace, and (2) defining new operational concepts that will lead to increased system safety and capacity. While legacy software in the form of simulations, prototypes, and operational prototypes is expected to play a large supporting role in I-Lab experimentation, such software cannot be expected to cover every experimental need, given that I-Lab experimentation, by definition, looks at the horizontal integration of NAS components and most legacy software has been derived for a vertical application. The objectives were selected to provide a way to assess the ability of legacy software to meet experimental needs, by defining a "common language" in which both could be described before carrying out a direct comparison. The objectives also support the ability to define new software in this common language. The basic steps of the methodology used to meet the objectives are as follows:

(1) Create a model of ATC as a reference model, i.e., derive the "common language."

(2) Define the experiment of interest in the common language, i.e., create an experiment model.

(3) Define the available legacy software in the common language, i.e., create legacy software models.
(4) Directly compare the experiment model with the legacy software models, thus assessing the ability of the legacy software to meet experiment needs.

In the following description of the task, the term "model" is used to refer to a collection of information about a domain, including its structures, views, and specifications (see figure 2-1). Each
Figure 2-1. A Model
of these concepts is described further below in section 2.2: specifically, structures are described in section 2.2.3, views in section 2.2.5, and specifications in section 2.2.6.

The task was carried out in two subtasks, as illustrated by figure 2-2: a domain analysis of the ATC system followed by a problem analysis for a possible experiment. The domain analysis subtask resulted in an ATC model, which is described in more detail in section 3 and appendix A. The problem analysis subtask resulted in an experiment model and two legacy software models, based upon the ATC model and upon the written and verbal knowledge of the experiment and software authors. (Although there are more than two pieces of legacy software, it was decided that analysis of only two legacy software models was needed to demonstrate the methodology.) The process of using the ATC model as a reference model in creating new models, shown in the figure as information "passing through" the ATC model, is described in section 2.3. The experiment model and legacy software models themselves are described in more detail in section 4 and appendices B, C, and D. A direct comparison was made between experiment needs (represented by the experiment model) and legacy software capabilities (represented by legacy software models), and some conclusions were drawn concerning the ability of legacy software to meet experiment needs. The comparison of the experiment model with the legacy software models is shown formally in section 4.3.

The remainder of section 2 contains the following:

- Modeling concepts, including model components and notation (section 2.2)
- Model usage (section 2.3)
- Implications for object-oriented design (section 2.4)

The concept of OOA described in this paper is a consolidation of ideas from many sources, especially (Coad and Yourdon, 1991) and (Booch, 1991), augmented with original ideas sparked by the domain of interest. Unless stated otherwise, a reference to Coad or to Booch is a reference to (Coad and Yourdon, 1991) or to (Booch, 1991), respectively.

2.2 MODELING CONCEPTS

A result of an object-oriented analysis is a model of the domain. The building blocks of a model include subjects, classes, objects, structures, instance connections, message connections, and views. Detailed information about a model is contained in the model specification.

Figure 2-3 provides notation for the subject, class, object, attribute, service, instance connection, and message connection concepts described below. This notation is adapted from (Coad and Yourdon, 1991). Other notations are introduced below as needed. This paper also adopts the convention that the name of a model component (e.g., a class) is spelled with initial capital letters (e.g., the Aircraft class) and the name of a real-world instance of that component is spelled with lower case letters (e.g., the aircraft).

Figure 2-4 is part of a model which is used throughout this section to provide examples of concepts as they are introduced.
Unless otherwise stated, the conventions and terminology discussed in this section are those adopted for the methodology used by the task.
Figure 2-2. The Object-Oriented Analysis Task
Abstract Class - A class without objects - which is also a Generalization Class

Gen-Spec Structure

Class

<table>
<thead>
<tr>
<th>Name (top section)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute1</td>
</tr>
<tr>
<td>Attribute2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Service1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service2</td>
</tr>
</tbody>
</table>

Attribute (middle section)

Service (bottom section)

Whole Object

Whole-Part Structure

Whole

Part Objects

Part1

Part2

Instance Connection

Message Connection

Class and Object 1

Specialization Classes

Class and Object 2

Figure 2-3. Model Notation

© 2000 The MITRE Corporation. ALL RIGHTS RESERVED.
Figure 2-4. Examples of Model Components
2.2.1 Classes and Objects

An object is an abstraction of a real or theoretical entity, and is described by its attribute values and exclusive services. (From a programming point of view, an object is a program module that contains both its data and its procedures for operating on that data - no more and no less.) The three intrinsic characteristics of an object are its state, its behavior, and its identity. The state of an object takes into account its static properties and the static or dynamic values of these properties. The behavior of an object describes how the object acts or reacts as its state changes or as it sends or receives messages. The identity of an object is that property that uniquely distinguishes it from all other objects.

For example, in figure 2-4, an Airspace Surveillance System object (e.g., a particular air route surveillance radar [ARSR] system) has one attribute, Area of Coverage, and one service, Locate, Identify & Report Flight. The states of such an object might be "on," off," and "test." To illustrate change in behavior due to state change, the object would probably respond differently to the presence of an Aircraft object in its area of coverage (e.g., perhaps not at all) when the state is "test" versus when the state is "on."

When all objects in a collection have the same attributes and services, the term "class" is used to refer to them all collectively. For example, the set of all Airspace Surveillance System objects is represented by the Airspace Surveillance System class. An object can be considered to be a realization or an "instance" of a class, and the terms "object" and "instance" are frequently used synonymously. (Winblad, Edwards, and King, 1990) refers to classes as "static templates" and objects as "dynamic entities."

Although each object must belong to a class, it is not necessary for a class to have objects. An "abstract class" exists to provide structure for other classes; in one sense, its objects are really instances of its subclasses. In figure 2-3, the class called "Class" is an abstract class, indicated by the lack of a outer box. Each of the other classes can have objects. In figure 2-4, each of the classes pictured can have objects.

Coad believes that most classes and objects in a domain fall into the following categories:

- External systems (e.g., aircraft) with which the system under study will interact
- Devices (e.g., sensors) with which the system under study will interact
- Needed remembrances (e.g., a point in time or an historical event) that must be observed and recorded by the system under study
- Roles played (e.g., human roles [actions and interactions]) represented by the system under study
- Operational procedures which guide the human-computer interaction in the system under study
- Sites (e.g., physical locations) which must be known by the system under study
• Organizational units to which the humans belong

In addition, other classes and objects can be derived from existing classes and objects by considering the following types of structures:

• Classes which are specializations or generalizations of other classes (the "gen-spec" structure)
• Objects which are part of other objects or contain other objects as parts (the "whole-part" structure)

Structures are discussed in section 2.2.3.

Each object in a class can have its own value for each object attribute. According to (Shlaer and Mellor, 1988), the attributes of an object of a class fall into the following three categories:

• Descriptive attributes - characteristics intrinsic to the object
• Naming attributes - arbitrary names and labels which can be changed without changing the object itself
• Referential attributes - facts which tie the object to an object of another class

Schlaer and Mellor also describe identifier attributes, which uniquely distinguish each object of a class. In this paper, it is assumed that the use of identifiers is deferred to the object-oriented design step, making such attributes "implicit", i.e., invisible as attributes for analysis purposes. This paper does admit descriptive attributes which appear to be identifiers (e.g., the attribute Aircraft Identification of an Aircraft object) but are not because uniqueness is not assured.

Services (or "operations") are the behaviors that a class or object is responsible for performing. Services take into account the different states of the object, when a change in attribute values causes a change in object behavior. Coad places required services into two general categories: algorithmically-simple services and algorithmically-complex services. Algorithmically-simple services include the following:

• Create - to create a new object in a class and initialize its attribute values
• Connect - to establish or break off a connection between two objects
• Access - to get or set the attribute values of an object
• Release - to disconnect and delete an object

Each attribute and service is distinguished as being a class attribute or service or an object attribute or service; a class attribute is a quality of the class, not of an object, and a class service is a service
performed by the class, not by one of its objects. For example, the service "create" is usually considered to be a class service rather than an object service: a class can create an instance (an object) of itself, but an object cannot directly create another object. An object can bring about the creation of another object by sending a message to the appropriate class, requesting that class to perform the service create.

Algorithmically-simple services are treated as "implicit" services, and are not shown explicitly in the model notation. For example, the notation for an Aircraft object in figure 2-4 is empty in the services (lower third) part, because (at least in this example) an Aircraft object has only algorithmically-simple services.

Algorithmically-complex services include services of the following types:

- Calculate - to calculate a result from attribute values
- Monitor - to monitor an external system or device (e.g., to acquire data or control)

The service Locate, Identify & Report Flight service of an Airspace Surveillance System object in figure 2-4 is an algorithmically-complex service.

With respect to the data-control-process aspects of OOA (as described in section 2.1 above), process and data are hidden in classes and objects.

### 2.2.2 Instance Connections and Message Connections

Instance connections reflect associations between objects considered necessary because of timing or some other consideration. For example, figure 2-4 shown an instance connection between a Vehicle Surveillance System object and an Airspace Surveillance System object, created here because the aircraft is located in the area of coverage of the airspace surveillance system.

An instance connection between two objects reflects a constraint, in the form of an amount or range (also called the multiplicity) of mappings possible between the objects (e.g., exactly one connection must exist between two specific objects). An instance connection between two objects is defined in terms of one or more referential attributes. The multiplicity of an instance connection is shown in the model notation by numeric labels at each end of the instance connection line. A single number designates a fixed number of connections. A pair of numbers designates the lower and upper bounds of a range: when "0" is the lower bound, the connection is optional; when "m" is the upper bound, more than one connection is possible. For example, in figure 2-4, an aircraft is in the area of coverage of no, one or several airspace surveillance systems (represented by the "(0,m)" on the line between the Vehicle Surveillance System object and the Airspace Surveillance System object as the line joins the Airspace Surveillance System object) and there are no, one or several aircraft in the area of coverage of an airspace surveillance system (represented by the ",(0,m)" on the line as it joins the Vehicle Surveillance System object).

A message connection between a class or object and another class or object represents a request that the latter entity (the receiver) perform a service for the former entity (the sender) and return a
response. Message connections which would exist only to request implicit services (i.e., the algorithmically-simple services create, connect, access, and release) may also be treated implicitly (i.e., not shown explicitly) in the model notation. However, if considered necessary, such a message can be shown and labeled with the name of the implicit service spelled with lower case letters. For example, in figure 2-4, the message "access Location attribute" is a message sent by an Airspace Surveillance System object to an Aircraft object requesting the Aircraft object to perform its implicit service "access." The message Request Altitude Report is sent from an Airspace Surveillance System object to a Vehicle Surveillance System object to request the latter to perform its algorithmically-complex service Report Altitude.

With respect to the data-control-process aspects of OOA (as described in section 2.1 above), control is represented by messages between classes or objects.

2.2.3 Structures

Structure helps to clarify and reduce the complexity of a system under study. Two types of structures are recognized: generalization-specialization structures (Booch's "kind of" hierarchy called the "class structure"), and whole-part structures (Booch's "part of" hierarchy called the "object structure").

A generalization-specialization structure defines relationships between classes. A class is a generalization of a second class when the attributes and services of the first class all belong also to the second class. The second class is considered a specialization of the first class, and may contain additional attributes and services not belonging to the first class. The specialization class is said to inherit its attributes and services from the generalization class. In figure 2-4, the Transponder class is a specialization of the Vehicle Surveillance System class, and the Vehicle Surveillance System class is a generalization of the Transponder class. A Transponder object can perform the service Report Altitude because that service is inherited.

A generalization-specialization structure can be either a hierarchy (e.g., one generalization class with its specialization classes beneath it) or a lattice (e.g., more than one generalization class having one or more specialization classes in common). Booch uses the term "base class" for the generalization class.

A whole-part structure defines relationships between objects. An object is a whole object in a whole-part structure with a set of one or more part objects if the whole object can be considered to be made up of, or a container for, or a collection of, the part objects. There is no implicit relationship between the attributes and services of the whole object and those of a part object. A whole-part relationship also includes a multiplicity of relationships possible, similar to the multiplicity of instance connections. For example, in figure 2-4, an Aircraft object has no, one or several Vehicle Surveillance System objects as parts. A whole object usually has several part objects, whether from the same class (as in the figure) or from several different classes.

The collection-member form of the whole-part structure is especially useful for mental model abstractions, when either the "whole" or "part" is a concept rather than a "touchable" entity. For example, in ATC, it is useful to think of a Flight object with part objects Aircraft, Flight Plan, and
Flight Clearance. Only the Aircraft object is always touchable. A Flight Plan object might begin as touchable (e.g., a filed piece of paper) but quickly becomes only an electronic image. A Flight Clearance object represents an accumulation of verbal or electronic communications, each of which is at best an electronic image.

Generalization-specialization structures and whole-part structures can be combined together in versatile ways to represent more complex structures. Some examples of mixed structures include the following:

- An object of a specialization class can be decomposed in a whole-part structure (as in figure 2-3).
- A class containing a part object can be a generalization class in a generalization-specialization structure.
- A class can be a generalization class in a generalization-specialization structure while an object of the same class is a whole object in a whole-part structure (as in figure 2-5).

The last example above also illustrates a powerful side effect of some mixed structures: the inheritance of structure. Figure 2-5 shows that the Route class is a generalization class with the four
Figure 2-5. Inheritance of Whole-Part Structure
specialization classes Jet Route, Airway, Area Route, and Military Training Route, and a Route object is a whole object with two part objects, Fix/Waypoint and Route Segment. Each of the specialization classes inherit the whole-part structure of the Route object; e.g., an Airway object is also made up of one or more Fix/Waypoint objects and one or more Route Segment objects.

2.2.4 Subjects

A subject is a grouping of one or more related structures. A large, complex model might be organized into a small number of subjects, in order to simplify the presentation of basic concepts about the model. No subjects are shown in figure 2-4, but if the figure represented a complete model, two subjects called "Aircraft" (representing the structure containing the Aircraft class and object) and "Surveillance" (representing the class and object Airspace Surveillance System) might be considered. In this way, a presenter could talk about the interactions of Surveillance and Aircraft without worrying about what specific component within the Aircraft subject actually performed a service.

2.2.5 Views

Views are used to focus upon the limited subset of a model involved in a particular object state change, event, or sequence of events. Only pertinent details are shown; all other details are hidden.

Views usually show interactions among subjects. A typical view contains at least two objects (usually three or more selected from two or more of the subjects), only the instance connections needed by the view, two or three message connections, and only the attributes needed by the instance connections and the services requested by the message connections. Views are graphically arranged with subject components occupying the same relative positions in each view. Instance connections that exist at the beginning of an event being considered are shown by a solid line, whereas instance connections that are created dynamically during the event are shown by a dashed line. When the multiplicity of each instance connection is "1" for each involved object, it need not be shown. A message that requests an implicit service is shown with lower case letters to emphasize that it is considered an implicit message. A message between two classes or objects that allows two implicit services to interact need not be shown in a view.

An illustration of a typical view is shown in figure 2-6. This view contains a class and object from each of three subjects in a model. The views in this model always place the Manager subject components at the lower left of the view, the Resource subject components at the top middle, and the User subject components at the lower right.

The event illustrated is a request by a user-type object to use a resource-type object, and the resulting reactions. It is assumed that the user-type object does not now have permission to use the resource-type object; otherwise (in this model), an instance connection would exist between the user-type object and the resource-type object illustrating that the user-type object is taken into account in the Demand attribute of the resource-type object. The instance connection between the resource-type object and the manager-type object shows that there is only one resource-type object of interest, but possible several manager-type objects (e.g., there may be several ground controllers associated with a particular airport movement area). The instance connection between the user-type object and the
manager-type object shows that there is exactly one user-type object and exactly one manager-type object in this view.

The user-type object, in performing the service Plan Resource Usage, sends a message to the manager-type object; which manager-type object receives the message is determined by the instance connection between the user-type object and the manager-type object. The manager-type object accesses the attributes Demand, Saturation Threshold, and Usage Restrictions of the resource-type
Figure 2-6. View
object before making a decision. Finally, the manager-type object returns a response to the user-type object; the response is not shown explicitly, because it is assumed that the original message will have a response. Assuming that the request was granted by the manager-type object, the user-type class needs to add an instance connection between the user-type object and this resource-type object. This instance connection is shown as a dashed line to emphasize that the connection is created dynamically as a result of this event. The connect service of the resource-type object reacts to the addition of the instance connection by causing the Update Demand service of the resource-type object to be performed.

The view notation was constructed by the authors using parts of Coad's notation and some invented notation (e.g., the dashed line representing a dynamically-created instance connection).

2.2.6 Model Specification

A set of specifications is provided for each model. The specifications are ordered by class name and include the following:

- Class and object name

- Structure - an identification of the classes and objects directly above and beneath this class and its objects in a generalization-specialization or whole-part structure, specified by "part of," "contains parts," "specialization of," or generalization of"

- Instance connections - an identification of the instance connections between objects

- Message connections - an identification of the message connections between this class or object and other classes or objects; unless otherwise stated, message connections are assumed to be between objects

- Object states - an identification (in text, a table, a state transition diagram, Petri net, etc.) of the states and transitions between states of an object, possibly including attribute values associated with achieving each state

- For each attribute:
  - Attribute name
  - Attribute description - a textual description of the attribute, which might include constraints (such as default values or the impact of the values of other attributes) on the value of the attribute

- For each service:
  - Service name
- Service description - a description (by a bulleted list, service chart, or other device) of the processing performed by the service, including (when necessary) an identification of the object states associated with the service and/or how the service behaves differently depending on the state of the object

- External Inputs - information provided by sources external to the domain of study
- External Outputs - information provided to entities external to the domain of study
- Notes - additional information about the class which is not covered by any of the above categories

The form of the model specification is a variation of the specification described in (Coad and Yourdon, 1991).

2.3 MODEL USAGE

The models described in this paper were composed for the reasons stated in section 1.2. The ATC model (described in section 3.2) is both a model of the ATC system and a template which helps to describe parts (or proposed parts) of the ATC system with reference to the complete system.

This paper focuses on models of the ATC system. This paper does not attempt to describe an "I-Lab model," where the I-Lab itself is taken as the domain of study. An I-Lab model would have other components, corresponding to scenario generation, experiment control, experimental subject interface, and results extraction.

2.3.1 Reference Model

To use the ATC model as a reference model for an experiment or for legacy software (the "problem domain"), the modeler first identifies the components (e.g., classes, instance connections) of the problem domain using the components of the ATC model as the initial set of candidates. For example, the problem domain may reuse the class "Flight" from the ATC model. On the other hand, when there is no corresponding component in the ATC model, the modeler is free to add a problem-domain-unique component. For example, the problem domain may need a class "Current Plan" which does not exist in the ATC model. (The ATC model class "Flight Plan" is close to, but not the same as, the class "Current Plan.") The model notation for a problem domain model is modified to show the problem domain model's dependence on the ATC model. In a problem domain model, the "borrowed" ATC model components are shown with a shaded background, and problem domain component names unique to the problem domain are not. Some ATC model components are included to provide structure for the problem domain model, even though the problem domain might not have a component of the same name.

2.3.2 Evaluation of Experiment vs. Legacy Software

The first step in using the ATC model in an evaluation of experiment requirements vs. legacy software capabilities is to use the ATC model as a reference model as described in section 2.3.1
above. Once an experiment model and one or more legacy software models have been defined using the ATC model as a common template, the experiment model can be compared directly with the legacy software models to see if the legacy software can be used to carry out the experiment. The comparison is done by component (e.g., experiment class vs. legacy software class, experiment view vs. legacy software view). When an experiment component has no matching legacy software component, the legacy software alone is not sufficient to carry out the experiment. When two components match, the experimenter can further evaluate the suggested legacy software component to see if it can be used. For example, the legacy software component may be exactly what the experimenter had in mind, or it may need minor additions or subtractions in functionality, or it may carry along with it additional "unneeded" functionality which must be evaluated as extra overhead in the experiment. When the experiment model is compared with several legacy software models simultaneously, it is also possible that the legacy software models themselves overlap with each other or that each legacy software model provides only part of the functionality needed by the experiment, with no one of them sufficient for the experiment by itself. In this case, decisions must be made concerning how to use the legacy software (e.g., modify each one, augment one with the other's needed functionality, rewrite).

2.3.3 Extensibility and Reusability

In section 2.4.2, an implicit assumption is made that the experiment and legacy software are "looking at" ATC from the same point of view as the ATC model. Using the ATC model as a reference model primarily involved selecting existing classes and objects and creating specializations of them. However, the ATC model should be "reusable" for other modeling efforts with different points of view. Many classes and objects (e.g., Aircraft, Airspace) should be commonly needed regardless of the point of view; it might be possible to use their attributes and services "as is," modified, or augmented. For example, the ATC model does not now include the concept of an ATC facility; the "parts" of a facility are available, but they are not related in a way that would be meaningful to someone interested in facilities. A facility object could be defined as made up of the appropriate existing part objects selected from manager-, airspace/ground-, surveillance-, navigation-, communications-, and weather-type objects. Thus, with a new whole-part structure, the ATC model could be extended to include the concept of facility.

2.4 IMPLICATIONS FOR OBJECT-ORIENTED DESIGN

The ATC model (and hence the models based upon it) has some classes which are closely related (see section 3.2.3). Booch discusses the concept of "friend," which he defines as a "method [service] typically involving two or more objects of different classes, whose implementation for any one class may reference the [normally hidden] parts of all the corresponding classes that are also friends." Because of the tight coupling, it might be appropriate to consider the use of friends in any design based upon the models in this paper.

In addition, if legacy software is to be kept "intact," it may be necessary to link the common components in some way. For example, each legacy software model contains an Aircraft class. The use of friends may be the answer here also.
SECTION 3
DOMAIN ANALYSIS

3.1 ATC

This section describes the scope of study for the ATC domain model, motivates the selection of subjects and classes through a brief review of the genesis of air traffic control, and presents the ATC model.

Because of size limitations of the printed page, the figures in sections 3 and 4 which contain parts of models show only class and object names and not attributes and services. Attributes and services are described in the model specification appendices.

3.1.1 Scope of the ATC Model

The broad scope of study is ATC in the AERA 2 timeframe, i.e., the year 1999 and approximately 10 years beyond. The scope is limited by the following:

- The amount of resources (time and staffing) available - For example, the depth of study in any one area is limited.

- A (subjective) consideration of I-Lab objectives - For example, horizontal (across application) details are emphasized over vertical (within application) details. Applications such as the following which are unlikely to be the subject of experiments within I-Lab were omitted:
  - Research and development (e.g., activities of the Research and Development Computer Complex), except that prototypes or models resulting from research and development might be considered at some future time as legacy software
  - Maintenance (e.g., the Remote Maintenance Monitoring System)
  - Backup (e.g., facility backup, hardware backup provided for reliability)
  - Training (e.g., activities carried out by the FAA Academy or at facilities, or functionality of the dynamic simulator [DYSIM])
  - Testing (e.g., activities directed toward system testing and verification)
  - Military ATC (i.e., functionality unique to facilities operated by the military services)
  - Foreign ATC (i.e., ATC services provided by foreign countries)
- Supervisory (i.e., non-operational) duties

- Weather (i.e., the provision of wind and weather sensors and the collection and processing of weather data from sensors), except that weather products and meteorologist support to ATC are included

Figure 3-1 categorizes NAS elements and subelements as in the domain, partially in the domain, or not in the domain of study, subject to the restrictions noted above.

In performing the analysis of the ATC domain, it was desirable to eliminate design from the resulting model. That is, the model should document what the ATC system does rather than how it is or will be implemented. However, at this stage of development of the ATC system, almost every component is the result of some past design decision. These decisions were not necessarily explicit design decisions in today's sense of designing an automated system, but rather were practical approaches to solving urgent problems. For example, flight plans were devised fifty years ago to provide information concerning the planned route of an aircraft in flight (as well as other information). The flight strip was created as the automatically printed version of the clearance associated with the flight plan, and will have evolved into an electronic flight strip during the time period of interest. All of these components of ATC - the flight plan, the printed flight strip, and the electronic flight strip - represent design. The analysis attempted to extract the requirement ("what") from the design ("how"). However, the decision was made to keep a small number of components of ATC, even though they represent design to some extent, because they are now an integral part of ATC and are unlikely to change much during the time period of interest. (Changes are more likely to occur in the form of the component than in the concept supporting it.) Some examples of retained design are the flight plan, the clearance, and the route. (Each of these classes are described in later sections.)

It was also desirable, in performing the analysis of the ATC domain, to make the resulting model as modular as possible. While the ATC model is forced by its structures to be cohesive, thus increasing modularity, it also contains close coupling between its subjects, thus decreasing modularity. An alternative to the tight coupling would be to eliminate one of the subjects, subsuming the services of its classes into the second subject. However, the tight coupling in this case is a better reflection of the ATC system.

### 3.1.2 Evolution of ATC from Air Traffic

In a narrow sense, ATC is, as the name implies, the control of air traffic. Before ATC was created, the air traffic world consisted of aircraft to transport passengers and goods, the air (space) in which the aircraft flew, and the airports or fields (ground) which provided a surface for takeoffs, landings, and storage of the aircraft, and for the loading and unloading of the passengers and goods. A representation (model) of this air traffic world showing subjects and structures with two levels of classes is given in figure 3-2. Because the collection of aircraft airborne at the same time was not considered traffic, in the same sense that pedestrians or vehicles along a walkway or street are considered traffic, the class Aircraft is used instead of a class more representative of an organized stream or collection of aircraft. Nevertheless, the model is called an air traffic model to distinguish it from the ATC model described later in this paper.
There are three subjects in the air traffic model, divided in the figure by shaded lines:

- Manager, represented by the class Pilot
- Resource, represented by the classes Airspace, Ground and Aircraft
- User, represented by the classes Passenger, Goods, and Aircraft

An Aircraft object is both a user (of the resources airspace and ground) and a resource (for the users passenger and goods). The pilot is the manager of the aircraft.
### NAS Elements vs. ATC Model

<table>
<thead>
<tr>
<th>Element</th>
<th>Subelement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air Traffic Control</td>
<td>DUAT Service Vendor</td>
</tr>
<tr>
<td></td>
<td>Area Control Facility</td>
</tr>
<tr>
<td></td>
<td>Automated Flight Service Station</td>
</tr>
<tr>
<td></td>
<td>Airport Traffic Control Tower</td>
</tr>
<tr>
<td></td>
<td>Traffic Management Facility</td>
</tr>
<tr>
<td></td>
<td>National Aviation Weather Processing Facility</td>
</tr>
<tr>
<td>Ground-to-Air</td>
<td>Weather Sensing Facilities</td>
</tr>
<tr>
<td></td>
<td>Landing Facilities</td>
</tr>
<tr>
<td></td>
<td>Remote Communications Facilities</td>
</tr>
<tr>
<td></td>
<td>Navigation Facilities</td>
</tr>
<tr>
<td></td>
<td>Surveillance Facilities</td>
</tr>
<tr>
<td>Maintenance and Operations Support</td>
<td>Remote Maintenance Monitoring System</td>
</tr>
<tr>
<td></td>
<td>National Radio Communications System</td>
</tr>
<tr>
<td></td>
<td>Training Facilities</td>
</tr>
<tr>
<td></td>
<td>System Support Facilities</td>
</tr>
<tr>
<td></td>
<td>Environmental Facilities</td>
</tr>
<tr>
<td></td>
<td>Maintenance Control Center</td>
</tr>
<tr>
<td>Interfacility Communications Element</td>
<td>NAS Interfacility Communications System</td>
</tr>
</tbody>
</table>


**Key to NAS Elements/Subelements**

- In Domain
- Partially In Domain
- Not In Domain

---

Figure 3-1. NAS Elements vs. ATC Model
Figure 3-2. Air Traffic Model: Subjects
Because the aviation world has changed over time, its representation must also change from the simple air traffic model. The control aspects of ATC added many classes to air traffic, to represent functionality such as the following:

- Planning, monitoring, and control responsibilities of human specialists
- Communications between human specialists and aircraft, to transmit information and control instructions
- Surveillance of air traffic, to determine the location, altitude, and speed of aircraft
- Weather product availability, to assist in planning aircraft movement between locations

The addition of air traffic control also resulted in the demarcation of airspace and ground into well-defined smaller subsets because of human limitations in controlling large numbers of aircraft. Some pieces of airspace were defined for special purposes, to allow for military training activities without endangering civilian aircraft. With more humans involved, a greater communications network was needed to support coordination between individual controllers.

The navigation of the aircraft itself became more sophisticated, taking advantage of navigation aids in flying between defined locations.

### 3.2 THE ATC MODEL

The ATC model also contains only three subjects: User, Resource, and Manager. Each subject contains a structure with a base class of the same name as the subject. In addition, the Manager subject is contained in the User subject. Figure 3-3 shows the three subjects of the ATC model; only the second level of classes is shown.

In general, the following definitions apply to objects of user-, resource-, and manager-type classes (and to specializations of such classes):

- A user is a person or inanimate object which avails itself of a resource, or carries out a purpose or action by means of a resource.
- A resource is a source of supply or support, or a source of information or expertise.
- A manager is a person or inanimate object which handles or directs with a degree of skill, or one that alters by manipulation, or a supervisor, executive, or administrator.

#### 3.2.1 The User Subject

The User subject (see figure 3-3 again) contains two structures: one headed by the Manager class (i.e., the Manager subject) and the other headed by the Traffic class. The Manager class and its
structure is discussed in section 3.2.3 below. The remainder of this section concentrates on the Traffic class structure.

The Traffic class of the ATC model (see figure 3-4) is an expansion of the Aircraft class of the air traffic model. Because of the shift in emphasis from air traffic (i.e., the aircraft) to air traffic control, the other user classes of the air traffic model (the Passenger class and the Goods class) do not appear in the ATC model, and the aircraft is no longer considered to be a resource. In other words, the user
Figure 3-3. ATC Model: Subjects
Figure 3-4. ATC Model: User Subject (Traffic)
in the ATC model is a user of resources from an air traffic control viewpoint, rather than a user of the Aircraft as a resource.

3.2.1.1 The Aircraft Class

Because an Aircraft object is no longer considered a resource, a Pilot object is no longer considered a manager and becomes instead a part of the Aircraft object. Other parts are added to the Aircraft object to reflect its interaction with air traffic control resources, including the Vehicle Surveillance System object, the Vehicle Navigation System object, and the Vehicle Communications System object. The Transponder class is provided as a specialization of the Vehicle Surveillance System class, because the term "transponder" is a more common term for that equipment.

3.2.1.2 The Ground Vehicle Class

ATC at an airport applies to ground vehicles, such as fuel trucks and baggage carts, as well as to aircraft. As with an Aircraft object, the parts of a Ground Vehicle object reflect interaction with air traffic control resources. These parts include the Ground Vehicle Operator object (analogous to the Pilot object), the Vehicle Surveillance System object, the Vehicle Navigation System object, and the Vehicle Communications System object. The Transponder class is still appropriate as a specialization of the Vehicle Surveillance System class.

3.2.1.3 The Flight and Vehicle Classes

Because control has been introduced for vehicles, the simple Aircraft and Ground Vehicle classes are not sufficient by themselves. The Flight class and the Vehicle class are added to reflect new functionality, taking into account the following factors:

• The type of vehicle and its location (in the air or on the ground) when control applies. An aircraft, when it is on the ground, is considered a vehicle as well as part of a flight. (Stated in object-oriented terms, an Aircraft object can simultaneously have a whole-part relationship with both a Flight object and a Vehicle object.)

• The formality inherent in control when it exists (the planning initiated for the movement of a vehicle, as reflected by a plan, and the approval to move as planned, as reflected by a clearance). A Flight object is considered to have three intrinsic parts: zero to many Aircraft objects, at most one Flight Plan object, and at most one Flight Clearance object. Similarly, a Vehicle object has three parts: an Aircraft object or a Ground Vehicle object, a Ground Vehicle Route object, and a Ground Clearance object.

The Flight Class. The rationale for the whole-part relationship between a Flight object and its parts (Aircraft object, Flight Plan object, and Flight Clearance object) is most easily understood for an en route instrument flight rules (IFR) flight. In this case, by necessity, there is an aircraft (at least one), a filed (perhaps amended) IFR flight plan, and an en route clearance (perhaps changed several times) to fly according to the flight plan. (On the other hand, for an en route visual flight rules [VFR] flight, there must be an aircraft, there may be a VFR flight plan, and there is no en route clearance.) Throughout the flight, the attributes of the Flight Plan object and the Flight Clearance
object are updated to reflect the current values for the flight plan and clearance, as amended during flight.

A Flight Clearance object can also have three parts, a Departure Clearance object, an En Route Clearance object, and an Arrival Clearance object, although probably not all three at the same time. For most of the duration of a flight, the Flight Clearance object has only an En Route Clearance part. The Flight Clearance object usually has at most two parts: either a Departure Clearance object and an En Route Clearance object when the flight is awaiting departure (from a controlled airport) or is in the process of departing, or an En Route Clearance object and an Arrival Clearance object when the flight is about to begin its approach procedures (to a controlled airport). Thus, an IFR flight which has pushed back from the gate and is awaiting takeoff, and which has received instructions to use a Standard Instrument Departure (SID) in takeoff, is represented by a Flight Object with the parts Aircraft object, Flight Plan object, and Flight Clearance; the Flight Clearance object itself has the parts En Route Clearance object and Departure Clearance object.

The Vehicle Class. On the other hand, the Vehicle object always has three parts, since a vehicle can be present on the airport movement area only when the vehicle operator has a plan for its movement (its route) and a human specialist has cleared it to follow this route. The plan for vehicle movement is not a formal instrument in the same sense as the flight plan, but the planned route of the vehicle must be known before movement (even if only seconds before) to the human specialist and the vehicle operator. The ground clearance also might only be given minute-by-minute and not completely specified from source to destination. However, at any point in the vehicle movement, a Ground Vehicle Route object reflects the currently-planned route through the airport movement area and a Ground Clearance object reflects the currently-issued clearance.

3.2.14 The Clearance Class

A clearance is an authorization for flight or vehicle movement. In this ATC model, a clearance is the accumulation from the present time of all outstanding clearances and control instructions (those not yet carried out and those in the process of being carried out). ATC instructions (e.g., "turn left heading nine zero," or "clear the runway") issued to a flight (or vehicle) are considered to be amendments to the flight clearance (or ground clearance, respectively) when such instructions change the route, altitude, or speed to be used (where route, altitude, and speed refer to dimensions of movement rather than only to fields in a flight plan).

The Clearance class was created as a convenience to be the generalization class for all other clearance-type classes. The specialization classes of the Clearance class are the Flight Clearance, Ground Clearance, Departure Clearance, En Route Clearance, and Arrival Clearance classes. Although a clearance is not a user, it is part of a user and therefore clearance-type classes are placed in the User subject.

3.2.1.5 The Air Traffic, Ground Traffic, and Traffic Classes

An Air Traffic object allows a grouping of different Flight objects, so that a distinction can be made between when control applies to many flights versus one flight. An Air Traffic object is a whole object in a whole-part relationship, with Flight objects as part objects. A particular Air Traffic object
has a defined criterion or set of criteria which determine the Flight object parts. The Air Traffic class is flexible enough that an Air Traffic object can be defined to represent only a few flights, or all the flights in a sector of airspace, or all the flights currently under control in the ATC system. Similarly, a Ground Traffic object forms a whole-part relationship with Vehicle objects. Finally, the Traffic class is a generalization class with the Air Traffic and Ground Traffic classes as specialization classes. The Traffic class is a specialization of the User class.

### 3.2.2 The Resource Subject

The Resource subject (see figure 3-3) contains one structure, headed by the Resource class. The Resource subject of the ATC model is an expansion of the Resource class of the air traffic model, with the exception that (as stated above) the Aircraft class is removed from the Resource subject. The Resource class is a generalization class with five specialization classes: the Airspace/Ground Resource class, the Surveillance Resource class, the Navigation Resource class, the Communications Resource class, and the Aviation Weather Resource class. The last four classes are not found in the air traffic model.

Each of these five specialization classes is further decomposed into a whole-part structure, rather than a generalization-specialization structure, because the user of the ATC system typically looks at each of these resources as one entity and is not always concerned with exactly which lower-level part is being used. For example, a flight may fly out of one sector and into another, and may be aware of this change in sectors because a radio frequency must be changed, but the flight is not otherwise impacted by the knowledge of exactly which sector it occupies. Similarly, when a flight leaves one area control facility for another with a change in radio frequency, the flight is not impacted by a change in the radio transmitter/receiver used by the ATC ground station (the Air/Ground Communications System object).

Although whole-part decomposition has been used for conceptual clarity, it might be useful to introduce generalization-specialization relationships in addition for ease of implementation. For example, many of the classes in the structures beneath the Airspace/Ground Resource class have the common attributes Capacity, Demand, and Load. Creating a generalization-specialization structure in which each of these classes is a specialization of the Airspace/Ground Resource class would permit such attributes to be inherited directly. With only whole-part structures, the attributes must be redefined for each pert.

The Airspace/Ground Resource class heads a more detailed structure than the Surveillance Resource, Navigation Resource, Communications Resource, or Aviation Weather Resource classes, because it is expected to receive more emphasis in experimentation. The bottom of the Aviation Weather Resource structure is a whole-part decomposition into products and the management of these products. The management is represented by objects separate from the product objects because the management is automated in some cases and manual (human) in others. The Surveillance Resource, Navigation Resource, and Communications Resource structures could also have lowest-level whole-part decompositions into product and management objects. These decompositions were not done because the management of the products is expected to be automated and hence are subsumed as services of the lowest-level objects (i.e., the lowest-level object contains both product and management).
### 3.2.2.1 The Airspace/Ground Resource Class

The Airspace and Ground classes of the air traffic model are renamed the Controlled Airspace and Airport Movement Area classes, respectively, to reflect the formalization of the structure of the airspace and of the ground. A new Airspace/Ground Resource class (see figure 3-5) is introduced to head the structure; an Airspace/Ground Resource object is a whole object with Controlled Airspace and Airport Movement Area objects as its part objects. (Figure 3-5 is a two part figure; the two parts have the Route and Terminal Route classes in common.)

When the new airspace system is in effect (c. 1995), airspace structure in the contiguous 48 states will be designated by letters (Luffsey, 1990). Luffsey defines the new airspaces as follows:

**Airspace A.** For airspace at 18,000 feet and above; replaces positive control airspace.

**Airspace B.** Identical to current Terminal Control Areas.

**Airspace C.** Identical to current Airport Radar Services Areas, typically covering airspace from the surface to 4,000 feet above the surface.
Figure 3-5. ATC Model: Resource Subject (Airspace/Ground Resource) (Part 1)
Figure 3-5. ATC Model: Resource Subject (Airspace/Ground Resource) (Part 2)
Airspace D. This replaces current Airport Traffic Areas and Control Zones. All Class D airspace will have a common ceiling at 4,000 above the surface.

Airspace E. It replaces all other control airspace ..., including transition areas, controlled areas (airways), and the continental control area. It will be shown on charts. Covers the airspace between the surface and Flight Level 180, or the ceiling of uncontrolled airspace and FL 180.

Airspace G. Corresponds to uncontrolled airspace. Reaches from the ground to the indicated floor of controlled airspace.

With respect to the ATC model, the En Route Controlled Airspace object corresponds to the union of Airspace A and Airspace E, a Terminal Control Area object to a terminal control area (TCA) of Airspace B, an Airport Radar Service Area object to an airport radar services area (ARSA) of Airspace C, and an Airport Traffic Area to an airport traffic area (ATA) of Airspace D.

The Controlled Airspace Class. A Controlled Airspace object is made up of Oceanic Controlled Airspace, En Route Controlled Airspace, and Terminal Controlled Airspace objects as parts. The whole-part structure was chosen because it emphasizes, better than the generalization-specialization structure, the physical decomposition of airspace (even though, as specialization classes, the three classes could have inherited attributes and services from the Controlled Airspace class).

The Oceanic Airspace Class. An Oceanic Controlled Airspace object is made up of Oceanic Airway objects. Non-radar methods are used by ATC for separation purposes, because position reporting rather than radar surveillance must be used for locating aircraft. Hence, for ATC purposes, oceanic airspace is composed of airways rather than of volumes of airspace. (Oceanic airspace volumes might be used in the future if satellite-based surveillance were to be used.) Two types of oceanic airway are the track and the route. A track is a high-altitude one-way airway defined by whole degrees of latitude and longitude. A route is a low-altitude one-way airway defined (as are en route airways) by NAVAIDs. The airspace occupied by a route is defined by its fixes (e.g., VORs, NDBs) and route segments. Thus, the Oceanic Airway class has two specialization classes, the Track class and the Oceanic Route class. An Oceanic Route object has Fix/Waypoint and Route Segment objects as parts, as does a Track object.

The En Route Controlled Airspace Class. The En Route Controlled Airspace object is made up of Route objects and Airspace Volume objects. In some senses, this whole-part relationship introduces redundancy, because any route must occupy airspace (i.e., the intersection of the airspace represented by a Route object and the airspace represented by the appropriate Airspace Volume object is not empty). An alternative representation would be to place the Route class as a specialization of the Airspace Volume class, which only defers the redundancy problem to a lower level in the structure. However, the whole-part structure is believed to be closer to the ATC perspective. An implication for design is that any third object connected to a Route object (by an instance connection) may need to be connected also to the appropriate Airspace Volume object, and vice versa.
The Route class is involved in two structures: a whole-part relationship between a Route object and Fix/Waypoint and Route Segment objects (as does the Oceanic Route object, above) and a generalization-specialization relationship between the Route class and the Jet Route, Airway, Area Route, and Military Training Route classes. While the specialization classes do not add attributes or services to distinguish them from the generalization class, these classes are included because they are commonly differentiated from each other in the ATC world on the basis of aircraft equipage and performance (which, in turn, affects a manager's ability to perform certain services).

The Airspace Volume class was introduced to permit the definition of any arbitrarily-defined volume of airspace. The list of its specialization classes (ACF Airspace [to represent the airspace of an area control facility], Sector, Protected Airspace Volume, and Holding Pattern Airspace) is not meant to be exhaustive, but includes commonly-defined airspace types. Note that an ACF Airspace object has a whole-part relationship with Sector objects, while the Sector class is also a specialization of the Airspace Volume class.

**The Terminal Controlled Airspace Class.** A Terminal Controlled Airspace object, similar to an En Route Controlled Airspace object, is made up of Terminal Route and Terminal Airspace Volume objects.

The Terminal Route class is involved in two structures: a whole-part relationship between a Terminal Route object and Fix/Waypoint and Route Segment objects (as does the Oceanic Route object, above) and a generalization-specialization relationship between the Terminal Route class and the Standard Terminal Arrival Route (STAR), Standard Instrument Departure Route (SID), and Preferential Route classes. The Preferential Route class has three specialization classes: the Preferential Departure Route (PDR) class, the Preferential Arrival Route (PAR) class, and the Preferential Departure and Arrival Route (PDAR) class. A STAR object can have a whole-part relationship with a PAR object (with the PAR as the part), and a SID object can have a whole-part relationship with a PDR object (with the PDR as the part).

The Terminal Airspace Volume class has three specialization classes: the Terminal Control Area (TCA) class, the Airport Radar Services Area (ARSA) class, and the Airport Traffic Area (ATA) class. While the specialization classes do not add attributes or services to distinguish them from the generalization class, these classes are included because they are commonly differentiated from each other in the ATC world. In addition, this decomposition of a Terminal Controlled Airspace object does not represent all of the airspace around an airport, but instead focuses on those types of airspace for which capacity and saturation are of interest. Note that a particular PDAR usually is physically contained in more than one terminal airspace volume.

The Preferred IFR Route class is introduced here, although it is related to both en route airspace and terminal airspace. A Preferred IFR Route object has a whole-part relationship with the part objects Route, PDR, and PAR.

**The Airport Movement Area Class.** The Airport Movement Area class represents the ground surface used by ground vehicles and aircraft at an airport. For consistency, a helicopter that is taxiing around an airport (whether ground taxiing, hover taxiing, or air taxiing) is considered to be using the Airport Movement Area. A particular Airport Movement Area object can have a whole-
part relationship with one or more Runway objects, none to several Taxiway objects, and none to several Holding Area objects.

3.2.2.2 The Surveillance Resource Class

Figure 3-6 contains the structures for the following four specializations of the Resource class: the Surveillance Resource class, the Navigation Resource class, the Communications Resource class, and the Aviation Weather Resource class. As described in section 3.2.2, these structures are less detailed than the Airspace/Ground Resource structure, because the corresponding resources are expected to receive less emphasis in I-Lab experimentation. This section discusses the Surveillance Resource class.
Figure 3-6.  ATC Model: Resource Subject (Other Resources)
A Surveillance Resource object, made up of Surveillance System objects, represents the equipment for the tracking of aircraft positions and the management of the information provided (e.g., the reporting of positions). Here, "management" means "management of the product" or "management information system" rather than "equipment maintenance." In addition, the Surveillance System class refers only to the surveillance functionality of the ATC system: surveillance capabilities of an individual Aircraft object used in interacting with a Surveillance System object (e.g., the ability of an aircraft transponder to reply to interrogation by surveillance radar) are included in the Vehicle Surveillance System object that is a part of the Aircraft object.

Specializations of the Surveillance System class include the Airspace Surveillance System class and the Ground Surveillance System class. While the specialization classes do not add attributes or services to distinguish them from the generalization class, these classes are included because they are commonly differentiated from each other in the ATC world.

### 3.2.2.3 The Navigation Resource Class

A Navigation Resource object (see figure 3-6), made up of Navigation System objects, represents the navigation aids provided by the ATC system to assist aircraft in navigating between points and the management of the information provided (e.g., the broadcasting of a NAVAID position by radio signal). Again,"management" means "management of the product" or "management information system" rather than "equipment maintenance." Again, the Navigation System class refers only to the navigation functionality of the ATC system: navigation capabilities of an individual Aircraft object used in interacting with a Navigation System object (e.g., the ability of an aircraft VOR/DME-based RNAV unit to receive, interpret, and use radial and distance information from a VOR/DME facility) are included in the Vehicle Navigation System object that is a part of the Aircraft object. Also, the Navigation System class refers only to those navigation systems certified by the FAA to be a primary navigation system, for example, VOR-DME systems and LORAN-C systems, but not VLF/OMEGA systems. When satellite-based navigation is certified as primary, it can be added to the model.

The Navigation System class has one specialization, the Landing Navigation System class. The primary reason for including a Landing Navigation System class is that many landing navigation systems can provide elevation information to an aircraft whereas most navigation systems used in en route or terminal airspace do not. Thus, a Landing Navigation System object might provide the additional service Transmit Elevation but the more general Navigation System object can provide only the inherited services.

### 3.2.2.4 The Communications Resource Class

A Communications Resource object (see figure 3-6), made up of Communications System objects, represents the communications media provided by the air traffic control system for the transfer of information between the ATC system and aircraft (e.g., the transfer of a clearance amendment by radio from a human specialist to an aircraft) and among ATC system components (e.g., for coordinating a handoff). Again,"management" means "management of the product" or "management information system" rather than "equipment maintenance." Again, the Communications System class refers only to the communications functionality of the ATC system: communications capabilities of an individual Aircraft object used in interacting with a Communications System object
(e.g., the ability of an aircraft to receive the radio transmission of clearance information) are included in the Vehicle Communications System object that is a part of the Aircraft object.

Specializations of the Communications System class include the Interfacility Communications System class, the Intrafacility Communications System class, and the Air/Ground Communications System class. While the specialization classes do not add attributes or services to distinguish them from the generalization class, these classes are included because they are commonly differentiated from each other in the ATC world.

### 3.2.2.5 The Aviation Weather Resource Class

An Aviation Weather Resource object (see figure 3-6), made up of Aviation Weather System objects, represents the wind and weather products available to users of the ATC system (e.g., an area forecast, a surface aviation weather report) to assist in flight planning, and the management of basic wind/weather products in the creation of specialized products. (e.g., a preflight weather briefing). Again,"management" means "management of the product" or "management information system" rather than "equipment maintenance."

An Aviation Weather System object has two parts: Wind/Weather Product object(s) and Aviation Weather System Manager object(s). The Aviation Weather System Manager class is included in this ATC model for convenience, but the Aviation Weather System Manager class, strictly speaking, does not belong in this ATC model: the functionality represented by this class (e.g., the duties of a meteorologist) is not a function of ATC. For this model, the services of an Aviation Weather System Manager object are restricted to creating standard wind and weather reports and to creating specialized reports at the request of a Flight Manager object or a Traffic Manager object.

### 3.2.3 The Manager Subject

As stated in section 3.2.1 and illustrated by figure 3-3, the Manager structure belongs to both the Manager subject and the User subject.

The Manager subject (see figure 3-7) contains one structure, headed by the Manager class. Because the Pilot class has moved from the Manager subject of the air traffic model to the User subject of the ATC model, the Manager subject of the ATC model is completely different from the Manager subject of the air traffic model.

The structure and classes beneath the Manager class are closely coupled to the structure and classes beneath the Traffic class of the User subject, which is made obvious by the class names within both subjects.

The Manager class is a generalization of the Traffic Manager class, the Flight Manager class, and the Vehicle Manager class. The class names were selected to emphasize that the traffic manager manages traffic, the flight manager manages flights, and the vehicle manager manages vehicles. The Traffic Manager class is a generalization of the Air Traffic Manager Class and the Ground Traffic Manager class. As above, the air traffic manager manages air traffic, and the ground traffic manager manages ground traffic.
The manager classes are not synonymous with today's positions (e.g., air traffic controller, flight service station specialist, traffic management specialists, CARF specialists, traffic management coordinator). Rather, each manager class encapsulates the services inherent in managing a particular type of user (e.g., vehicle) and the airspace/ground resources it needs (e.g., airport movement area). However, any existing position can be characterized as a specialization of one or more of the manager classes. For example, the local controller could be considered a specialization of both the Flight Manager class (by accepting control of a flight and delivering a landing clearance to that flight) and the Vehicle Manager class (by controlling an aircraft's use of a runway resource).

As stated above, a manager-type object manages the use of an airspace/ground-type resource. The decision to place these resource management services in a manager-type object was made because, in
Figure 3-7. ATC Model: Manager Subject
the timeframe of interest, such management services will be shared by the human and automation. The services of a manager-type object do not make a distinction between the human and automation.

In contrast, the management of the other types of resources, which is expected to remain automated, is left encapsulated within the resource-type object. An Aviation Weather System Manager object, which does contain services performed by the human, is not properly within the ATC model (i.e., is not an ATC manager) and is left as a part of an Aviation Weather System object.

As part of the User subject, a manager-type object uses resources, including communications systems and aviation/weather systems. A manager-type object needs to use an Air/Ground Communications System object in order to talk with a Pilot object, an Intrafacility Communications System object to talk with another manager-type object in the same facility, and an Interfacility Communications object to talk with a manager-type object in another facility. A manager-type object needs to use aviation weather reports and the services of an aviation/weather system manager in order to get wind and weather information to assist flights in flight planning and weather avoidance.

### 3.2.4 INSTANCE CONNECTIONS

Because a relationship between a class (or an object) and another class (or object) in the same subject tends to be represented by a generalization-specialization structure or a whole-part structure, the instance connections in the ATC model (with one exception) represent relationships between objects in different subjects. The one exception is that two or more Flight objects (both in the User subject) might be connected by instance connections, as described in section 3.2.4.4 below.

Figure 3-8 summarizes the instance connections of the ATC model. The attributes shown in the figure are those used in establishing specific instance connections. The structure beneath the Traffic class and object (which does not represent the entire structure) is provided only as a convenient framework to facilitate understanding the various relationships.

The figure also clarifies exactly which classes and objects are "user-type" or "manager-type" classes and objects. A "user-type" object is any user of a resource. User-type refers to the following classes and objects: Traffic, Air Traffic, Ground Traffic, Flight, Vehicle, Aircraft, Ground Vehicle, Vehicle Surveillance System, Vehicle Navigation System, Vehicle Communications System, and all the classes and objects in the Manager subject.

A "manager-type" object is any manager of a resource, in the sense of planning the usage of the resource or granting permission to use it. Manager-type refers to the following classes and objects: Traffic Manager, Air Traffic Manager, Ground Traffic Manager, Flight Manager, and Vehicle Manager. The names of the manager-type classes parallel the names of the user-type classes. Notice that there are no managers for Surveillance System, Navigation System, or Communications System objects, because such objects are essentially first-come first served with no explicit requests for usage. The management of such objects is assumed to be a service of the resource itself.

A "resource-type" class or object is any class or object from the structure headed by the Resource class and object; an "airspace resource-type" class or object is any class or object from the structure...
headed by the Controlled Airspace class and object; and a "ground resource-type" class or object is any class or object from the structure headed by the Airport Movement Area class and object.

The following sections describe the instance connections in the ATC model and are organized by subject pairs. The views in section 3.2.6 provide specific examples.
Figure 3-8. ATC Model: Instance Connections
3.2.4.1 User-Manager Instance Connections

The instance connections between user-type objects and manager-type objects reflect the close coupling between users and managers in the ATC model. In each case, the instance connections exist because the user-type object is in the area of responsibility of the manager-type object. Typically, the instance connection between a Flight object and a Flight Manager object is created when the flight manager assumes control responsibility for the flight (e.g., via handoff) and deleted when control responsibility is given up. Before a flight is active, an instance connection might exist based on the fact that the Flight Manager is providing flight planning assistance; here, control responsibility may or may not be involved. An instance connection between a Vehicle object and a Vehicle Manager object typically exists to show control responsibility.

An instance connection between a traffic-type object (Traffic, Air Traffic, or Ground Traffic object) and a traffic manager-type object (Traffic Manager, Air Traffic Manager, or Ground Traffic Manager object) exists to facilitate the planning and management of airspace/ground-type resource usage. Such an instance connection is defined by that grouping of Flight objects or Vehicle objects with which the manager is concerned. Two different traffic-type objects, with instance connections to different traffic manager-type objects, might have Flight objects in common. For example, one Air Traffic Manager object might be interested in the planned airspace usage by all Flights in a given Area Control Facility (ACF), while another Air Traffic Manager object might be interested in all Flights in that ACF as well as in the adjacent ACFs.

3.2.4.2 Manager-Resource Instance Connections

An instance connection exists between a manager-type object and an airspace/ground resource-type object to show that the resource is in the manager's area of responsibility. The instance connection is created when the resource-type object is created, deleted when the resource-type object is deleted, and typically not otherwise changed. There are no instance connections between manager-type objects and objects of other types of resources (e.g., surveillance system objects) because the management of these other resources is encapsulated as services within the resource-type objects.

An instance connection between a traffic manager-type object (e.g., a Ground Traffic Manager) and an airspace/ground resource-type object (e.g., an Airspace Movement Area object) shows that the traffic manager-type object is responsible for the capacity planning and general usage planning of the resource-type object. An instance connection between another type of manager object (e.g., a Vehicle Manager object) and an airspace/ground resource-type object (e.g., a Taxiway object) shows that the manager-type object is responsible for the specific usage planning (by a Vehicle object or Flight object) and the granting of permission to use the resource-type object.

3.2.4.3 User-Resource Instance Connections

Instance connections between user-type objects and resource-type objects are created for different reasons, depending on whether the resource-type object is an airspace/ground resource-type object or another type.
**Instance Connections Between a User and an Airspace/Ground-Type Resource.** An instance connection between a user-type object and an airspace/ground resource-type object is based on either demand (i.e., the user-type object plans to use the airspace/ground resource-type object) or load (i.e., the user-type object is using the airspace/ground resource-type object). A Flight object that plans to use an airspace resource-type object has an instance connection with it and contributes to the value of its Demand attribute. Similarly, a Vehicle object that plans to use an ground resource-type object has an instance connection with it and contributes to the value of its Demand attribute. On the other hand, an Aircraft object with Location attribute value contained in the Location attribute value of an airspace resource-type object has an instance connection with that resource-type object and contributes to the value of its Load attribute. Similarly, a Vehicle object with Location attribute value contained in the Location attribute value of a ground resource-type object has an instance connection with that resource-type object and contributes to the value of its Load attribute.

**Instance Connections Between a User and Another Type of Resource.** An object of a Surveillance System class, Navigation System class, or Communications System class (or one of its specializations) has an instance connection with an object of a Vehicle Surveillance System class, Vehicle Navigation System class, or Vehicle Communications System class (respectively) when the Equipment Type attributes of the former and latter objects are compatible, the value of the Status attribute of each (when it exists) is "active" or "operational" or is otherwise appropriate, and the latter object is in the area of coverage of the former object. This relationship is somewhat like the "load" relationship above, but there is no corresponding "demand" relationship.

A manager-type object has an instance connection with an object of the Communications System class (or one of its specializations) when the manager-type object is in the area of coverage of the communications system and the status of the communications system is "operational."

### 3.2.4.4 User-User Instance Connections

A Flight object represents only a flight segment, that is, the flight from departure at one airport to arrival at the next airport. However, it is of interest sometimes to examine a sequence of flight segments. Instance connections between pairs of Flight objects can be created to define, for example, a flight itinerary or an airframe itinerary. A flight itinerary describes a "continuing flight" with stops en route (e.g., a flight known to passengers by a particular flight number) and can be used to study accumulated flight delay. An airframe itinerary describes the use of a particular aircraft ("airframe") during flight segments and can be used to track and schedule maintenance for that aircraft.

More than one instance connection is possible between the same two Flight objects, and other associations are possible. Of course, different attributes and rules are needed to represent each kind of instance connection. For example, the instance connection between two Flight objects of the same flight itinerary is based upon the following:

- The two flights have the same aircraft callsign (Aircraft Identification attribute of the Flight Plan object).
• The arrival airport (Destination attribute of the Flight Plan object) of the first Flight object is the same as the departure airport (Departure Point attribute of the Flight Plan object) of the second Flight object.

• The estimated arrival time (based on the Departure Time and Estimated Time En Route attributes of the Flight Plan object) of the first Flight object reasonable precedes the estimated departure time (Departure Time attribute of the Flight Plan object) of the second Flight object.

3.2.5 Message Connections

Message connections are not as easily depicted in one figure as are instance connections, and are described in appendix A. The views in section 3.2.6 provide examples of services.

3.2.6 Views

This section contains a small set of views, some considered to be of special interest and others merely illustrative, but all in line with the task scope to emphasize horizontal details over vertical details. Other views can easily be constructed. In one case, an object state diagram is provided because the view illustrates a change in object state.

The following views are included in this section:

• Clearance to enter airport movement area event

• Handoff to flight manager (flight state change) event

• Problem detection and resolution events
  - Aircraft-to-aircraft conflict event
  - Aircraft-to-airspace conflict event
  - Flow instruction problem event

• Airspace/ground saturation detection and resolution event

• Clearance delivery (communications) event

• NAVAID use (navigations) event

• Tracking (surveillance) event

• Weather events
  - Preflight briefing event
  - Weather area event
3.2.6.1 Clearance to Enter Airport Movement Area Event

Figure 3-9 is the view of an event in which a vehicle (in this case, an aircraft) is given permission ("clearance") to enter an airport movement area. At the beginning of the event, the Vehicle object is made up of only an Aircraft object (i.e., there is no Ground Vehicle Route object or Ground Clearance object). The Aircraft object has two parts of interest, a Pilot object and a Vehicle Communications System object. The Vehicle object has an instance connection with the Airport Movement Area object because the vehicle plans to use the airport movement area, but the Vehicle object does not yet have any instance connections with any other part of the airport movement area (e.g., taxiway) at this time because the vehicle has no plan for or permission for the use of any particular part of the airport movement area. The Airport Movement Area object, in turn, has instance connections with one or more Vehicle Manager objects, because any vehicle wishing to enter the Airport Movement Area must receive permission from a Vehicle Manager object. The instance connection between the Vehicle object and the appropriate Vehicle Manager object is created dynamically (shown by the dashed line) based on these other two connections and upon the fact that the Vehicle and the Vehicle Manager must use the same communications frequency.

Typically, a user-type object from the traffic hierarchy, an airspace/ground resource-type object, and a manager-type object exist as a triad connected by instance connections. The airspace/ground resource-type object needs an instance connection with a manager-type object to indicate which
Figure 3-9. Clearance to Enter Airport Movement Area Event
manager has the authority to grant permission to use the resource. The user-type object needs an instance connection with a manager-type object to request permission to use an airspace/ground resource. The airspace/ground resource-type object needs an instance connection with the user-type object to indicate that the demand by the user is accounted for.

The three instance connections are created as needed. The instance connection between an airspace/ground resource-type object and a manager-type object is created when the airspace/ground resource-type object is created, and is determined by the area of responsibility of the manager. When a Vehicle object is first created, an instance connection is created between the Vehicle object and the appropriate Airport Movement Area object. No other instance connections between the Vehicle object and parts of the Airport Movement Area object are possible until permission for use is granted and recorded in a Ground Vehicle Route object. Similarly, when a Flight object is first created, an instance connection is created between the Flight object and the Controlled Airspace object. Other instance connections can be made only after a Flight Plan is created. The instance connections between the user and the resource, and between the manager and the resource, typically exist first, and the appropriate instance connection between the user and the manager is determined based upon the first two connections and other appropriate information.

Although communications are deliberately not emphasized here, the pilot must use a frequency of an air/ground communications system and the active frequency of the vehicle communications system must be that same frequency. (See the clearance delivery [communications] event in section 3.2.6.5 also.) In this example, the Vehicle object already had an instance connection with an Airport Movement Area object, and a Vehicle Manager object already had an instance connection with the same Airport Movement Area object. Thus an instance connection could be created by the Vehicle class between the Vehicle object and the Vehicle Manager object once the common frequency was known, allowing the Pilot object to send messages to the correct Vehicle Manager object.

The Pilot object, in performing the service Request Movement, sends a message to the Vehicle Manager object with which the Vehicle object has an instance connection requesting a pushback clearance. The Vehicle Manager, in performing its service Generate & Deliver Clearance, must first determine the availability of a taxiway for use by the aircraft: this the Vehicle Manager does by accessing the Configuration attribute of the Airport Movement Area (to see what taxiways are appropriate) and the attributes of the appropriate Taxiways objects (to see if any can be used). The Vehicle Manager object has instance connections with the Taxiway objects within its area of responsibility. When an appropriate taxiway is planned to be available (i.e., capacity is predicted to be available to meet the new demand), the Vehicle Manager object returns a response to the Pilot object with the pushback clearance.

Upon receiving the pushback clearance, describing at least part of the approved route (e.g., which taxiway to follow), the Pilot object requests the Ground Vehicle Route class to instantiate a Ground Vehicle Route object with appropriate values for the Ground Route attribute, and requests the Ground Clearance class to instantiate a Ground Clearance object with appropriate values of the Route attribute. The Ground Vehicle Route object then requests the Vehicle class to establish an instance connection (shown by dotted line) between the original Vehicle object and the Taxiway object. The purpose of this instance connection is to establish that this particular Vehicle object is to be accounted for in determining the value of the Taxiway attribute Demand; in fact, the Vehicle object then sends a message to the Taxiway object requesting the Update Demand service of the
Taxiway object be performed. None of these instance connections are needed after the Vehicle object is deleted, which can happen once the vehicle leaves the airport movement area (e.g., the aircraft leaves the ground on takeoff).

This view looks at an Aircraft object as part of a Vehicle object. Considering the Aircraft object as part of a Flight object, a view of (say) a predeparture clearance could be constructed with Flight Manager, En Route Controlled Airspace, Flight, Aircraft, Flight Plan, and Flight Clearance objects. Thus an Aircraft object can be viewed simultaneously as part of both a Vehicle object and a Flight object.

3.2.6.2 Handoff to Flight Manager (Flight State Change) Event

The Flight object has five states (see figure 3-10): predeparture, departure, en route, arrival, and postarrival. The following short description applies to the usual flight; some details differ for particular flights (e.g., a VFR flight without a flight plan, a flight that files its flight plan after it is airborne). The predeparture state corresponds to that time before the aircraft makes use of the airport movement area of an airport; during this time period, the initial flight planning, including the receipt of weather briefings and the filing of the flight plan, is completed. The departure state corresponds to that time spent on the airport movement area of the departure airport for taxiing and takeoff and in the air until handed off to an en route controller. (There is no en route state for a tower en route flight.) The en route state lasts until the flight is given the arrival clearance for its destination airport. The arrival state lasts until the aircraft lands and leaves the airport movement area of the destination airport. The postarrival state exists essentially to allow for flight plan closing (although an IFR flight plan is considered closed upon arrival at a controlled airport). The states of a flight are defined to correspond to the assignment of the manager rather than to the airspace/ground resource used (although this is a fine point, because the managers and the airspace/ground resources are also closely related). That is, a change in flight state is associated with a change in manager rather than with a change in resource usage.

Figure 3-11 is a view of an event causing a change in flight state from departure to en route. At the beginning of the event, a Flight object exists with three parts of interest: an Aircraft object, a Flight Plan object, and a Flight Clearance object. The Flight Clearance object has two parts: a Departure Clearance object and an En Route Clearance object. The Flight object has an instance connection with a Flight Manager object, called here "Flight Manager 1." Flight Manager 1, in performing its service Monitor Flight Location, compares the boundary and altitude limits of the Terminal Control Area (TCA) to the aircraft location and determines that the flight is about to leave the TCA. From the flight plan, Flight Manager 1 infers that the aircraft is following its planned route and decides to attempt to transfer control responsibility to the flight manager ("Flight Manager 2") associated with the en route sector to which the flight is headed. Flight Manager 1 sends a message to Flight Manager 2 requesting the acceptance of control responsibility for the flight. Flight Manager 2 accesses the Capacity, Demand, Load, Saturation Threshold, and Usage Restriction attributes of the Sector object to determine if control of the flight should be accepted, replying to Flight Manager 1's message that control is accepted. Flight Manager 1 then performs its Direct Transfer of Communications service to request the Pilot object to send future communications to Flight Manager 2. The Pilot object sends a message to the Vehicle Communications System object requesting that the Active Frequency attribute be updated. The Vehicle Communications System
object, in turn sends a message to the Flight object requesting it to delete the instance connection between the Vehicle object and Flight Manager 1 and to create an instance connection between the Vehicle object and Flight Manager 2. The Pilot object also sends a message to the Departure Clearance class requesting it to delete the Departure Clearance object.

At the beginning of the event, there are instance connections between the Flight object and the Flight Manager object called Flight Manager 1 (because Flight Manager 1 controls the flight), between the Flight object and the Terminal Control Area object (because the flight still contributes to the value of the Demand attribute of the Terminal Control Area object), and between the Flight object and the Sector object (because the Flight object also contributes to the value of the Demand attribute of the Sector object). As a result of the transfer of control responsibility ("handoff"), the Flight class can delete the instance connection between the Flight object and Flight Manager 1, delete the instance
Figure 3-10. Flight States

© 2000 The MITRE Corporation. ALL RIGHTS RESERVED.
Figure 3-11. Handoff to (En Route) Flight Manager Event
connection between the Flight object and the Terminal Control Area object, and add an instance connection between the Flight object and Flight Manager 2. As a consequence of deleting the instance connection between the Flight object and the Terminal Control Area object, the Flight class also sends a message to the Terminal Control Area object requesting it to update the value of its Demand attribute.

The instance connections between the Flight object and airspace/ground resource-type objects are created as soon as the Flight Plan object part of the Flight object exists, are based on the resources to be used by the flight (as documented by the flight plan), and are used to update the Demand attribute of each applicable resource object. Each of these instance connections is updated with each update of the Flight Plan object and can be deleted as soon as the planned use of the resource-type object is no longer in the future. When the Flight class deletes an instance connection between a Flight object and an airspace/ground resource-type object, the Flight class must also send a message to the airspace/ground resource-type object requesting it to update the value of its Demand attribute.

The instance connection between the Aircraft object and the Terminal Control Area object indicates that the aircraft is occupying the TCA and that the aircraft is being taken into account in the Load attribute of the Terminal Control Area object. This instance connection is maintained by the Aircraft class upon request by the appropriate Surveillance System object (not shown here). The Aircraft class can delete the instance connection between the Aircraft object and the Terminal Control Area object as soon as the surveillance system detects that the aircraft has left the TCA, and, as above, the Aircraft class must send a message to the Terminal Control Area object to update the value of its Load attribute.

Throughout the event, there is an instance connection between Flight Manager 1 and the Terminal Control Area object, and between Flight Manager 2 and the Sector object, indicating that each of the airspace resource-type objects is in the area of responsibility of the appropriate manager-type object.

Note that services performed by the manager-type objects in this view (e.g., offering and accepting handoff, checking for availability of resources) might be performed by automation or by humans.

### 3.2.6.3 Problem Detection and Resolution Events

The following three types of problems in the AERA 2 timeframe are of interest:

- **Aircraft-to-aircraft conflict**: the actual or predicted violation of separation minima between two aircraft
- **Aircraft-to-airspace conflict**: the actual or predicted violation of separation minima between an aircraft and a protected airspace volume
- **Flow instruction problem**: the actual or predicted noncompliance of an aircraft with a flow instruction

This section contains a view illustrating an event involving each type of problem.
**Aircraft-to-Aircraft Conflict Detection and Resolution Event.** Figure 3-12 is a view of an aircraft-to-aircraft conflict event. At the beginning of the event, a Flight Manager object (called "Flight Manager 1") with an instance connection with the En Route Controlled Airspace object has been performing its service Detect, Predict & Report Separation Violation. In this event, a violation of separation minima is predicted between two IFR flights that are planned to be in the same sector at the same time (put another way, two Flight objects with type IFR and with instance connections to the
Figure 3-12. Aircraft-to-Aircraft Conflict Detection and Resolution Event
same Sector object in the same time period are predicted to have locations closer than separation minima). (This situation might arise because of earlier uncertainty in prediction, because neither flight would likely be granted permission to plan to cause an aircraft-to-aircraft conflict.) For this example, it is assumed that the two Flight objects now have instance connections to the same Flight Manager object, called "Flight Manager 2." Flight Manager 1 sends a message to Flight Manager 2, requesting the performance of the Separate IFR Flights service. In carrying out this service, Flight Manager 2 also performs the general service Generate & Deliver Clearance, requesting the pilot of one of the two Flight objects ("Flight 2") to perform its service Plan Flight. (If Flight Manager 2 had decided to move both aircraft, then a message would be sent to the Pilot object of Flight 1 as well.) If the Plan Flight service determines that the proposed change to the flight plan and clearance cannot be accepted, then a negative response is returned to Flight Manager 2; otherwise, a positive response is returned and the Pilot object requests the Flight Plan object to amend itself and requests the En Route Clearance object to amend itself. The attributes of the Flight Plan which are affected by the Amend Flight Plan service are determined by the actual amendment. Similarly, the affected En Route Clearance attributes are determined by the actual amendment.

This view makes use of two (possibly) different Flight Manager objects, called Flight Manager 1 and Flight Manager 2. Flight Manager 1, which is connected to the En Route Controlled Airspace object, is given the responsibility (the service) of determining aircraft conflicts, even though the conflict itself, in this example, is occurring in a lower-level part of the object. (It is not sufficient to have each sector checked independently for conflicts, because conflicts can occur between flights in two different sectors.) During the design phase, Flight Manager 1 and Flight Manager 2 may be defined to be the same object by placing an instance connection between Flight Manager 2 and each sector in its area of responsibility and each adjacent sector.

Note that the Flight Manager object is not necessarily a human being: the services of a Flight Manager object could be carried out partly by automation and partly by a human.

**Aircraft-to-Airspace Conflict Detection and Resolution Event.** Figure 3-13 is a view of an aircraft-to-airspace conflict event. In this case, the conflict is a detected violation of separation minima rather than a predicted violation (as in the previous view). At the beginning of the event, a Flight Manager object (called "Flight Manager 1") connected with a Protected Airspace Volume object (by instance connection) has been monitoring the actual locations of aircraft in the vicinity of the protected airspace volume as part of its service Monitor Flight Location. This monitoring is done by assessing the Location attribute of Aircraft objects within or adjacent to Flight Manager 1's area of responsibility. The service Detect, Predict & Report Separation Violation (using the attributes Type, Boundary, Altitude Limits, Schedule, and Separation Minima) determines that the location of an aircraft is within a separation minima distance of the airspace, and the aircraft has not been granted an exemption to enter the airspace (according to the attribute Exemptions). Flight Manager 1 determines that the Aircraft object is part of a specific Flight object which has an instance connection to a specific Flight Manager object (called "Flight Manager 2"). Flight Manager 1 then sends a message to Flight Manager 2 requesting the service Separate IFR Flights. In performing that service, Flight Manager 2 uses the service Generate & Deliver Clearance to request (by message to the pilot of the Flight object) that the Flight adhere to its existing clearance (which did not permit flying closer to the airspace than the separation minima). (Note: If the clearance had permitted flying close to or through the protected airspace volume, the permission would have been reflected in the Exemptions.
attribute of the Protected Airspace Volume object. Alternatively, Flight Manager 1 could have accessed the Flight Plan object or the En Route Clearance object of each aircraft of interest.)

As in the previous view, Flight Manager 1 and Flight Manager 2 might be the same object. When a Protected Airspace Volume object is physically contained within more than one Airspace Volume object, the Flight Manager objects might be different objects.
Figure 3-13. Aircraft-to-Airspace Conflict Detection and Resolution Event
In this view, the triggering event was related to the activity of an Aircraft object. An alternative airspace conflict view could be constructed to illustrate the result of a change to a Protected Airspace Volume object. Here, it would be necessary to determine the instance connections between the Protected Airspace Volume object and all appropriate Flight objects (creating instance connections if the Protected Airspace Volume object is just being created) in order to predict violations of separation minima as well as to detect violations of separation minima (as above).

**Flow Instruction Problem Detection and Resolution Event.** Figure 3-14 is a view of a flow instruction problem event, in this case a predicted noncompliance with a flow instruction recorded in the Usage Restrictions attribute of a Route object. At the beginning of the event, the Pilot object wants to change the Route attribute of the Flight Plan object in such a way that the flight will use the Route object. The Flight object is connected by an instance connection to a Flight Manager object (called "Flight Manager 2"). The Pilot object sends a message to Flight Manager 2 requesting approval of the change in route. Flight Manager 2, in turn, determines that the Route object is connected by instance connection with another Flight Manager object, called "Flight Manager 1." Flight Manager 2 sends a message to Flight Manager 1 requesting permission for the Flight object to use the Route object. Flight Manager 1 performs its service Detect, Predict & Report Restriction Violation. Comparing the Route attribute of the Flight Plan object with the Usage Restrictions attribute of the Route object, Flight Manager 1 determines that a noncompliance with a flow instruction would exist if the route change were approved, and so informs Flight Manager 2. Flight Manager 2 then performs its service Ensure Restriction Compliance to determine if a feasible alternative exists, and finding none (in this example), returns a response to the Pilot object denying the request.

For simplicity in this view, only one Route object is shown as the new resource to be used. Most likely, a proposed route change would involve several resources, and Flight Manager 2 would need to find out the impact of the change on each before responding to the Flight object.

Also for simplicity, the view shows Flight Manager 1 only checking for restriction violations; in fact, separation violations would also be checked for by the Flight Manager service Detect, Predict & Report Separation Violation.

**3.2.6.4 Airspace/Ground Resource Saturation Detection and Resolution Event**

Figure 3-15 illustrates an event in which a ground-type resource (an Airport Movement Area object) is predicted to become saturated at a future time. The Airport Movement Area object is connected to a Ground Traffic Manager object by an instance connection. The Ground Traffic Manager, in performing its service Detect, Predict & Report Saturation, compares the value of the Airport Movement Area object's attribute Demand with the values of its attributes Saturation Threshold and Capacity and finds that demand for a particular future time period is no longer less than the saturation threshold for that same time period. The Ground Traffic Manager object then performs its service Resolve Ground Saturation Problem. In performing that service, the Ground Traffic Manager object has several alternatives available; for this example, the Ground Traffic Manager object decides to request the Air Traffic Manager object associated, by instance connection, with a particular Fix/Waypoint object to perform its service Determine Airspace Capacity by placing restrictions for a corresponding time period on the usage of the Fix/Waypoint object by Flight
objects. The Air Traffic Manager object approves the request and in turn requests the Fix object to update the value of its Usage Restrictions attribute (i.e., access the attribute to set a new value).

While the event pictured stops here, there will be a ripple effect as Flight Manager objects now perform their services Detect, Predict & Report Restriction Violation and Ensure Restriction
Figure 3-14. Flow Instruction Problem Detection and Resolution Event
Figure 3-15. Airspace/Ground Resource Saturation Detection and Resolution Event
Compliance, leading to changes to Flight Plan objects and Clearance objects, leading to changes in demand for airspace/ground resource-type objects, leading to reassessments of saturation, and so on.

3.2.6.5 Clearance Delivery (Communications) Event

Figure 3-16 illustrates the use of communications in delivering a clearance. In fact, for all events involving clearance delivery, there is a communications event. However, in order to keep the previous event views simple, the communications part of the event has not been shown.

During a flight, it is not physically possible for a flight manager to talk directly with a pilot; some communications media must be available. Specifically, the Flight Manager object "talks with" a Communications System object, a Pilot object talks with a Vehicle Communications System object, and the Communications System object and the Vehicle Communications System object talk with each other. In fact, the instance connection between a Flight Manager object and a Flight object is defined in terms of the communications frequency of the interfacing communications systems. (The instance connection exists to reflect the control responsibilities of the Flight Manager object with respect to the Flight object, but can be defined in terms of the communications frequency because a change in communications frequency and a handoff of control responsibilities normally occur at the same time.)

In this event, a Flight Manager object wishes to deliver a clearance amendment to a Pilot object. The Pilot object is part of an Aircraft object which is part of a Flight object. The other part of interest of the same Aircraft object is a Vehicle Communications System object. The Flight Manager object is connected with the Flight object by an instance connection. The Flight Manager object knows what communications frequency to use, as well as the ATC communications systems, because the frequency is the basis of the instance connection. In this view, the ATC communications system is represented by an Air/Ground Communications System object. The Flight Manager object sends a message to that Air/Ground Communications System object requesting that object to perform its service Transmit Message in sending the clearance amendment to the Vehicle Communications System object. The Air/Ground Communications System object then sends the message as requested, the Vehicle Communications System object makes the message contents available to the Pilot object, and (assuming the clearance amendment is acceptable) the Pilot object requests the Vehicle Communications System object to return an acknowledgement, the Flight Plan object to amend itself, and the En Route Clearance object to amend itself. The acknowledgement to the clearance is returned by the Vehicle Communications System object via the Air/Ground Communications System object to the Flight Manager object, using the return paths implicit in the original messages.

3.2.6.6 NAVAID Use (Navigation) Event

Figure 3-17 illustrates the use of a NAVAID in the navigation of an aircraft. The NAVAID is considered to be an object of the Navigation System class. The view would also be valid were the NAVAID an object of the Landing Navigation System class.

For this example, it is assumed that the NAVAID is a VOR/DME station and that the aircraft is equipped with a VOR receiver and DME unit. To illustrate the functioning of the VOR receiver as
distinguished from the DME unit, an object was created for each as an instantiation of the Vehicle Navigation System class, called (for convenience) "Vehicle Navigation System 1" and "Vehicle Navigation System 2," respectively. Although not done for this view, it would have been possible to use two Navigation System objects to separate out the VOR and DME parts of the navigation system.
Figure 3-16. Clearance Delivery (Communications) Event
Figure 3-17. NAVAID Use (Navigation) Event

3-50
This view shows two independent events in the navigation of an aircraft: the acquisition of azimuth information (the radial of a NAVAID) and the acquisition of distance information (the slant-range distance from the NAVAID).

The Navigation System object sends the message Offer Azimuth Information to represent the signals transmitted by the VOR transmitter. This message is sent not just to the Aircraft object in particular (as shown here), but is available for any appropriate receiver. Upon receiving the message, the service Accept Navidational Guidance Data of Vehicle Navigation System 1 (the VOR receiver) is performed.

Independently of the VOR receiver, Vehicle Navigation System 2 (the DME Unit) sends the message Request Distance Information in performing its service Interrogate Navigational Aid. The Navigation System object returns information from which the DME unit can calculate the distance to the VOR/DME station.

This view also illustrates some subjective decisions made in constructing the ATC model. The VOR/DME station is considered an object of the Navigation System class. For this example, a Navigation System object could easily be shown in a whole-part relationship with two parts named, for example, VOR System object and DME System object. Similarly, a Vehicle Navigation System object could easily have appeared in this example as a whole-part relationship with two parts named, say, VOR Receiver and DME Unit. However, in each case, the decision of what parts to add depends upon having more knowledge than is available from a generic class, and would necessitate a proliferation of specialization classes beneath the more generic Navigation System class and Vehicle Navigation System class. Adding so much detail did not seem appropriate at this time, but could be necessary in other situations (e.g., during the design phase). Instead of creating many specialization classes, the existing classes were made flexible for use as in this view.

This view is unlike the previous views because it does not include a manager-type object. As discussed in section 3.2.3, the management aspects of a navigation system resource are left within the resource itself in the ATC model and hence do not appear as services for manager-type objects within the Manager subject.

### 3.2.6.7 Tracking (Surveillance) Event

Figure 3-18 illustrates the use of an ATC surveillance system in locating an aircraft. As in the previous example, the view does not involve a Manager object. This view would also be valid for any specialization of the Surveillance System class.

It is assumed for this view that the Surveillance System object can detect, identify, track, and predict aircraft targets laterally and longitudinally, as well as vertically when the aircraft has an altitude-encoding transponder. This view also assumes that the vehicle surveillance system is an altitude-encoding transponder (i.e., a Transponder object with the inherited service Report Altitude is appropriate).
The Air Traffic object in this view has, as its part objects, each Flight that meets the selection criteria that its Aircraft object part has its Location attribute value within the Area of Coverage attribute value of the Surveillance System object.

The Surveillance System object performs the service Locate, Identify & Report Flight. The Surveillance System object locates an aircraft by transmitting a radar pulse and interpreting the returned pulse (the echo), identifies the aircraft by comparing the computed location of the aircraft with the Location attribute of an existing Aircraft object, acquires the altitude of the aircraft by
Figure 3-18. Tracking (Surveillance) Event
sending the message Request Altitude Report to the Transponder object, and then sends a message to
the Aircraft object requesting it to update its Location attribute. Note that the value of the Location
attribute of an Aircraft object is the value determined by the ATC system, which is not necessarily
the real location of the aircraft.

The Transmit & Receive Radar Pulse message is shown as a message from the Surveillance System
object to itself. An alternative way to show this is to introduce another part object for the Aircraft
object to represent the physical hull of an aircraft, so that the radar pulse could be sent to the hull,
and the hull could return the echo. This alternative was not taken because the response to the pulse
seemed to be more a passive response than the active performance of a service.

A surveillance system tracks aircraft and manages the information provided (e.g., identifies the
aircraft and reports their positions). A Surveillance System object subsumes the surveillance sensors
(radars) and the automation that processes that data. It would be possible to define a new whole-part
structure for each object of the Surveillance System class by adding the parts "Sensor" and
"Manager." For this view, two Sensor objects would be needed, one for the primary radar and one
for the secondary radar. However, only the Surveillance System object is used in this view.

### 3.2.6.8 Weather Events

As discussed in section 3.2.2, aviation weather is represented in the ATC model only as reports and
report managers. This section describes the following two weather events:

- A VFR flight receives a preflight briefing from a Flight Service Station.
- A weather area becomes significant enough that it is defined to be a protected airspace
  volume.

**Preflight Briefing (Weather) Event.** Figure 3-19 illustrates a preflight briefing event. In this
view, a pilot for a VFR flight needs a preflight briefing. A Pilot object (part of an Aircraft object
which is part of a Flight object in the predeparture state) sends a message to a Flight Manager object
requesting a standard preflight briefing. Because the Flight object is in the predeparture state, and
because no Flight Plan object with Departure Point attribute is available, there is no obvious recipient
Flight Manager object. Thus the Flight Manager object selected to receive the message must be
identified by the Pilot object itself. The Flight Manager object provides the preflight briefing as part
of the service Assist in Flight Planning. Part of the standard preflight briefing is a summary of
forecast en route and destination weather. To get this information, the Flight Manager object
accesses appropriate Area Forecasts and the Terminal Forecast for the destination airport. The view
shows messages to Wind/Weather Product objects for Area Forecasts as part of the Aviation Weather
System object with the Area of Coverage of the National Aviation Weather Advisory Unit
(NAWAU) and a message to a Wind/Weather Product object for a Terminal Forecast as part of an
Aviation Weather System with the Area of Coverage of a particular Weather Service Forecast Office
(WSFO). This information is put together with other information and returned to the Pilot object.

Of course, much more information than that contained in Area Forecasts and a Terminal Forecast is
made available in a preflight briefing, but only these forecasts are shown in this view for simplicity.
This view also illustrates that the Flight Manager class represents functionality rather than existing positions, as discussed in section 3.2.3. In usual terminology, the functionality described above is that performed by an Automated Flight Service Station Specialist in the Preflight Position.
Figure 3-19. Preflight Briefing (Weather) Event
Weather Area Event. Figure 3-20 illustrates an event in which a weather area is determined to be significant enough that active measures should be taken to prevent aircraft from entering it. In this view, an Aviation Weather System Manager object, in performing the service Monitor, Forecast & Report Weather, predicts that thunderstorm activity in the object's area of responsibility will increase in size and intensity. The Aviation Weather System Manager object sends a message to the Air Traffic Manager object with area of responsibility containing the thunderstorm to inform that object of the prediction. The Air Traffic Manager object, in performing its service Assist in Weather Avoidance, decides that a Protected Airspace Volume object should be created to enclose the thunderstorm, and sends a message to the Protected Airspace Volume class requesting that this be done. The Protected Airspace Volume class creates such an object.

Although not shown in this view, once the Protected Airspace Volume object is created, the alternative view described in the Aircraft-to-Airspace Conflict Detection and Resolution Event of section 3.2.6.3 (above) applies.

3.2.7 Model Specification

The model specification for the ATC model is contained in appendix A.
Aviation Weather System Manager

Area of Responsibility

Monitor, Forecast & Report Weather

Provide Weather Forecast

Protected Airspace Volume

(attributes)

request creation of Protected Airspace Volume object

Air Traffic Manager

Area of Responsibility

Assist in Weather Avoidance

Figure 3-20. Weather Area Event
SECTION 4
PROBLEM ANALYSIS

This section provides the results of the problem analysis subtask, including descriptions of an experiment model and two legacy software models and a comparison of experiment needs (represented by the experiment model) with legacy software capabilities (represented by legacy software models). The models are based upon the ATC model and upon the written and verbal knowledge of the experiment and software authors. The specifications for the models are contained in appendices B, C, and D.

The methodology for constructing and comparing the models is described in section 2.3.

4.1 THE EXPERIMENT

In March 1992, I-Lab will be ready to perform ATC experiments of interest to the FAA and to The MITRE Corporation, as well as to establish an environment for operational personnel to provide input to new or proposed automation for future enhancements to the ATC system. The first I-Lab experiment has been proposed to the FAA, and is a result of many iterations in the areas of problem definition, operational concepts, and experimental design. The experiment, as it was defined in December 1990, was expressed in terms of the ATC model.

The proposed experiment focuses on the interactions between AERA 2, the terminal area with Terminal Advanced Automation System (TAAS) capabilities, and metering with Advanced Automation System (AAS) capabilities. During periods of heavy ground demand, the metering program of the AAS will sequence and schedule the traffic and will issue flow restrictions to flights still in en route airspace. The flow restrictions will specify the time each flight should cross a meter fix at the boundary of the terminal airspace. The en route controllers must issue clearances to ensure that flights projected to arrive at the meter fix will arrive when scheduled. The controllers are assisted in this task by AERA 2 software.

Successful interaction among metering, en route, and terminal automation would be reflected as a smooth flow of traffic into the terminal area that fully utilizes available arrival capacity without overburdening the terminal area arrival controller. Successful interaction also implies that traffic in the en route area is managed in such a way that it remains controllable in en route airspace (i.e., excessive tactical maneuvering is not required to maintain separation or to meet metering restrictions).

One of the factors likely to affect the interaction among the automation programs is the timing of flow restrictions. The experiment will vary the delay absorption interval, which is defined to be the interval of time between when AERA 2 receives a meter fix restriction for an aircraft and the time that the aircraft would arrive at the meter fix if the aircraft were not delayed. Proposed values for the delay absorption interval are 90 minutes, 60 minutes, and 30 minutes.
If the delay absorption interval is too long, it may cause imprecise delivery to the meter fixes so that controllability in the terminal area is not acceptable (e.g., many aircraft must be maneuvered to absorb delay in the relatively small terminal area). Shortening the delay absorption interval so that capacity utilization and terminal area controllability is accomplished may cause en route difficulties. If this is the case, an en route delay allocation mechanism may be needed, such as additional control points.

At the time the ATC model was used to describe the experiment, the New York Metroplex was considered the most probable site to study these interactions. The airport of interest would be La Guardia, where the arrival acceptance rate depends on several factors (e.g. runways in use, weather conditions). Arrival demand would be allowed to vary during the experiment. The terminal area would be represented by the New York Terminal Radar Approach Control (TRACON), and all meter fixes would be located on the boundary between en route airspace and the TRACON.

Testbed requirements were identified in the experimental design. These requirements specified categories of airspace and ground resources and how they would be modelled in the laboratory. The airspace requirements included specific automation capabilities for terminal and en route ATC and for metering.

4.1.1 Analysis of the Experiment vs. the ATC Model

Running an I-Lab experiment is similar to running any other type of experiment: it is a process which involves setting initial conditions, executing a set of procedures, and analyzing the results. Initial conditions are set by assigning values to the attributes at the start of the experiment. Measurements of ATC system performance are derived by analyzing the variation of certain attributes throughout the experiment.

Components (e.g., objects, attributes, services) necessary to run the experiment were identified by performing an object-oriented problem analysis of the proposed experiment definition, resulting in a model of the experiment (hereafter called the "Experiment model"). The process of abstracting high level classes and determining their hierarchical structure was not necessary because the ATC model could be used as a reference model. It was necessary to abstract the data, control, and processes required by the experiment, thus creating lower-level domain-unique classes and structures.

By using a common reference model (the ATC model) for both the experiment analysis and the legacy software analysis, it was possible to make direct comparisons between the components that must exist in the I-Lab to run the proposed experiment and the components that can be provided by the legacy software. The process of evaluating and comparing the Experiment model with legacy software models is described in section 2.3, Model Usage.

4.1.2 The Experiment Model

The Experiment model contains all three subjects identified by the ATC model: User, Resource, and Manager. Each subject contains a structure with a base class of the same name as the subject, and with the Manager subject being contained within the User subject. Figure 4-1 shows the three subjects of the Experiment model; only the first and second levels of classes are shown.
Lower-level domain-unique components were abstracted from the experiment definition proposed in December 1990. The interactions of interest defined in the proposed experiment narrowed the experiment problem domain from the ATC domain. The class and object types identified in section 2.2.1 (e.g., other systems, devices, things or events remembered, roles played, operational procedures, sites, and organizational units) were used as a guideline in identifying classes and objects.

For the experiment, sources and sinks of flights were of importance in determining airspace boundaries and airports. The three proposed values of the delay absorption interval also influenced the shape and size of airspace volumes. An experiment scenario traffic pattern was also identified within these airspace boundaries.
Figure 4-1. Experiment Model: Subjects
Attributes were determined both by an analysis of the metrics important to the experiment and by the need for descriptive attributes for the domain-unique classes and objects.

In defining domain-unique specializations in the Manager class structure, services performed by automation were not distinguished from services accomplished by human intervention. In particular, the services provided by a regional traffic manager were not distinguished from services provided by metering algorithms, and services provided by an en route controller were not distinguished from services provided by AERA 2 software.

Although the experiment focuses on the AERA 2 functions which interact with metering and terminal area functions, the Experiment model does not include a complete representation of the AERA 2 model. A complete AERA 2 model is included in section 4.2.2.

The following subsections explain the Experiment model in detail. The notation used, whereby classes are shaded to indicate components borrowed from the ATC model, is the notation defined in section 2.3, Model Usage.

4.1.2.1. The User Subject

The User subject, borrowed from the ATC model, contains the same two structures as the ATC model: the Manager class structure and the Traffic class structure. Since the Manager class also belongs to the Manager subject, the Manager class structure will be discussed separately in section 4.1.2.3. The remainder of this section discusses the Traffic class structure.

The Traffic Class Structure. The Traffic class of the experiment, shown in part 1 of figure 4-2, is only concerned with the Air Traffic class. The experiment is not concerned with the movement of vehicles on the ground. Only after traffic originating on the ground has departed and must be sequenced into the terminal area traffic is it of concern to the experiment. However, it is necessary to identify the ground movement areas (discussed in section 4.1.2.2, The Resource Subject) since they are a source of air traffic.

A domain-unique specialization of the Air Traffic class, the Experiment Scenario Traffic Pattern class, was identified in the experiment by the definition of a scenario traffic pattern needed, both in en route (AREA 2) airspace and the terminal area. Specific traffic patterns define the volume and complexity of traffic, and influence resource load and demand throughout the experiment.

Individual flights are also of interest to the experiment since each en route flight will be assigned a meter fix time and can be maneuvered individually while en route, and each flight in the terminal area will also be maneuvered on an individual basis. Relevant parts of a Flight object are an Aircraft object, a Flight Plan object, and a Flight Clearance object.

The relevant parts of an Aircraft object are a Pilot object, a Vehicle Communications System object, and a Vehicle Navigation System object. Pilot object services in the terminal area are performed by automation and by a staffed simulated pilot ("sim-pilot") position, while Pilot object services in en route airspace are automated. The Vehicle Communications System class is necessary in terminal airspace so that the terminal area controllers can communicate with the sim-pilots, but Vehicle...
Communications System object services are automated in en route airspace. The Vehicle Navigation System class is necessary so that flights can fly either a Microwave Landing System (MLS) or Instrument Landing System (ILS) approach to La Guardia airport, as well as fly point-to-point.

The domain-unique Flight Plan object attributes of interest to the experiment are the Expected Meter Fix Time, Expected Airport Arrival Time, and the Flow Restriction Time. A complete representation
Figure 4-2. Experiment Model: User Subject (Traffic) (Part 1)
Figure 4-2. Experiment Model: User Subject (Traffic) (Part 2)
of the Flight Plan class and object is identified in the AERA 2 model. All parts of a Flight Clearance object are of interest to the experiment, i.e., an En Route Clearance object, a Departure Clearance object, and an Arrival Clearance object.

Classes Supporting Problem Detection. The AERA 2 domain-unique classes of interest to the experiment pertain to the way in which AERA 2 detects and handles problems. Trajectory objects are used in predicting problems. The problems of interest to the experiment, shown in part 2 of figure 4--2, are aircraft-to-aircraft conflicts, aircraft-to-airspace conflicts, and flow instruction non-compliance problems. (The attributes and services of these objects are explained in the AERA 2 model.) For the experiment, the "best" resolution to a problem (as determined by AERA 2 software), called the highest-ranked resolution (HRR), will automatically be chosen when a problem occurs.

4.1.2.2. The Resource Subject

The Resource class specializations of interest to the experiment are the Airspace/Ground Resource class, the Surveillance Resource class, the Communications Resource class, and the Navigation Resource class. The Aviation Weather Resource class is not involved in the experiment.

The Airspace/Ground Resource Class. Both parts of the Airspace/Ground Resource object are necessary; these parts are the Controlled Airspace object and Airport Movement Area objects. The Airport Movement Area class, shown in part 1 of figure 4-3, is referred to in the experiment as "airport." Two domain-unique, specialized classes of the Airport Movement Area class are the La Guardia Airport and Washington Area Airport classes. Since the site of the terminal area in the experiment is the New York TRACON, the attributes of the La Guardia Airport object that are of interest to the experiment are Arrival Capacity, Departure Capacity, Runway Configuration, and Expected Departure Rate. Washington Area Airport objects are identified since they will be sources and sinks of flights in the en route airspace. The attributes of interest of a Washington Area Airport object are Runway Configuration and Expected Departure Rate.

The Controlled Airspace Class. Shown in part 2 of figure 4-3, the parts of the Controlled Airspace object needed for the experiment are the En Route Controlled Airspace object and a Terminal Controlled Airspace object. One important aspect of the experiment is the location of the meter fixes which are on the boundary between the en route and terminal areas. For this reason, the attribute Boundary is defined as a domain-unique attribute for both airspace-type objects.

The Terminal Controlled Airspace Class. The parts of the Terminal Controlled Airspace object (a Terminal Route object and a Terminal Airspace Volume object) are of interest. A specialization of the Terminal Airspace Volume class is the New York TRACON class. The Terminal Route class specializations identify the ILS Route class and the MLS Route class.

The En Route Controlled Airspace Class. The En Route Controlled Airspace object is composed of Route objects and an Airspace Volume object, all required for the experiment. The airspace volume which of interest is the 90-minute flying-time wedge which extends southeast from the New York TRACON toward and past the Washington area airports. Of interest from a Route object is the part Fix/Waypoint object. Instances of the Fix/Waypoint class will be used by all flights in the en route airspace (whether metered or not). A specialization of the Fix/Waypoint class, the
Meter Fix class, is identified for those fixes which are metered. An instance (object) of a Flow Instruction class would be associated (by instance connection) with an instance of the Meter Fix class.

**The Other Resource Classes.** The Surveillance Resource structure, Communications Resource structure, and the Navigation Resource structure are shown in figure 4-4.
Figure 4-3. Experiment Model: Resource Subject (Airspace/Ground Resource) (Part 1)
Figure 4-3. Experiment Model: Resource Subject (Airspace/Ground Resource) (Part 2)
Figure 4-4. Experiment Model: Resource Subject (Other Resources)
The Surveillance Resource Class. The surveillance resource of interest is the Airspace Surveillance System class, a specialization of the Surveillance System class. Two specializations of the Airspace Surveillance System class are identified in the experiment. One specialization, the ARTS-Like Terminal Surveillance System class, is identified for use in the terminal area. An ARTS-Like Terminal Surveillance System object is assumed to include the primary radar antenna, the radar data acquisition system, the ATC radar beacon system (secondary radar surveillance) antenna, the beacon data acquisition system, and the data processing subsystem (commonly called an Automated Radar Terminal System [ARTS] by itself). At a design level, the object is also assumed to include a data entry and display subsystem and a continuous data recording subsystem. An ARTS-Like Terminal Surveillance System object is required to provide the services for a user interface and to provide TAAS capabilities. The experiment is similarly provided with radar data processing (RDP) services for en route airspace by the other specialization of the Airspace Surveillance System class, the RDP-Like En Route Surveillance System class. An RDP-Like En Route Surveillance System object is assumed to include primary and secondary radar antennas, a beacon data acquisition system, an RDP subsystem, a data entry and display subsystem, and a continuous recording subsystem.

The Communications Resource Class. The communications resource of interest is the Air/Ground Communications System class, a specialization of the Communications System class. Two specializations of the Air/Ground Communications System class identified in the experiment are the Data Link-Like Communications System class and the Voice Radio-Like Communications System class.

The Navigation Resource Class. The navigation resource of interest is the Landing Navigation System class, a specialization of the Navigation System class. The Landing Navigation System class is needed to provide the resource counterparts to the navigational equipment which exists in the aircraft. The two specializations of the Landing Navigation System class identified by the experiment are the ILS-Like Landing Navigation System class and the MLS-Like Landing Navigation System class. For the experiment, these systems and equipment will be used for flying approaches to La Guardia airport.

4.1.2.3 The Manager Subject

The Manager subject is shown in figure 4-5. As mentioned in section 3.2.3, a close coupling exists between the Manager class structure and the Traffic class structure. Those managers of interest to the experiment are specializations of the Traffic Manager class and of the Flight Manager class. The Traffic Manager class specialization of interest is the Air Traffic Manager class since only air traffic is of interest. The domain-unique specialization identified is the Regional Air Traffic Manager which is necessary because metering is a regional traffic management function. The services of a Regional Air Traffic Manager object include services which would be performed by the metering program and those performed by a human traffic manager. Those services performed by the metering program are Impose Flow Instruction, Create Metering Flow Restriction, Sequence/Resequence Flight, and Adjust Flight Schedule. The service provided by the human traffic manager is Monitor.

Three Flight Manager class specializations were identified: the En Route Controller class, the Terminal Controller class, and the Tower Controller class. The En Route Controller class
encapsulates services performed by AERA 2 software and those performed by a human en route controller (automated for the experiment). Those services performed by AERA 2 are Identify Conflicts, Propose Resolutions, and Compute Delay-Absorbing Maneuvers. Those performed by the automated human controller are Issue Delay-Absorbing Maneuvers and Handoff Aircraft to Terminal Controller.
Figure 4-5. Experiment Model: Manager Subject
The New York TRACON has a terminal controller to perform those services necessary to control traffic in the terminal area. Services identified by the experiment for a Terminal Controller object are Final Arrival Sequencing and Spacing, Coordinate Arrivals With Departures, and Accept/Initiate Handoffs. La Guardia Airport has a tower controller to perform those functions which will affect the coordination of arrivals and departures in the terminal area. Each Washington area airport has a tower controller to respond to flow instructions that require ground delay of Washington area departures.

4.1.2.4 Event View: Handoff of a Flight from En Route Controller to Terminal Controller

As discussed in section 3.2.6.2, a change in flight state is associated with a change in manager rather than with a change in resource usage. The change in flight state of interest to the experiment is the state change from en route to arrival. This state change occurs when the flight is handed off from the en route controller to the terminal controller.

For the experiment, the en route controller services will be automated and the terminal controller position (actually the arrival controller position) will be staffed. The handoff event will occur without a means of coordination between the two controllers. The en route controller will determine when a flight should be handed off, and the terminal controller will accept it at that time. This will be true whether or not AERA 2 has successfully maneuvered the flight to meet the meter fix time.

Figure 4-6 is a view of the event causing this state change, using the classes and objects identified in the experiment definition. At the beginning of the event, an En Route Controller object exists with instance connections to an En Route Airspace Volume object for which the controller is responsible and to a Flight object under the controller's control. All three parts of the Flight object, the Aircraft object, the Flight Plan object, and the Flight Clearance object, are necessary. The Flight Clearance object has two parts, the Arrival Clearance object and the En Route Clearance object. An instance connection exists between the Aircraft object and the same En Route Airspace Volume object. This instance connection indicates that the Aircraft object is being taken into account in the Load attribute of the En Route Airspace Volume object and is observed by the en route airspace surveillance systems. An instance connection also exists between the Flight object and the Terminal Airspace Volume object and between the Flight object and the En Route Airspace Volume object indicating that the Flight is being taken into account in the Demand attribute of each airspace-type object.

The En Route Controller object monitors the Location attribute of the Aircraft object and the Route and Altitude Profile of the Flight Plan object with respect to the Boundary and Altitude Limit attributes of the En Route Airspace Volume object. The controller determines when the flight is about to leave the en route airspace volume and enter the terminal airspace volume. At this point, the En Route Controller object sends a message to the Terminal Controller object requesting acceptance of control responsibility for the flight, and the Terminal Controller object automatically accepts. At this time, the instance connection between the Flight object and the En Route Controller object can be deleted and an instance connection between the Flight object and the Terminal Controller can be created. The state change of the Flight object is now complete.

Although not shown in this view, the following will also occur:
• When the En Route Controller object tells the Pilot object of the Aircraft object to change radio frequency to the frequency for the Terminal Controller object, the Pilot object will comply and will also tell the En Route Clearance class to delete the En Route Clearance object (see section 3.2.6.2 for a similar event).
Figure 4-6. Experiment Model View: Handoff Of A Flight From En Route Controller To Terminal Controller
• When a Surveillance System object detects that the aircraft object is no longer in the en route airspace volume, the Surveillance System object will request the Aircraft class to delete the instance connection (based on load) between the Aircraft object and the En Route Airspace Volume object and will request the Flight class to delete the instance connection (based on demand) between the Flight object and the En Route Airspace Volume object.

• When a surveillance system detects that the aircraft is in the terminal airspace volume, the Surveillance System object will request the Aircraft class to add an instance connection (based on load) between the Aircraft object and the Terminal Airspace Volume object.

4.2 MAPPING THE LEGACY SOFTWARE ONTO THE ATC MODEL

The first milestone of the I-Lab was met six months into the project and was called the Six Month Illustration of Technical Feasibility, or 6-MITF. It was a proof of concept that demonstrated the ability to integrate six functionally different ATC software components, which were developed independently and without the requirement of interoperability, to form a complete ATC system. These software components, referred to as "legacy" software components, were existing simulations, prototypes, and operational prototypes developed over the past five years in support of various MITRE projects. Mapping the legacy software components used for the 6-MITF onto the ATC model was intended to explore the possibilities of using the legacy software to perform the proposed experiment. The process by which the experiment capabilities were compared to the capabilities offered by legacy software is discussed in section 2.3, Model Usage.

Two pieces of legacy software, the Terminal Area Simulation Facility (TASF) and the Automated En Route Air Traffic Control Phase 2 (AERA 2) Computer-Human Interface (CHI) Prototype were defined in terms of the ATC model and are discussed in this paper.

4.2.1 TASF

TASF was built as a simulation platform to support the development and evaluation of automation in the terminal area. TASF simulates the necessary environment so that various ATC automation aids can be developed and evaluated using "real life" scenarios before being tested in the field. For the purposes of accurately developing and assessing the ATC automation aids, TASF attempts to separate the simulated environment from the automation. The simulated models used in the six-month configuration were an ARTS III computer, a wind model, and an aircraft true trajectory modeler; the automation aid used was the ghosting aid. The ghosting, or "imaging," aid generates and projects false targets, called "ghosts," based on real targets and displays them on ARTS III displays in TRACON facilities.

The true trajectory modeler, which calculates the point position of aircraft in the terminal area, accepts many different forms of input: a route, route segment, maneuver, or point position. At any of these levels, the input can be retrieved from files; at the route segment and maneuver level, the input can be made by a simulated pilot ("sim-pilot"). TASF does not use filed flight plan information, such as proposed route data, to predict the future position of the aircraft. The aircraft target displayed by the simulated ARTS III computer was the actual calculated position (taking into
consideration output from the wind model). For the 6-MITF, the simulated ARTS III computer supported a bright radar indicator tower equipment (BRITE) display for the La Guardia tower and three displays for the New York TRACON (North arrivals, South arrivals, and departures/final approach).

Currently in the terminal area, there are no standard or accepted airways or jetways (STARs and SIDs are precursors), but the standard procedure is for a pilot to fly the current course until vectored by a controller. However, for the six-month illustration, TASF had the "concept" of two defined routes. One route was defined by the ILS approach procedure, and the other was defined by a curved MLS approach route. A flight could be maneuvered by a sim-pilot, or could capture and fly the ILS route, or could capture and fly the MLS route, or could join the MLS route.

The configuration of TASF used for the six-month illustration provided the laboratory the capability to simulate, display, and maneuver aircraft through the New York terminal area. Operations centered around arrivals into La Guardia Airport on runway 13 during inclement weather conditions. Arrivals on the La Guardia ILS approach route interfere with arrivals bound for nearby Teterboro airport. To create gaps in the ILS route which would allow arrivals into Teterboro, a curved MLS approach route was used.

For the 6-MITF, the La Guardia arrivals on both approach routes had to be merged by the final approach controller for landing. The controller was aided in the task of sequencing and spacing the aircraft by the ghosting aid, which projected MLS aircraft onto the ILS route. The final approach controller issued clearance directives to the sim-pilots using simulated radio communications, and the sim-pilot entered the individual aircraft maneuvers into a VT 220 computer terminal which was input to the true trajectory modeler of TASF.

4.2.1.1 Analysis of TASF vs. the ATC Model

The configuration of TASF used for the six-month milestone was described using the ATC model. TASF may not be the terminal area component used in March 1992, but the resulting analysis was intended to identify existing terminal area components that might correspond to terminal area components required by the experiment. TASF was not designed as an object-oriented system, and the model (hereafter called the "TASF model") merely describes the TASF software component in an object-oriented manner for the purposes of comparing the TASF capabilities with experiment requirements.

Using the ATC model as a reference model was accomplished the same way as for the experiment: high-level classes were provided, and it was necessary to determine the domain-unique subset of these classes and to define the lower-level domain-unique classes and structures. The data, control, and processes performed by this configuration of TASF are abstracted in these domain-unique classes and structures.

TASF identifies several classes which are not strictly within the terminal area domain. An example of this in the Manager subject is the En Route Controller class specialization which is also, more appropriately, found in the AERA 2 model. Other examples are found in the Resource subject where, in addition to identifying the Terminal Controlled Airspace class, the En Route Controlled

© 2000 The MITRE Corporation. ALL RIGHTS RESERVED.
Airspace class is identified, as well as the Airport Movement Area class. These classes provide a means to interface with neighboring (en route and ground) resources and managers when TASF is operating in a stand-alone mode, and were not used when TASF was integrated in the 6-MITF.

### 4.2.1.2 The TASF Model

The TASF model contains all three subjects identified by the ATC model: User, Resource, and Manager. Each subject contains a structure with a base class of the same name as the subject, and with the Manager subject being contained within the User subject. Figure 4-7 shows the three subjects of the Experiment model; only the first and second level of classes are shown.

Lower-level domain-unique components were found by reading available documentation, talking with domain-knowledgeable people, and examining sections of the source code. Documentation and discussions gave a good overall understanding of how TASF modelled the terminal area. Data
Figure 4-7. TASF Model: Subjects
structures identified in the source code were good sources for classes and their attributes, and subroutines were sources for services. When using all sources of information, the categories identified in section 2.2.1 for identifying classes and objects were kept in mind.

Sources and sinks of flights in the terminal area include the en route airspace and the airport movement areas. Interfaces to these sources and sinks are abstracted in the services of the controller objects, which include accepting and delivering aircraft handoffs and flight data.

The following subsections explain the TASF model in detail. The same notation used to map the experiment onto the ATC model was used to map TASF; the notation is defined in section 2.2.

**The User Subject.** The TASF User subject also consists of the same two structures as the ATC model, the Traffic class which is discussed immediately following and the Manager class which is discussed later. The Traffic class structure of the TASF model is shown in figure 4-8.

Traffic in the terminal area is all air traffic; therefore, the TASF model is only concerned with the Air Traffic class from the ATC model. TASF is concerned with flights on an individual basis as well. Flight objects and all the parts identified in the ATC model are relevant to the TASF model, including the Flight Clearance object, the Flight Plan object, and the Aircraft object. Domain-unique attributes of Flight are an attribute which identifies the airline and an attribute which identifies the flight number.

Flight Clearance objects and the parts Departure Clearance object and Arrival Clearance object are needed, without identifying domain-unique components. Although the departure clearance is issued on the ground, the cleared route extends into the terminal area and must be known by the departure controller in the terminal area. The approach clearance is a dynamic clearance for an aircraft that is maneuvered by the controller and the sim-pilot and is a static clearance for an aircraft following a specified route.

Flight information required by TASF is a subset of what is contained in an actual flight plan. As discussed earlier, TASF does not use flight plan data for route prediction; however, TASF does need arrival and departure information for those flights that originate or land within the terminal area. The pertinent information is found in the following attributes: Flight Type, Departure Airport, Arrival Airport, Runway, and Departure Fix.

An Aircraft object has attributes defined in the aircraft data structure which identify the aircraft limits, position, and velocity. Position attributes are input to the simulated ARTS III computer and displayed. Aircraft services were abstracted from the trajectory modeler and determine the true position of the aircraft. These services allow any accepted form of input described in section 4.2.1 above.

Relevant parts of an Aircraft object are a Pilot object, a Vehicle Navigation System object, and a Vehicle Communications System object. The two domain-unique services of a Pilot object are Accept Clearance Directives From Controller and Maneuver Aircraft.
Two domain-unique classes which are specializations of the Vehicle Navigation System class are the Aircraft ILS Navigation System class and the Aircraft MLS Navigation System class; these classes are needed so that the aircraft can fly the appropriate approach route. The Vehicle Communications System class is taken directly from the ATC model with no modifications.
Figure 4-8. TASF Model: User Subject (Traffic)
The Resource Subject. The Resource structure, shown in figure 4-9, (parts 1, 2, and 3) borrowed all the specializations of the ATC model with the exception of the Aviation Weather Resource class.

The Airspace/Ground Resource object of TASF, shown in figure 4-9, (part 1) borrowed both objects which were part of the Airspace/Ground object of the ATC model: the Controlled Airspace object and the Airport Movement Area object. TASF is concerned with the location of airports for the purposes of spacing and sequencing flights for landing and as a source of flights. The specializations of the Airport Movement Area class identified in the six-month configuration of TASF are the La Guardia Airport class and the Teterboro Airport class. A part of an Airport Movement Area object of interest is a Runway object. The specialization of the Runway class identified is the La Guardia Runway 13 class. There is only one object belonging to the La Guardia Runway 13 class, and it represents the active runway at La Guardia Airport.

Parts of the Controlled Airspace object of importance to TASF are the En Route Controlled Airspace object and Terminal Controlled Airspace objects. Classes and objects identified in the En Route Controlled Airspace class and object structures, shown in figure 4-9, (part 2) are not domain-unique but are also found in the AERA 2 model. Both parts of the En Route Controlled Airspace object are identified, Route ("En Route Airway") objects and Airspace Volume objects. A specialization of the Airspace Volume class was identified, the Holding Pattern Airspace ("Holding Pattern") class. Part of a Route ("En Route Airway") object identified is the Fix/Waypoint ("En Route Waypoint") object.

Parts of the Terminal Controlled Airspace object, shown in figure 4-9, (part 3) are used in TASF, including Terminal Route objects and the Terminal Airspace Volume object. A specialization of the Terminal Airspace Volume class, called the New York TRACON class, was identified. Four specializations of the Terminal Route class are the MLS Route class, the ILS Route class, the STAR class (which was not used in the 6-MITF) and the SID class (which was also not used).

Other resources identified in the six-month configuration of TASF are shown in figure 4-10. The Airspace Surveillance System class, a specialization of the Surveillance System class, was borrowed from the ATC model. (The Ground Surveillance System specialization class is not a TASF concern.) The domain-unique specialization of the Airspace Surveillance System class is the ARTS III class. The ARTS III computer provides a display for the controller. A specialization of the ARTS III class identified by TASF is the ARTS III With Ghosting class.

The Navigation Resource class and the Navigation System class were borrowed from the ATC model. Of importance to TASF is the Landing Navigation System class, a specialization of the Navigation System class. The two domain-unique specializations of the Landing Navigation System class are the Instrument Landing System (ILS) class and the Microwave Landing System (MLS) class. The attributes of the Instrument Landing System (ILS) class identify variable data (signals) produced by the localizer and the glide slope transmitters. A part of the Instrument Landing System (ILS) object, the ILS Marker Beacon object, was also identified (although not used in the 6-MITF). An ILS Marker Beacon object can be used to represent either an outer marker beacon or a middle marker beacon. Services of a Microwave Landing System (MLS) object provide lateral and vertical position information to the aircraft.

© 2000 The MITRE Corporation. ALL RIGHTS RESERVED.
The Communication Resource class and the Communication System class were borrowed from the ATC model. The specialization of the Communications System class used in TASF is the Air/Ground Communications System. Two domain-unique specializations of the Air/Ground Communications System class, called the Data Link Communications System class and the Voice Radio Communications System class, are also used. The Voice Radio Communications System class was used in the laboratory for the communications between the final approach controller and the sim-
Figure 4-9. TASF Model: Resource Subject (Airspace/Ground Resource) (Part 1)
Figure 4-9. TASF Model: Resource Subject (Airspace/Ground Resource) (Part 2)
Figure 4-9. TASF Model: Resource Subject (Airspace/Ground Resource) (Part 3)
Figure 4-10. TASF MODEL: Resource Subject (Other Resources)
pilots. Since it was a direct communication, the frequency attribute defined in TASF was not modeled. The Data Link Communications System class was identified in TASF but not used in the 6-MITF.

**The Manager Subject.** The Manager subject of TASF, shown in figure 4-11, contains the Flight Manager class borrowed from the ATC model with several domain-unique classes identified. The Terminal Flight Manager class and the En Route Flight Manager class are the domain-unique specializations of the Flight Manager class. The en route flight manager is of importance to TASF for the purposes of accepting and receiving handoffs from a terminal flight manager in the stand-alone mode.

The Terminal Flight Manager class has four domain-unique specialization classes: the North Arrival Controller class, the South Arrival Controller class, the Departure Controller class, and the Final Approach Controller class. The North and South arrival controllers are the controllers that will accept flights from the en route controllers. The final approach controller is the controller who interacts with the sim-pilot to space and sequence flights on the approach routes. The departure controller accepts flights from the local controller on the ground and spaces them on the departure routes.

For the six-month illustration, there was no coordination between the controllers during handoffs. It was assumed that a controller in the next area of responsibility would be able to take the aircraft when the current controller was ready to hand it off.

**Event View: Spacing ILS Flight on Arrival Route.** This view is an example of how aircraft were maneuvered in the terminal area so that MLS and ILS approaches were merged for landing on runway 13 at La Guardia Airport. The final approach controller spaced the aircraft with the aid of the ARTS III ghosting aid. MLS approaches were projected onto the ILS route, and, in the 6-MITF, only the ILS aircraft were maneuvered in the spacing process. Figure 4-12 is a view of the spacing event using the classes and objects identified in TASF.

At the beginning of the event, a flight is flying an ILS approach to La Guardia Airport with the use of an ILS-specialized navigational system. The Flight object has an instance connection to a Final Approach Controller object which has control of the Flight. The Final Approach Controller is continually requesting the ARTS III With Ghosting object to perform the services of Display Aircraft Location and Project MLS Flight Onto ILS Route.

The Final Approach Controller object is performing the service Space Aircraft on Arrival Route. If the particular flight is not spaced properly with the other ILS flights or the ghosts of the MLS flights, the Final Approach Controller object will perform the service Issue Clearance Directives to Pilot and request the Pilot object to perform the services of Accept Clearance Directives From Controller and Maneuver Aircraft. The Pilot object will then request the Aircraft ILS Navigation System object to perform the service Accept/Implement Maneuver Information and will request the Aircraft object to perform the services Calculate Aircraft Point Position and Send Aircraft Point Position to ARTS (in this instance, ARTS III With Ghosting).
After the above-described services are performed, the flight should be properly spaced; however, this event is a continual process for the final approach controller.
Figure 4-11. TASF Model: Manager Subject
Figure 4-12. TASF Model View: Spacing ILS Flight On Arrival Route
4.2.2 AERA 2

The NAS Plan states that Automated En Route Air Traffic Control (AERA) will provide interactive software for use by the area control facility to plan and monitor the 4-dimensional flow of air traffic. Specifically, AERA will: (1) permit most aircraft on IFR flight plans to fly fuel-efficient profiles, (2) increase safety of the system by reducing the potential for operational error, (3) increase system capacity by integrating en route metering with local and national flow control, and (4) increase controller productivity by increasing the number of aircraft and volume of airspace that a control team can safely manage.

AERA Phase 2 (AERA 2) provides the following functionality: automatic reconformance of a flight plan with the flight track; trial planning; automatic detection and resolution of aircraft conflicts, airspace conflicts, and flow instruction problems; automated coordination aids; automatic clearance generation; automatic generation of controller reminders; and support for the automated use of data link in clearance delivery and information interchange.

The AERA 2 CHI Prototype (Mayo, 1990) was created for the purpose of supporting the evaluation of operational concepts, and provides a simulated ATC environment, simulated aircraft behavior, a subset of AERA 2 functionality, an interface for controller subjects and prototype developers, and tools for customizing (e.g., "scripting") sessions. In creating a model of the AERA 2 CHI Prototype, the human interface and the customizing tools were not modeled, but when information was needed for a display, the creation of that information was modeled. In addition, no distinction was made between processing that occurs offline and processing that happens "real time." For example, a Flight Plan is always created offline but a Flight Plan can be converted into a Current Plan, by the addition of a Trajectory, either offline during scenario generation or online during a simulation session. The service Create (an intrinsic service) was merely placed with the appropriate plan type (Flight Plan or Current Plan). When the plan type is "current," the service Create also involves a message sent to the Trajectory class to create a Trajectory object as part of the Current Plan object. (A Flight Plan does not have a Trajectory object as a part object.)

The AERA 2 CHI Prototype was not designed as an object-oriented system. The model of the AERA 2 CHI Prototype (hereafter called the "AERA 2 model") merely describes the prototype as if it were object-oriented, for the purpose of comparing prototype capabilities with experimental requirements. The AERA 2 model should not be interpreted as the design for the AERA 2 CHI Prototype.

4.2.2.1 Analysis of AERA 2 vs. the ATC Model

AERA 2 is usually described as a collection of functions (see, for example, [Ball and Taber, 1990] for a functional decomposition of AERA 2), and descriptions of the AERA 2 CHI Prototype follow this convention. Primary sources of information for classes, objects, attributes, and instance connections for the AERA 2 model were descriptions of the AERA 2 CHI Prototype database. Services and messages were inferred from functional descriptions and were placed with appropriate classes and objects. Except for (Mayo, 1990), the sources of information were unpublished internal MITRE documents and MITRE management and staff.
The AERA 2 model adds a large number of classes and objects to the ATC model for the purpose of storing data for use by the services of a few classes and objects. These new classes and objects tend to have only the intrinsic services and to behave passively (i.e., respond to requests rather than initiate requests). If the ATC model had been developed several levels lower, then many of these new classes and objects would have naturally appeared. These new classes and objects, and their relationships with classes, objects and services of the ATC model, are described in section 4.2.2.2 below.

Table 4-1 summarizes the assignment of AERA 2 CHI Prototype functionality to services of classes and objects. As stated above, processing to support logical displays (for the human interface) is not included in this assignment, e.g., the Situation Monitor functionality Conflict Notification is not included, but the processing to create the information (Automated Problem Detection) is. In the ATC model, no distinction is made between the human and the automation in providing a service; however, in the AERA 2 model, when a function is clearly automatic with no human intervention (e.g., the automatic handoff of an aircraft), that function might be placed as the service of an object that is not a manager (e.g., Handoff is a service of a Trajectory object). More information about the services in the AERA 2 model is provided in the model specification in appendix D.

### 4.2.2.2 The AERA 2 Model

In creating the AERA 2 model, the ATC model was examined for reusable components. AERA 2, for the most part, provides en route services for individual aircraft, and thus, for the most part, the Traffic class structure can be collapsed to the Flight class level, manager resources can be limited to flight managers, and airspace/ground resources can be limited to en route airspace resources (with new specializations). The AERA 2 CHI Prototype also uses information about terminal airspace and winds in order to model the aircraft planned path (the trajectory) and uses information about radar weather areas for display. The AERA 2 CHI Prototype does not need the Navigation Resource or Surveillance Resource structures of the ATC model. A Track object, which might have been produced by a Surveillance System object, is instead simulated based upon a Trajectory object. This restriction in the AERA 2 model scope is illustrated by figure 4-13 showing the AERA 2 model subjects. In this figure, and in later figures illustrating the AERA 2 model, components borrowed from the ATC model are shaded.

The AERA 2 model adds new classes and objects to represent the increase in detail needed by the AERA 2 CHI Prototype. While the ATC model contains only the flight plan, the AERA 2 model must contain all the plan types of interest to the AERA 2 CHI Prototype, thus adding a new structure for plans with new specializations and parts which subsume the flight plan. Processing of the new plan types requires special services. The AERA 2 CHI Prototype trajectory modeling capabilities require information about aircraft classes, normal routings near airports, and winds in order to build trajectories and require new classes and objects to store trajectories once they are built. Because of the new problem detection and resolution capabilities, new classes and objects are added to help in the detection of problems (e.g., airspace specializations and restrictions), and new classes and objects are added to represent the problems themselves and their resolutions. These additions are described in more detail below.
The User Subject. Figure 4-14 (a three-part figure) shows the User subject for the AERA 2 model. Part 1 of the figure emphasizes the classes and objects associated with the Flight class, part 2 emphasizes the Plan class, and part 3 emphasizes the Problem class. Parts 1 and 2 have the Flight Plan class in common, while parts 2 and 3 have the Plan and Machine Plan classes in common. (Alternatively, two new subjects called Plan and Problem could have been introduced.)

As shown in part 1 of the figure, a Flight object is made up of an Aircraft object and a Flight Plan object. (The clearance-type classes are not used in the AERA 2 model.) Each Flight Plan object may be connected to one or more Flight Plan Amendment objects. An Aircraft object may be connected with one or more Track objects, representing the actual position (past and present) of the aircraft (as opposed to the intended position, represented by attributes of the Flight Plan object). The Track object would normally be created based upon reports from a surveillance system-type object, but, in
Table 4-1. Translation of AERA 2 CHI Prototype Functions to AERA 2 Model Services

<table>
<thead>
<tr>
<th>AERA 2 CHI Prototype Functionality</th>
<th>AERA 2 Model Class &amp; Object</th>
<th>AERA 2 Model Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan Construction (All)</td>
<td>En Route Controller</td>
<td>Assist in Flight Planning*</td>
</tr>
<tr>
<td>Plan Construction (Creation) adds to Plan Construction (All)</td>
<td>En Route Controller</td>
<td>Request New Plan Creation</td>
</tr>
<tr>
<td>Plan Construction (Amendment) adds to Plan Construction (All)</td>
<td>En Route Controller</td>
<td>Request Plan Amendment</td>
</tr>
<tr>
<td>Plan Construction (Reconformance Aid) adds to Plan Construction (All)</td>
<td>En Route Controller</td>
<td>Request Reconformance Aid</td>
</tr>
<tr>
<td>Plan Construction (Limited Resolution Aid) adds to Plan Construction (All)</td>
<td>En Route Controller</td>
<td>Request Limited Resolution Aid</td>
</tr>
<tr>
<td>Automated Replan Processing adds to Plan Construction (All)</td>
<td>En Route Controller</td>
<td>Request Automated Replan</td>
</tr>
<tr>
<td>Automated Coordination adds to Plan Construction (All)</td>
<td>En Route Controller</td>
<td>Request Automated Coordination</td>
</tr>
<tr>
<td>Trajectory Estimation</td>
<td>Trajectory</td>
<td>Trajectory Estimation</td>
</tr>
<tr>
<td>Flight Simulation</td>
<td>Trajectory</td>
<td>Request Track Creation</td>
</tr>
<tr>
<td>Situation Monitoring (Resynchronization)</td>
<td>Trajectory</td>
<td>Resynchronization</td>
</tr>
<tr>
<td>Situation Monitoring (Handoff Control)</td>
<td>Trajectory</td>
<td>Handoff</td>
</tr>
<tr>
<td>Situation Monitoring (Top-of-Descent Controller Reminder)</td>
<td>Controller Reminder</td>
<td>Trajectory Estimation</td>
</tr>
<tr>
<td>Situation Monitoring (Out-of-Conformance Detection)</td>
<td>Track</td>
<td>Out-of-Conformance Detection</td>
</tr>
<tr>
<td>Problem Detection and Resolution (Automated Problem Detection)</td>
<td>En Route Controller</td>
<td>Detect, Predict &amp; Report</td>
</tr>
<tr>
<td>Problem Detection and Resolution (Automated Resolution Generation)</td>
<td>En Route Controller</td>
<td>Separate IFR Flights;*</td>
</tr>
</tbody>
</table>

* indicates a mandatory action.
<table>
<thead>
<tr>
<th>Data Link</th>
<th>En Route Controller</th>
<th>Generate &amp; Deliver Clearance*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Data Link</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pending Plan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Create</td>
</tr>
</tbody>
</table>

* Inherited
Figure 4-13. AERA 2 Model: Subjects
Figure 4-14. AERA 2 Model: User Subject (Flight) (Part 1)
Figure 4-14. AERA 2 Model: User Subject (Flight) (Part 2)
Figure 4-14. AERA 2 Model: User Subject (Flight) (Part 3)
the AERA 2 model, Track objects are created based upon simulated information from a Trajectory object. An Aircraft object is also connected with an Aircraft Class object, which is further connected with several Climb Profile, Descent Profile, and Long Range Cruise Profile objects. These new classes and objects behave passively and exist to provide information for trajectory modeling.

Part 2 of the figure introduces the new Plan class. A Plan object has at least three part objects: a Flight Plan object, a Trajectory object, and one or more Clearance Directive objects. The Flight Plan class of the AERA 2 model is essentially the same as the Flight Plan class of the ATC model. A trajectory provides a more detailed description of the four-dimensional predicted path of an aircraft; a Trajectory object is made up of State Segment objects, each of which is connected to a State Segment Start object and a State Segment End object. State segments describe the physical movement of an aircraft in terms of its bearing, acceleration, and gradient at given times and places. The Planning Region State Segment class is a specialization of the State Segment class, and represents that part of a trajectory that is contained within the planning region of interest (the "Planning Region" object). A Clearance Directive object represents an action, or maneuver, planned for the aircraft. A clearance directive is not an ATC clearance.

The Plan class has two specialization classes, the Current Flight Plan class and the Non-Current Flight Plan class. Each of the specialization classes, of course, also "inherit" the whole-part structure of the Plan class. The Non-Current Flight Plan class also has specializations, the Pending Plan class, the Trial Plan class, and the Machine Plan class. A specialization of the Trial Plan class is the Automated Replan Plan class. An Automated Replan Plan object is connected (by instance connection) to an Automated Replan Amendment object. An Automated Replan amendment represents a template for creating trial plans, called here automated replan plans, at specified times for re-evaluation.

Part 3 of the figure introduces problems and resolutions to problems. A Plan object may be connected to one or more Problem objects. A Multiple Problem object has one or more Problem objects as parts, where each of the Problem objects is connected to the same Plan object. The Problem class has four specializations: the Non-Conformance class, the Aircraft-Aircraft Conflict class, the Aircraft-Airspace Conflict class, and the Flow Instruction Non-Compliance class. Each Problem object is connected to up to ten Resolution objects, of which one is the highest-ranked resolution (i.e., the Resolution class has a specialization, the Highest-Ranked Resolution (HRR) class). A Resolution object is connected to one or two Machine Plan objects (i.e., a resolution to a problem can maneuver one or two aircraft).

**The Resource Subject.** Figure 4-15 (a two-part figure) shows the Airspace/Ground Resource structure of the Resource subject for the AERA 2 model. Part 1 contains all of the Airspace/Ground Resource class structure except for the structure beneath the En Route Controlled Airspace class, which is found in part 2.

The Airspace/Ground Resource object has several parts: a Controlled Airspace object and one or more Airport Movement Area objects. The Airport Movement Area class has a specialization, the Airport class. The AERA 2 model provides only limited information about an Airport object, needed for trajectory modeling. The Controlled Airspace object has several parts, including an En Route Controlled Airspace object and several Terminal Controlled Airspace objects. A Terminal
Controlled Airspace object has several parts: at least one Terminal Route object and at least one Terminal Airspace Volume object. The Terminal Route class has the STAR class and the Aircraft Descent Altitude Profile class as specializations, which are used by trajectory modeling. Since problem detection and resolution are not performed for that part of an aircraft trajectory in an approach control area or an Automated Problem Detection Inhibited Area (APDIA), the Terminal Airspace Volume class has the Approach Control Area class and the APDIA class as specializations. An Airport object
Figure 4-15. AERA 2 Model: Resource Subject (Airspace/Ground Resource) (Part 1)
Figure 4-15. AERA 2 Model: Resource Subject (Airspace/Ground Resource) (Part 2)
An En Route Controlled Airspace object has Route objects and Airspace Volume objects as parts. The Route class has one specialization, the Victor/Jet Route class (similar to the Jet Route class and Airway class of the ATC model). A Route object has one or more Route Segment (called "Airway Segment") objects and Fix/Waypoint (called "Fix") objects as parts. The Airspace Volume class has specialization classes ACF Airspace (called "Planning Region"), Sector, Protected Airspace Volume (called "Blocked Airspace"), and Holding Pattern Airspace (called "Holding Pattern"). An ACF Airspace object also contains Sector objects as parts. The Protected Airspace Volume class has specialization classes Restricted Area, Heavy Weather Area, and Holding Pattern. The Holding Pattern Airspace class is a specialization of the Airspace Volume class because of the ATC model, and a specialization of the Protected Airspace Volume class because of the AERA 2 CHI Prototype. An aircraft is allowed to enter a blocked airspace only by permission.

An ACF Airspace ("Planning Region") object may have an instance connection with one or more Restriction objects, representing flow restrictions. The Metering Restriction class is a specialization of the Restriction class, and a Metering Restriction object is made up of Metering Slot objects. A Metering Restriction object has an instance connection with a Fix/Waypoint (called "Fix") object, because the metering restriction calls for aircraft to be "metered to" a particular fix, which is accomplished by assigning aircraft to metering slots.

In part 2 of figure 4-15, many of the classes and objects coincide with classes and objects from the ATC model, differing mostly in the precise definitions of the attributes and services.

Figure 4-16 shows the other resources of the Resource subject for the AERA 2 model. These other resources have only peripheral use. The Data Link Communications System class is a specialization of the Air/Ground Communications System class, which is a specialization of the Communications System class. A Communications System object is a part of a Communications Resource object. The Data Link Communications System class, which is the only communications resource-type class unique to the AERA 2 model, represents an automatic acceptance of a plan amendment by a data link-equipped aircraft.

An Aviation Weather Resource object is made up of Aviation Weather System objects, each of which is made up of one or more Wind/Weather Product objects. Each of these classes and objects are borrowed from the ATC model. A Wind/Weather Product class has two AERA 2 model-unique specializations, the Radar Weather Area class and the Wind class. A Radar Weather Area object is only used for display. A Wind object is made up of Horizontal Wind Layer objects, and is used by trajectory modeling.

The Manager Subject. Figure 4-17 shows the Manager subject for the AERA 2 model. The Manager class has the Flight Manager class as a specialization, and the AERA 2 model adds the En Route Controller as a specialization of the Flight Manager class. The AERA 2 CHI Prototype does not include traffic manager or vehicle manager capabilities.
Event View: Aircraft-to-Aircraft Conflict Detection and Resolution. Because of the AERA 2 emphasis on automated problem detection and resolution, the ATC views in section 3.2.4.3 are applicable to the AERA 2 model with minimal changes: change the manager names from "Flight Manager" to "En Route Controller" and remove clearance-type classes and objects. For example, figure 4-18 illustrates an AERA 2 model version of figure 3-12, Aircraft-to-Aircraft Conflict Detection and Resolution Event. At the beginning of the event, an En Route Controller object (called
Figure 4-16. AERA 2 Model: Resource Subject (Other Resources)
Figure 4-17. AERA 2 Model: Manager Subject
En Route Controlled Airspace
Separation Minima

Sector
Separation Minima

En Route Controller 1
Area of Responsibility
Detect, Predict & Report Separation Violation

Request IFR Flight Separation

Flight 1
Type (IFR)
Flight Identification

En Route Controller 2
Area of Responsibility
Separate IFR Flights
Generate & Deliver Clearance

Request IFR Flight Separation

Flight 2
Type (IFR)
Flight Identification

Request Amend Flight Plan

Aircraft

Flight Plan
(attributes)
Amend Flight Plan

Pilot

Figure 4-18.  AERA 2 Model View: Aircraft-to-Aircraft Conflict Detection and Resolution

© 2000 The MITRE Corporation. ALL RIGHTS RESERVED.
"En Route Controller 1") with an instance connection with the En Route Controlled Airspace object has been performing its service Detect, Predict & Report Separation Violation. A violation of separation minima is predicted between two IFR flights that are planned to be in the same sector at the same time. It is assumed that the two Flight objects now have instance connections to the same En Route Controller object, called "En Route Controller 2." En Route Controller 1 sends a message to En Route Controller 2, requesting the performance of the Separate IFR Flights service. In carrying out this service, En Route Controller 2 also performs the general service Generate & Deliver Clearance, requesting the pilot of one of the two Flight objects ("Flight 2") to perform its service Plan Flight. If the Plan Flight service determines that the proposed change to the flight plan cannot be accepted, then a negative response is returned to En Route Controller 2; otherwise, a positive response is returned and the Pilot object requests the Flight Plan object to amend itself. The attributes of the Flight Plan which are affected by the Amend Flight Plan service are determined by the actual amendment.

This view makes use of two (possibly) different En Route Controller objects. En Route Controller 1, for which the service Detect, Predict & Report Separation Violation is automated in the AERA 2 CHI Prototype, identifies an aircraft conflict. En Route Controller 2, in performing the service Separate IFR Flights, determines resolutions for the conflict (automated) and selects a resolution (manual) for implementation. The Flight 2 services Plan Flight and the message Request Amend Flight Plan are simulated, since the amendment is always accepted by the flight.

4.3 AN ASSESSMENT OF NEEDED CAPABILITIES

Table 4-2 compares the needed capabilities of the experiment with the available capabilities of TASF and AERA 2 legacy software. The first column lists the classes and objects, attributes, and services of the experiment model. The second and third columns list the corresponding model components of the TASF model and the AERA 2 model components. A dash "-" in a model column means that this model does not have a component corresponding to the one listed on the same line for another model. Attribute and service names are prefixed by "a" and "s," respectively, and italicized so that they are easier to distinguish from class and object names. Using this table, a direct comparison between experiment needs and legacy software capabilities can be made. Some obvious conclusions are as follows:

(a) Terminal capabilities are available only from TASF, and are not provided by AERA 2 software.
(b) En route capabilities are available only from AERA 2 software, and are not provided by TASF software.
(c) Neither TASF nor AERA 2 provide some needed capabilities, such as the RDP-Like En Route Surveillance System; the Arrival Clearance, Departure Clearance, and En Route Clearance; the Flight Plan; the Point; the Washington Area Airport; the Regional Air Traffic Manager; the Tower Controller; certain services of the Terminal Controller (Final Arrival Sequencing & Spacing, Coordinate Arrivals With Departures, and Accept/Initiate Handoff); and certain services of the En Route Controller (Compute/Issue Delay-
Absorbing Maneuvers, Handoff Aircraft to Terminal Controller). These capabilities must be found elsewhere.

(d) For some experiment components, there are similar components in both the TASF model and the AERA 2 model, e.g., the Meter Fix and the Trajectory. In these cases, a more careful comparison is needed in order to determine which version of a component (if either) should be used.
<table>
<thead>
<tr>
<th>Experiment</th>
<th>TASF</th>
<th>AERA 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>90-Minute Flying Time Wedge</td>
<td>En Route Controlled Airspace</td>
<td>En Route Controlled Airspace</td>
</tr>
<tr>
<td>a: Boundary</td>
<td>a: Boundary Code</td>
<td>a: Location</td>
</tr>
<tr>
<td>s: None</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Aircraft-Aircraft Conflict</td>
<td>-</td>
<td>Aircraft-Aircraft Conflict</td>
</tr>
<tr>
<td>a: Object Aircraft</td>
<td>-</td>
<td>a: Object Aircraft</td>
</tr>
<tr>
<td>a: Start of Delay</td>
<td>-</td>
<td>a: Start of Delay</td>
</tr>
<tr>
<td>a: End of Delay</td>
<td>-</td>
<td>a: End of Delay</td>
</tr>
<tr>
<td>a: Subject Aircraft Start Location</td>
<td>-</td>
<td>a: Subject Aircraft Start Location</td>
</tr>
<tr>
<td>a: Object Aircraft Start Location</td>
<td>-</td>
<td>a: Object Aircraft Start Location</td>
</tr>
<tr>
<td>a: (many more - refer to AERA 2 model)</td>
<td>-</td>
<td>a: (many more)</td>
</tr>
<tr>
<td>s: None</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Aircraft-Airspace Conflict</td>
<td>-</td>
<td>Aircraft-Airspace Conflict</td>
</tr>
<tr>
<td>a: Separation</td>
<td>-</td>
<td>a: Separation</td>
</tr>
<tr>
<td>a: Minimum Miss Distance</td>
<td>-</td>
<td>a: Minimum Miss Distance</td>
</tr>
<tr>
<td>a: Start Location</td>
<td>-</td>
<td>a: Start Location</td>
</tr>
<tr>
<td>a: End Location</td>
<td>-</td>
<td>a: End Location</td>
</tr>
<tr>
<td>a: Airspace Name</td>
<td>-</td>
<td>a: Airspace Name</td>
</tr>
<tr>
<td>a: Airspace Type</td>
<td>-</td>
<td>a: Airspace Type</td>
</tr>
<tr>
<td>s: None</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Arrival Clearance</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>a: Approach Procedure (from ATC model)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s: Amend Clearance (from ATC model)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ARTS-Like Terminal Surveillance System</td>
<td>ARTS III/ARTS III With Ghosting</td>
<td>-</td>
</tr>
<tr>
<td>a: Area of Coverage</td>
<td>a: Area of Coverage</td>
<td>-</td>
</tr>
<tr>
<td>s: Provide User Interface</td>
<td>s: Display Aircraft location</td>
<td>-</td>
</tr>
<tr>
<td>s: Provide TAAS Capabilities</td>
<td>s: Project MLS Flight onto ILS Route (ARTS III With Ghosting only)</td>
<td>-</td>
</tr>
<tr>
<td>Data Link-Like Communications System</td>
<td>Data Link Communications System</td>
<td>Data Link Communications System</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>---------------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td><em>a</em>: Equipment Type (inherited)</td>
<td><em>a</em>: Type (Mode S/ACARS)</td>
<td>-</td>
</tr>
<tr>
<td><em>a</em>: Status (inherited)</td>
<td><em>a</em>: Status</td>
<td>-</td>
</tr>
<tr>
<td><em>s</em>: Provide Data Link-Like Services</td>
<td></td>
<td>-</td>
</tr>
</tbody>
</table>
Table 4-2. (Continued)

<table>
<thead>
<tr>
<th>Experiment Scenario Traffic Pattern</th>
<th>Air Traffic</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>a: Demand</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s: None</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experiment</th>
<th>TASF</th>
<th>AERA 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Departure Clearance</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>a: Departure Procedure</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s: Amend Clearance</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>En Route Clearance</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>a: Route</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>a: Altitude Profile</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>a: Speed Schedule</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>a: Holding Instructions</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>a: Clearance Limit</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s: Amend Clearance</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>En Route Controller</td>
<td>En Route Flight Manager</td>
<td>En Route Controller</td>
</tr>
<tr>
<td>a: Area of Responsibility</td>
<td>-</td>
<td>a: Sector Identification</td>
</tr>
<tr>
<td>(inherited)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s: Identify Conflicts</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>s: Propose Resolutions</td>
<td>-</td>
<td>s: Assist in Flight Planning</td>
</tr>
<tr>
<td>(inherited)</td>
<td></td>
<td>(inherited)</td>
</tr>
<tr>
<td>s: Compute Delay-Absorbing Maneuvers</td>
<td>-</td>
<td>s: Ensure Restriction Compliance (inherited)</td>
</tr>
<tr>
<td>s: Issue Delay-Absorbing Maneuvers</td>
<td>-</td>
<td>s: Generate &amp; Deliver Clearance (inherited)</td>
</tr>
<tr>
<td>s: Handoff Aircraft to Terminal Controller</td>
<td>-</td>
<td>s: (see Trajectory)</td>
</tr>
</tbody>
</table>

© 2000 The MITRE Corporation. ALL RIGHTS RESERVED.
<table>
<thead>
<tr>
<th>Flight Plan</th>
<th>Flight Plan</th>
<th>Flight Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>a: Expected Meter Fix Time</em></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><em>a: Expected Airport Arrival Time</em></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Experiment</td>
<td>TASF</td>
<td>AERA 2</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-------------------------------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>Flight Plan (concluded)</td>
<td>Flight Plan (concluded)</td>
<td>Flight Plan (concluded)</td>
</tr>
<tr>
<td>(a: \text{Flow Restriction Time})</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(s: \text{None})</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Flow Instruction Non-Compliance</td>
<td>-</td>
<td>Flow Instruction Non-Compliance</td>
</tr>
<tr>
<td>(a: \text{None})</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(s: \text{None})</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Highest-Ranked Resolution</td>
<td>-</td>
<td>Highest-Ranked Resolution</td>
</tr>
<tr>
<td>(a: \text{None})</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(s: \text{None})</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ILS-Like Landing Navigation System</td>
<td>Instrument Landing System (ILS)</td>
<td>-</td>
</tr>
<tr>
<td>(a: \text{Location (La Guardia)})</td>
<td>(a: \text{Localizer Bearing})</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(a: \text{Localizer Latitude})</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(a: \text{Localizer Longitude})</td>
<td>-</td>
</tr>
<tr>
<td>(s: \text{Provide ILS Data to Aircraft})</td>
<td>(s: \text{Provide Glide Slope to Aircraft Avionics})</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(s: \text{Provide Distance to Aircraft Avionics})</td>
<td>-</td>
</tr>
<tr>
<td>ILS Route</td>
<td>ILS Route</td>
<td>-</td>
</tr>
<tr>
<td>(a: \text{Location (La Guardia)})</td>
<td>(a: \text{Location (La Guardia)})</td>
<td>-</td>
</tr>
<tr>
<td>(s: \text{None})</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>La Guardia Airport</td>
<td>La Guardia Airport</td>
<td>-</td>
</tr>
<tr>
<td>(a: \text{Arrival Capacity})</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(a: \text{Departure Capacity})</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(a: \text{Runway Configuration})</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(a: \text{Expected Departure Rate})</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(s: \text{None})</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Meter Fix</td>
<td>Fix/Waypoint (&quot;En Route Waypoint&quot;)</td>
<td>Fix/Waypoint (&quot;Fix&quot;)</td>
</tr>
<tr>
<td>(a: \text{Location (En Route/Terminal Boundary)})</td>
<td>(a: \text{Waypoint Type})</td>
<td>(a: \text{Flow Instruction})</td>
</tr>
<tr>
<td></td>
<td>(a: \text{Region Code})</td>
<td>(a: \text{Position (latitude/longitude)})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(a: \text{Minimum/Maximum Altitude})</td>
</tr>
</tbody>
</table>
s: Update Demand (inherited)  -  -
s: Update Load (inherited)  -  -
<table>
<thead>
<tr>
<th>Experiment</th>
<th>TASF</th>
<th>AERA 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLS-Like Landing Navigation System</td>
<td>Microwave Landing System (MSL)</td>
<td></td>
</tr>
<tr>
<td>a: Location (La Guardia)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s: Provide MLS Data to Aircraft</td>
<td>s: Provide x-y-z to Aircraft Avionics</td>
<td></td>
</tr>
<tr>
<td>MLS Route</td>
<td>MLS Route</td>
<td>-</td>
</tr>
<tr>
<td>a: Location (La Guardia)</td>
<td>a: Location (La Guardia)</td>
<td>-</td>
</tr>
<tr>
<td>s: None</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Multiple Problem</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>a: None</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s: None</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>New York TRACON</td>
<td>New York TRACON</td>
<td>-</td>
</tr>
<tr>
<td>a: Boundary</td>
<td>a: Criteria</td>
<td>-</td>
</tr>
<tr>
<td>s: None</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Pilot</td>
<td>Pilot</td>
<td>-</td>
</tr>
<tr>
<td>a: None</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s: Receive/Implement Vectoring Instructions</td>
<td>s: Accept Clearance Directives from Controller</td>
<td>-</td>
</tr>
<tr>
<td>s: Maneuver Aircraft</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>Point</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>a: None</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s: None</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Problem</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>a: Subject Aircraft</td>
<td>-</td>
<td>a: Subject Aircraft</td>
</tr>
<tr>
<td>s: None</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>RDP-Like En Route Surveillance System</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>a: Area of Coverage</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s: Provide Radar Data Processing (RDP) Services for En Route Airspace</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Regional Air Traffic Manager</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>a: None</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s: <em>Impose Flow Instruction</em></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s: <em>Create Metering Flow Restriction</em></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s: <em>Sequence/Resequence Flights</em></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s: <em>Adjust Flight Schedule</em></td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Table 4-2. (Continued)

<table>
<thead>
<tr>
<th>Experiment</th>
<th>TASF</th>
<th>AERA 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution</td>
<td>-</td>
<td>Resolution</td>
</tr>
<tr>
<td>a: Aircraft Identification</td>
<td>-</td>
<td>a: Aircraft Identification</td>
</tr>
<tr>
<td>a: Expiration Time</td>
<td>-</td>
<td>a: Expiration Time</td>
</tr>
<tr>
<td>a: Maneuver Type</td>
<td>-</td>
<td>a: Maneuver Type</td>
</tr>
<tr>
<td>a: Maneuver Parameters</td>
<td>-</td>
<td>a: Maneuver Parameters</td>
</tr>
<tr>
<td>s: None</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Segment</td>
<td>-</td>
<td>State Segment</td>
</tr>
<tr>
<td>a: Bearing</td>
<td>-</td>
<td>a: Bearing</td>
</tr>
<tr>
<td>a: Acceleration</td>
<td>-</td>
<td>a: Acceleration</td>
</tr>
<tr>
<td>a: Gradient</td>
<td>-</td>
<td>a: Gradient</td>
</tr>
<tr>
<td>s: None</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Terminal Controller</td>
<td>North, South, Departure, Final Approach Controller</td>
<td>-</td>
</tr>
<tr>
<td>a: None</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>s: Accept/Initiate Handoffs</td>
<td>s: Accept North Arrival Flights from En Route Flight Manager (North Controller)</td>
<td>-</td>
</tr>
<tr>
<td>s: Accept South Arrival Flights from En Route Flight Manager (South Controller)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s: Accept Flights from Local Controller (Departure Controller)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s: Coordinate Arrivals With Departures.</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>s: Space Aircraft on Departure Routes (Departure Controller)</td>
<td>-</td>
</tr>
<tr>
<td>s: Final Arrival Sequencing &amp; Spacing</td>
<td>s: Space Aircraft on Arrival Route (Final Approach Controller)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>s: Issue Clearance Directives to Pilot (Final Approach Controller)</td>
<td></td>
</tr>
<tr>
<td>Tower Controller</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>a: None</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
s: Coordinate Arrivals With Terminal Controller
s: Coordinate Departures With Terminal Controller
Table 4-2. (Concluded)

<table>
<thead>
<tr>
<th>Experiment</th>
<th>TASF</th>
<th>AERA 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trajectory</td>
<td>-</td>
<td>Trajectory</td>
</tr>
<tr>
<td>a: None</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s: Trajectory Estimation</td>
<td>-</td>
<td>s: Trajectory Estimation</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>s: Handoff</td>
</tr>
<tr>
<td>Vehicle Navigation System</td>
<td>Aircraft ILS Navigation System</td>
<td>-</td>
</tr>
<tr>
<td>a: Equipment Type</td>
<td>a: Type (ILS)</td>
<td>-</td>
</tr>
<tr>
<td>s: Accept Navigational Guidance</td>
<td>s: Localizer Detected</td>
<td>-</td>
</tr>
<tr>
<td>s: Interrogate Navigational Aid</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Vehicle Navigation System</td>
<td>Aircraft MLS Navigation System</td>
<td>-</td>
</tr>
<tr>
<td>a: Equipment Type</td>
<td>a: Type (MLS)</td>
<td>-</td>
</tr>
<tr>
<td>s: Accept Navigational Guidance</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s: Interrogate Navigational Aid</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Voice Radio-Like Communications System</td>
<td>Voice Radio Communications System</td>
<td>-</td>
</tr>
<tr>
<td>a: Equipment Type (Voice Radio-Like)</td>
<td>a: Equipment Type (Voice Radio)</td>
<td>-</td>
</tr>
<tr>
<td>Washington Area Airport</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>a: Runway Configuration</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>a: Expected Departure Rate</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s: None</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
APPENDIX A

ATC MODEL SPECIFICATIONS

This appendix contains the class and object specifications for the ATC model. The specifications are ordered by class name, and are formatted as follows:

class - name and a general description of the class and its objects. A source, such as the Airman's Information Manual (AIM), might be given.

structure - an identification of the classes and objects directly above and beneath this class and its objects in a generalization-specialization or whole-part structure; unless otherwise stated, a generalization-specialization structure involves classes and a whole-part structure involves objects.

part of
contains parts
specialization of
generalization of

instance connections - an identification of the instance connections between an object of the class and other objects.

message connections - an identification of the message connections between this class or object and other classes or objects; unless otherwise stated, the message is between objects. Messages requesting intrinsic services are not identified.

object states - an identification (in text, a table, a state transition diagram, Petri net, etc.) of the states and transitions between states of an object, possibly including attribute values associated with achieving each state.

attributes - name and a textual description of each attribute, which might include constraints (e.g., default values, the impact of the values of other attributes) on its value.

services - name and a description (by a bulleted list, service chart, or other device) of the processing performed by each service, including (when necessary) an identification of the object states associated with the service and/or how the service behaves differently depending on the state of the object.

external inputs - information provided by sources external to the domain of study.

external outputs - information provided to entities external to the domain of study.

notes - additional information about the class or object which is not already provided.
specification  ACF Airspace - the airspace controlled by an area control facility.

structure

specialization of  Airspace Volume

contains parts  Sector

instance connections

instance connection  ACF Airspace (0, m) <-> Flight (0, m)
instance connection  ACF Airspace (0, m) <-> Aircraft (0, m)
instance connection  ACF Airspace (0, m) <-> Flight Manager (0, m)
instance connection  ACF Airspace (0, m) <-> Air Traffic Manager (0, m)
instance connection  ACF Airspace (0, m) <-> Traffic Manager (0, m)

message connection  Intrinsic only

object states  None

attributes

attribute  Capacity  (see Airspace/Ground Resource)
attribute  Configuration  (see Airspace/Ground Resource)
attribute  Demand  (see Airspace/Ground Resource)
attribute  Load  (see Airspace/Ground Resource)
attribute  Location  (see Airspace/Ground Resource)
attribute  Name  (see Airspace/Ground Resource)
attribute  Saturation Threshold  (see Airspace/Ground Resource)
attribute  Separation Minima  (see Airspace/Ground Resource)
attribute  Usage Restrictions  (see Airspace/Ground Resource)

services  The ACF Airspace class performs the implicit services create, connect, and release. An ACF Airspace object performs the implicit service access.
service  Update Demand

- Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service  Update Load

- Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input  None

external output  None

notes  None
specification  Aircraft - device(s) that are used or intended to be used for flight in the air. (AIM, 1991)

structure

part of  Flight

part of  Vehicle

contains parts  Pilot; Vehicle Surveillance System; Vehicle Navigation System; Vehicle Communications System

instance connection  Aircraft (0, m) <-> airspace-type resource (1, m)

message connection  Create Flight Object, sent to Flight class

object states  An Aircraft object has two states, airborne and not airborne, as determined by the value of its attribute Location.

attributes

attribute  Aircraft Identification - a unique identification; the radio call sign.

attribute  Location - the position of the aircraft as known by ATC, whether determined by ATC or reported to ATC by the aircraft itself and recorded by ATC. Location includes lateral and vertical dimensions and time.

services  The Aircraft class performs the implicit services create, connect, and release. An Aircraft object performs the implicit service access.

service create

• Create and initialize, as specified for the implicit service create.

• When an Aircraft object is created and there does not already exist a Flight object of which the Aircraft object is to be a part, requests the Flight class to create a Flight object of which the Aircraft object is a part.

external input  None

external output  None

notes  The AIM definition also adds "and when used in air traffic control terminology, may include the flight crew." However, in this model, the flight crew is described separately and only the pilot is of interest.
specification  Air/Ground Communications System - a system of communications media provided by the ATC system for the transfer of information between the ATC system and aircraft and the management of the information.

structure

specialization of  Communications System

instance connection  Air/Ground Communications System (0, m) <-> Vehicle Communications System (0, m)

message connections

message  Request Accept Message, to manager-type object
message  Request Accept Message, to Vehicle Communications System
message  Request Transmit Message, from manager-type object
message  Request Transmit Message, from Vehicle Communications System

object states  None

attributes

attribute  Area of Coverage (Inherited)
attribute  Equipment Type (Inherited)
attribute  Status (Inherited)

services  The Air/Ground Communications System class performs the implicit services create, connect, and release. An Air/Ground Communications System object performs the implicit service access.

service  connect (Inherited)
service  Transmit Message (Inherited)

external input  None

external output  None

notes  Specializations of the Air/Ground Communications System class could be created to distinguish between communications by voice radio and communications by data link.
specification  Airport Movement Area - all taxiways, runways, and other areas of an airport that are used for taxiing, takeoff, and landing of aircraft, exclusive of loading ramps and parking areas. (Springen, 1989)

structure

part of  Airspace/Ground Resource

contains parts  Runway; Taxiway; Holding Area

instance connections

instance connection  Airport Movement Area (1) <-> Vehicle (0, m)

instance connection  Airport Movement Area (1, 2) <-> Aircraft (0, m)

instance connection  Airport Movement Area (0, 1) <-> Vehicle Manager (0, m)

instance connection  Airport Movement Area (0, m) <-> Ground Traffic Manager (0, m)

instance connection  Airport Movement Area (0, m) <-> Traffic Manager (0, m)

object states  None

attributes

attribute  Airport Identification - the identification of the airport

attribute  Capacity  (see Airspace/Ground Resource)

attribute  Configuration  (see Airspace/Ground Resource)

attribute  Demand  (see Airspace/Ground Resource)

attribute  Load  (see Airspace/Ground Resource)

attribute  Location  (see Airspace/Ground Resource)

attribute  Name  (see Airspace/Ground Resource)

attribute  Saturation Threshold  (see Airspace/Ground Resource)

attribute  Separation Minima  (see Airspace/Ground Resource)

attribute  Usage Restrictions  (see Airspace/Ground Resource)
services  The Airport Movement Area class performs the implicit services create, connect, and release. An Airport Movement Area object performs the implicit service access.

service  Update Demand

• Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service  Update Load

• Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input  None

external output  None

notes  None
specification Airport Radar Services Area - ARSA; regulatory airspace surrounding designated airports wherein ATC provides vectoring and sequencing on a full-time basis for all IFR and VFR aircraft. The service provided in an ARSA is called ARSA service which includes IFR/IFR standard separation, IFR/VFR traffic advisories and conflict resolution, and VFR/VFR traffic advisories and, as appropriate, safety alerts. (AIM)

structure

specialization of Terminal Airspace Volume

instance connections

instance connection Airport Radar Services Area (0, m) <-> Flight (0, m)
instance connection Airport Radar Services Area (0, m) <-> Aircraft (0, m)
instance connection Airport Radar Services Area (0, m) <-> Flight Manager (0, m)
instance connection Airport Radar Services Area (0, m) <-> Air Traffic Manager (0, m)
instance connection Airport Radar Services Area (0, m) <-> Traffic Manager (0, m)

message connection Intrinsic only

object states None

attributes

attribute Capacity (see Airspace/Ground Resource)
attribute Configuration (see Airspace/Ground Resource)
attribute Demand (see Airspace/Ground Resource)
attribute Load (see Airspace/Ground Resource)
attribute Location (see Airspace/Ground Resource)
attribute Name (see Airspace/Ground Resource)
attribute Saturation Threshold (see Airspace/Ground Resource)
attribute Separation Minima (see Airspace/Ground Resource)
attribute Usage Restrictions (see Airspace/Ground Resource)
services  The Airport Radar Services Area class performs the implicit services create, connect, and release. An Airport Radar Services Area object performs the implicit service access.

service  Update Demand
  • Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service  Update Load
  • Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input  None

external output  None

notes  None
specification  Airport Traffic Area - unless otherwise specifically designated in FAR Part 91, that airspace within a horizontal radius of five statute miles from the geographical center of any airport in which a control tower is operating, extending from the surface up to, but not including, an altitude of 3,000 feet above the elevation of the airport.  (AIM)

structure

specialization of  Terminal Airspace Volume

instance connections

instance connection  Airport Traffic Area (0, m) <-> Flight (0, m)
instance connection  Airport Traffic Area (0, m) <-> Aircraft (0, m)
instance connection  Airport Traffic Area (0, m) <-> Flight Manager (0, m)
instance connection  Airport Traffic Area (0, m) <-> Air Traffic Manager (0, m)
instance connection  Airport Traffic Area (0, m) <-> Traffic Manager (0, m)

message connection  Intrinsic only

object states  None

attributes

attribute  Capacity  (see Airspace/Ground Resource)
attribute  Configuration  (see Airspace/Ground Resource)
attribute  Demand  (see Airspace/Ground Resource)
attribute  Load  (see Airspace/Ground Resource)
attribute  Location  (see Airspace/Ground Resource)
attribute  Name  (see Airspace/Ground Resource)
attribute  Saturation Threshold  (see Airspace/Ground Resource)
attribute  Separation Minima  (see Airspace/Ground Resource)
attribute  Usage Restrictions  (see Airspace/Ground Resource)
services  The Airport Traffic Area class performs the implicit services create, connect, and release. An Airport Traffic Area object performs the implicit service access.

service  Update Demand

•  Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service  Update Load

•  Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input  None

external output  None

notes  None
**specification**  Airspace/Ground Resource - the airspace and ground, used by flights and vehicles, over which the ATC system has control.

**structure**

contains parts  Controlled Airspace;  Airport Movement Area

instance connections

instance connection  Airspace/Ground Resource (1) <-> Flight (0, m)
instance connection  Airspace/Ground Resource (1) <-> Vehicle (0, m)
instance connection  Airspace/Ground Resource (1) <-> Aircraft (0, m)
instance connection  Airspace/Ground Resource (1) <-> Ground Vehicle (0, m)
instance connection  Airspace/Ground Resource (1) <-> manager-type object (0, m)

message connection  Intrinsic only

object states  None

attributes

attribute  Capacity - estimate of the number of flights or vehicles per unit time that can be accommodated by the resource (e.g., arrival capacity, departure capacity, sector capacity, fix capacity, airway capacity).

attribute  Configuration - the relative placement of parts of the resource (e.g., which sectors make up an area) or the usage of the resource (e.g., an arrival sector vs. a departure sector, the subset of runways in use and whether each is used for arrivals or departures).

attribute  Demand - estimate of the number of flights or vehicles per unit time that plan to use the resource (e.g., arrival demand, departure demand, sector demand, fix demand, airway demand).

attribute  Load - actual utilization of a resource by aircraft or ground vehicles in a specific time period.

attribute  Location - the physical location (e.g., lateral boundaries and altitude ranges) of the resource.

attribute  Name - the ATC-assigned identification.
attribute Saturation Threshold - a demand or load value that should cause an alert.

attribute Separation Minima - the minimum longitudinal, lateral, or vertical distances by which aircraft are spaced through the application of ATC procedures. (AIM)

attribute Usage Restrictions - restrictions on the usage of a resource by flights or vehicles (e.g., an arrival rate of 40 between 1700 and 1800 UTC).

services The Airspace/Ground Resource class performs the implicit services create, connect, and release. An Airspace/Ground Resource object performs the implicit service access.

service Update Demand

• Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service Update Load

• Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input None

external output None

notes (1) Instance connections with Flight objects and Vehicle objects are related to the value of the Demand attribute; instance connections with Aircraft objects and Ground Vehicle objects are related to the value of the Load attribute. (2) The Demand attribute is not single-valued, but has values for each time period of interest. (3) The Location attribute is not necessarily single-valued (e.g., a sector may be defined by shelves). (4) The Separation Minima attribute is not necessarily single-valued (e.g., may have separation values for longitudinal, lateral, and vertical minima).
**specification**  Airspace Surveillance System - a system for the tracking of airborne aircraft positions and the management of the information provided.

**structure**

**specialization of**  Surveillance System

**instance connection**  Airspace Surveillance System \((0, m) \leftrightarrow\) Vehicle Surveillance System \((0, m)\)

**message connections**

**message**  Request Altitude Report, sent from Airspace Surveillance System

**message**  Transmit & Receive Radar Pulse, sent from Airspace Surveillance System

**message**  Transmit & Receive Radar Pulse, sent to Airspace Surveillance System

**object states**  None

**attributes**

**attribute**  Area of Coverage (Inherited)

**attribute**  Equipment Type (Inherited)

**attribute**  Status (Inherited)

**services**  The Airspace Surveillance System class performs the implicit services create, connect, and release. An Airspace Surveillance System object performs the implicit service access.

**service**  Locate, Identify & Report Flight

- Locate each aircraft in the area of coverage, determine the identification of the flight of which the aircraft is a part, and report its identification and location.

**external input**  None

**external output**  None

**notes**  In reality, an airspace surveillance system might not be able to locate and identify all aircraft in its area of coverage, due in part to the lack of aircraft equipage and in part to capacity limitations of the surveillance system.
**specification**  Airspace Volume - any volume of en route airspace.

**structure**

*part of*  En Route Controlled Airspace

*generalization of*  ACF Airspace; Sector; Protected Airspace Volume; Holding Pattern Airspace

**instance connections**

*instance connection*  Airspace Volume (0, m) <-> Flight (0, m)

*instance connection*  Airspace Volume (0, m) <-> Aircraft (0, m)

*instance connection*  Airspace Volume (0, m) <-> Flight Manager (0, m)

*instance connection*  Airspace Volume (0, m) <-> Air Traffic Manager (0, m)

*instance connection*  Airspace Volume (0, m) <-> Traffic Manager (0, m)

**message connection**  Intrinsic only

**object states**  None

**attributes**

*attribute*  Capacity  (see Airspace/Ground Resource)

*attribute*  Configuration  (see Airspace/Ground Resource)

*attribute*  Demand  (see Airspace/Ground Resource)

*attribute*  Load  (see Airspace/Ground Resource)

*attribute*  Location  (see Airspace/Ground Resource)

*attribute*  Name  (see Airspace/Ground Resource)

*attribute*  Saturation Threshold  (see Airspace/Ground Resource)

*attribute*  Separation Minima  (see Airspace/Ground Resource)

*attribute*  Usage Restrictions  (see Airspace/Ground Resource)
The Airspace Volume class performs the implicit services create, connect, and release. An Airspace Volume object performs the implicit service access.

**service** Update Demand

- Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

**service** Update Load

- Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

**external input** None

**external output** None

**notes** (1) The Airspace Volume class was introduced to permit the inclusion of any arbitrarily-defined volume of airspace. (2) The list of specialization class is not exhaustive, but includes commonly-defined types of airspace.
specification  Air Traffic - a grouping of flights determined by a defined criterion or set of criteria.

structure

specialization of  Traffic

contains part  Flight

instance connection  Air Traffic (0, m) <-> Air Traffic Manager (0, m)

message connection  None

object states  None

attribute  Selection Criteria (inherited from Traffic)

services  The Air Traffic class performs the implicit services create, connect, and release. An Air Traffic object performs the implicit service access.

service create

•  When requested by an Air Traffic Manager object, create an Air Traffic object with identified Flight objects as parts

external input  None

external output  None

notes  The definition of air traffic is unique to this model, although the term is sometimes used to mean either all airborne flights or some subset of flights defined by some understood but not rigorously-defined criterion (e.g., Chicago's traffic). The Federal Aviation Regulations define "air traffic" as "aircraft operating in the air or on an airport surface, exclusive of loading ramps and parking areas." The Air Traffic class exists in this model to allow the creation of well-defined groupings of flights for specific purposes (e.g., all flights above FL240 in Cleveland Center between 2100 and 2400 UTC, all flights outbound from Washington Center and inbound to JFK Airport on 19 March).
**specification**  Air Traffic Manager - one who plans or supervises air traffic.

**structure**

**specialization of**  Traffic Manager

**instance connections**

**instance connection**  Air Traffic Manager (0, m) <-> Air Traffic (0, m)

**instance connection**  Air Traffic Manager (0, m) <-> airspace-type resource (0, m)

**instance connection**  Air Traffic Manager (0, m) <-> Interfacility Communications System (0, m)

**instance connection**  Air Traffic Manager (0, m) <-> Intrafacility Communications System (0, m)

**message connections**

**message**  Create Air Traffic object, sent to the Air Traffic class service "create"

**message**  Request Usage Restrictions, sent from a Ground Traffic Manager object

**message**  Request Usage Restrictions, sent to a Ground Traffic Manager object

**message**  Request Usage Restrictions, sent from an Air Traffic Manager object

**message**  Request Usage Restrictions, sent to an Air Traffic Manager object

**message**  Provide Weather Forecast, sent to an Aviation Weather System Manager object

**object states**  None

**attribute**  Area of Responsibility (inherited)

**services**  The Air Traffic Manager class performs the implicit services create, connect, and release. An Air Traffic Manager object performs the implicit service access.

**service**  Detect, Predict & Report Saturation

- Determine whether saturation of an airspace-type resource is happening (based on the Load attribute) or is planned to happen (based on the Demand attribute) and notify the appropriate manager-type object.

**service**  Determine Airspace Capacity

© 2000 The MITRE Corporation. ALL RIGHTS RESERVED.
• Determine the number of flights that can be accommodated by an airspace-type resource per unit time or instantaneously.

service Reserve Airspace

• Coordinate with users that have special requirements for reserved blocks of airspace.

service Resolve Airspace Saturation Problem

• Develop directives in response to present and predicted traffic flow problems.

service Set Airspace Saturation Threshold

• Based upon the airspace-type resource capacity and other factors, determine a utilization value which, once predicted to be achieved or actually achieved, should cause an alert.

external input  None

external output  None

notes  (1) The Air Traffic Manager class does not represent a currently-staffed position; instead, it includes parts of the functionality of the Traffic Management Specialist, the Central Altitude Reservation Function Specialist, the Traffic Management Coordinator, and ATCT controllers. Other parts of the functionality of these positions are placed with other manager-type objects (e.g., some of the functionality of the local controller, an ATCT controller, is placed with the Flight Manager class, and some with the Vehicle Manager class). (2) The services of an Air Traffic Manager object might be performed either by humans or by automation.
specification  Airway - a control area or portion thereof established in the form of a corridor, the centerline of which is defined by radio navigational aids (AIM)

structure

specialization of  Route

contains parts (inherited)  Fix/Waypoint; Route Segment

instance connections

instance connection  Airway (0, m) <-> Flight (0, m)

instance connection  Airway (0, m) <-> Aircraft (0, m)

instance connection  Airway (0, m) <-> Flight Manager (0, m)

instance connection  Airway (0, m) <-> Air Traffic Manager (0, m)

instance connection  Airway (0, m) <-> Traffic Manager (0, m)

message connection  Intrinsic only

object states  None

attributes

attribute  Capacity (Inherited)

attribute  Configuration (Inherited)

attribute  Demand (Inherited)

attribute  Load (Inherited)

attribute  Location (Inherited)

attribute  Name (Inherited)

attribute  Saturation Threshold (Inherited)

attribute  Separation Minima (Inherited)

attribute  Usage Restrictions (Inherited)
services  The Airway class performs the implicit services create, connect, and release. An Airway object performs the implicit service access.

  service  Update Demand
  •  Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

  service  Update Load
  •  Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input  None

external output  None

notes  None
specification  Area Route - a random area navigation route; a direct route, based on area navigation capabilities, between waypoints defined in terms of latitude/longitude coordinates, degree/distance fixes, or offsets from published or established routes/airways at a specified distance and direction. (adapted from AIM)

structure

specialization of  Route

contains parts (inherited)  Fix/Waypoint;  Route Segment

instance connections

instance connection  Area Route (0, m) <-> Flight (0, m)

instance connection  Area Route (0, m) <-> Aircraft (0, m)

instance connection  Area Route (0, m) <-> Flight Manager (0, m)

instance connection  Area Route (0, m) <-> Air Traffic Manager (0, m)

instance connection  Area Route (0, m) <-> Traffic Manager (0, m)

message connection  Intrinsic only

object states  None

attributes

attribute  Capacity (Inherited)

attribute  Configuration (Inherited)

attribute  Demand (Inherited)

attribute  Load (Inherited)

attribute  Location (Inherited)

attribute  Name (Inherited)

attribute  Saturation Threshold (Inherited)

attribute  Separation Minima (Inherited)

attribute  Usage Restrictions (Inherited)
The Area Route class performs the implicit services create, connect, and release. An Area Route object performs the implicit service access.

**service** Update Demand

- Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

**service** Update Load

- Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

**external input** None

**external output** None

**notes** None
specification  Arrival Clearance - the clearances issued to a flight, and not yet completely carried out, for the arrival phase of flight. The arrival clearance includes any standard or instrument approach procedure, as well as any altitude and vector assignments issued, or visual approach or contact approach approved, that must be followed in leaving the en route route until cleared to land on a runway. The model name "arrival clearance" is not to be confused with the ATC term "approach clearance." The approach clearance (authorization to conduct an instrument approach) is a subset of the arrival clearance.

structure

specialization of  Clearance

part of  Flight Clearance

instance connection  None

message connections

message  Request Amend Clearance, sent from Pilot object

object states  None

attribute  Approach Procedure - the identification of a standard or special instrument approach procedure, visual or contact approach, vectors and altitude assignments that the aircraft must follow in leaving the en route route until cleared to land

services  The Arrival Clearance class performs the implicit services create, connect, and release. An Arrival Clearance object performs the implicit service access.

service  Amend Clearance

• Determine the affected attributes of the Arrival Clearance, and the needed change

• Change the identified attributes to the new values

external input  None

external output  None

notes  The arrival clearance, for this model, does not include the clearance to land, because the clearance to land represents an allocation of a ground-type resource. The clearance to land is included as part of a ground clearance.
specification  Aviation Weather Resource - the wind and weather products available to users of the
ATC system (e.g., an area forecast, a surface aviation weather report) to assist in flight planning and
the management of the basic wind/weather products in the creation of specialized products (e.g., a
preflight weather briefing).

structure

  specialization of  Resource

  contains parts  Aviation Weather System

instance connection  None

message connections  None

object states  None

attribute  None

services  The Aviation Weather Resource class performs the implicit services create, connect,
and release. An Aviation Weather Resource object performs the implicit service access.

external input  None

external output  None

notes  (1) The Aviation Weather Resource class exists only for convenience as the highest-level
aviation weather resource. Aviation weather resource-type objects exists at a lower level
in the aviation weather hierarchy. (2) The Aviation Weather Resource class refers only to
the aviation weather functionality of the ATC system.
**specification**  Aviation Weather System - a system of wind and weather products available to users of the ATC system and the management of those products in the creation of specialized products.

**structure**

*contains parts*  Wind/Weather Product;  Aviation Weather System Manager

*instance connection*  None

*message connection*  None

*object states*  None

*attribute*  Area of Coverage - the geographic area covered by reports.

*services*  The Aviation Weather System class performs the implicit services create, connect, and release.  An Aviation Weather System object performs the implicit service access.

*external input*  None

*external output*  None

*notes*  None
**specification**  Aviation Weather System Manager - a position (person or automation) that prepares standard wind and weather reports and creates specialized reports at the request of a manager-type object.

**structure**

*part of*  Aviation Weather System

**instance connection**  None

**message connections**

*message*  Request Special Report, from manager-type object

**object states**  None

**attribute**  None

**services**  The Aviation Weather System Manager class performs the implicit services create, connect, and release. An Aviation Weather System Manager object performs the implicit service access.

Create Specialized Report

- Upon request by a manager-type object, create a specialized wind/weather report.

**external input**  The Aviation Weather System Manager is external to the ATC system.

**external output**  None

**notes**  An Aviation Weather System Manager object (e.g., a meteorologist) is not properly within the ATC model (i.e., is not an ATC manager), but is included for convenience.
Clearance - an authorization by ATC, for the purpose of preventing collisions between known aircraft, for an aircraft to proceed under specified traffic conditions within controlled airspace. (AIM)

structure

generalization of Flight Clearance; Ground Clearance; Departure Clearance; En Route Clearance; Arrival Clearance

instance connection None

message connection None

object states None

attribute None

services

service Amend Clearance

• Determine the needed change to the clearance and carry it out

external input None

external output None

notes The Clearance class was created as a convenience. In this model, the clearance associated with a flight or with a vehicle is the accumulation from the present time of all outstanding clearance instructions (those not yet carried out, and those in the process of being carried out).
**specification**  Communications Resource - the communications media provided by the ATC system for the transfer of information between the ATC system and aircraft (e.g., the transfer of a clearance amendment by radio from a human specialist to an aircraft) and among ATC system components, and the management of the information.

**structure**

specialization of  Resource

contains parts  Communications System

instance connection  None

message connection  None

object states  None

attribute  None

services  The Communications Resource class performs the implicit services create, connect, and release. A Communications Resource object performs the implicit service access.

external input  None

external output  None

**notes**  (1) The Communications Resource class exists only for convenience as the highest-level communications resource. Communications resource-type objects exists at a lower level in the communications hierarchy. (2) The Communications Resource class refers only to the communications functionality of the ATC system.
specification  Communications System - a system of communications media provided by the ATC system for the transfer of information between the ATC system and aircraft and among ATC system components, and the management of the information.

structure

  part of  Communications Resource

  generalization of  Interfacility Communications System; Intrafacility Communications System; Air/Ground Communications System

instance connection  Communications System (0, m) <-> Vehicle Communications System (0, m)

message connections

  message  Request Accept Message, to manager-type object

  message  Request Transmit Message, from manager-type object

  message  Request Transmit Message, from Vehicle Communications System object

  message  Request Transmit Message, to Vehicle Communications System object

object states  None

attributes

  attribute  Area of Coverage - the range of the communications equipment.

  attribute  Equipment Type - the type of communications equipment provided by the communications system.

  attribute  Status - operational or not operational.

services  The Communications System class performs the implicit services create, connect, and release. A Communications System object performs the implicit service access.

  service  connect

  •  Upon determination that a vehicle is within the Area of Coverage of the Communications System object, create an instance connection between the Communications System object and the Vehicle Communications System object which is part of the Aircraft or Ground Vehicle object.
• Upon determination that a vehicle is no longer within the Area of Coverage of the Communications System object, remove the instance connection between the Communications System object and the Vehicle Communications System object which is part of the Aircraft or Ground Vehicle object.

_**service**_ Transmit Message

• Upon request, transmit the provided message to the identified recipient.

_**external input**_ None

_**external output**_ None

_**notes**_ Instance connections and message connections exist almost exclusively for objects of specializations of communications system classes (not for objects of the Communications System class itself).
specification  Controlled Airspace - airspace designated as a control zone, airport radar service area, terminal control area, transition area, control area, continental control area, and positive control area within which some or all aircraft may be subject to ATC. (AIM)

structure

part of  Airspace/Ground Resource

contains parts  Oceanic Controlled Airspace;  En Route Controlled Airspace;  Terminal Controlled Airspace

instance connections

instance connection  Controlled Airspace (1) <-> Flight (0, m)

instance connection  Controlled Airspace (1) <-> Aircraft (0, m)

instance connection  Controlled Airspace (1) <-> Flight Manager (0, m)

instance connection  Controlled Airspace (1) <-> Air Traffic Manager (0, m)

instance connection  Controlled Airspace (1) <-> Traffic Manager (0, m)

message connection  Intrinsic only

object states  None

attributes

attribute  Capacity (see Airspace/Ground Resource)

attribute  Configuration (see Airspace/Ground Resource)

attribute  Demand (see Airspace/Ground Resource)

attribute  Load (see Airspace/Ground Resource)

attribute  Location (see Airspace/Ground Resource)

attribute  Name (see Airspace/Ground Resource)

attribute  Saturation Threshold (see Airspace/Ground Resource)

attribute  Separation Minima (see Airspace/Ground Resource)

attribute  Usage Restrictions (see Airspace/Ground Resource)
services The Controlled Airspace class performs the implicit services create, connect, and release. A Controlled Airspace object performs the implicit service access.

service Update Demand

• Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service Update Load

• Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input None

external output None

notes None
Departure Clearance - the clearances issued to a flight, and not yet completely carried out, for the departure phase of flight. The departure clearance, also known as departure instructions, describes the route (e.g., standard instrument departure [SID] route), headings, or radar vectors that the aircraft must follow from the airport to the beginning of the route or to join the route in the en route clearance.

**structure**

*specialization of* Clearance

*part of* Flight Clearance

*instance connection* None

*message connections*

*message* Request Amend Clearance, sent from a Pilot object

*object states* None

*attributes*

*attribute* Departure Procedure - the route (e.g., SID), heading, or radar vectors that the aircraft must follow from the airport to the beginning of the route or to join the route in the en route clearance.

*attribute* Release Time - the earliest time the flight may depart. (AIM)

*attribute* Void Time - clearance void time - the specific time after which, if the flight has not departed, the clearance is void. (AIM)

*services* The Departure Clearance class performs the implicit services create, connect, and release. A Departure Clearance object performs the implicit service access.

*service* Amend Clearance

- Determine the affected attributes of the Departure Clearance, and the needed change.
- Change the identified attributes to the new values.

*external input* None

*external output* None
The departure clearance, for this model, includes the takeoff clearance, which represents an allocation of an airspace-type resource. The Release Time and Void Time attributes have values only when the flight originates at an airport not served by a control tower.
specification  En Route Clearance - the clearances issued to a flight, and not yet completely carried out, for the en route phase of flight.

structure

specialization of  Clearance

part of  Flight Clearance

instance connection  None

message connections

message  Request Amend Clearance, sent from a Pilot object

object states  None

attributes

attribute  Altitude Profile - an altitude or sequence of altitudes assigned to the flight.

attribute  Clearance Limit - the farthest point along the route to which the flight is cleared.

attribute  Holding Instructions - special instructions to be followed when a holding pattern area is to be used; this attribute has a value only when holding is required by ATC.

attribute  Route - the route of flight approved for use by the flight; the approved route might terminate at a destination airport or at a clearance limit (which see).

attribute  Speed Schedule - a speed or sequence of speeds assigned to a flight.

services  The En Route Clearance class performs the implicit services create, connect, and release. An En Route Clearance object performs the implicit service access.

service  Amend Clearance

•  Determine the affected attributes of the En Route Clearance, and the needed change.

•  Change the values of the identified attributes to the new values.

external input  None

external output  None

notes  None
**specification**  En Route Controlled Airspace - that part of controlled airspace used primarily for the en route phase of flights over land

**structure**

*part of*  Controlled Airspace

*contains parts*  Route; Airspace Volume

**instance connections**

*instance connection*  En Route Controlled Airspace (1) <-> Flight (0, m)

*instance connection*  En Route Controlled Airspace (1) <-> Aircraft (0, m)

*instance connection*  En Route Controlled Airspace (1) <-> Flight Manager (0, m)

*instance connection*  En Route Controlled Airspace (1) <-> Air Traffic Manager (0, m)

*instance connection*  En Route Controlled Airspace (1) <-> Traffic Manager (0, m)

**message connection**  Intrinsic only

**object states**  None

**attributes**

*attribute*  Capacity (see Airspace/Ground Resource)

*attribute*  Configuration (see Airspace/Ground Resource)

*attribute*  Demand (see Airspace/Ground Resource)

*attribute*  Load (see Airspace/Ground Resource)

*attribute*  Location (see Airspace/Ground Resource)

*attribute*  Name (see Airspace/Ground Resource)

*attribute*  Saturation Threshold (see Airspace/Ground Resource)

*attribute*  Separation Minima (see Airspace/Ground Resource)

*attribute*  Usage Restrictions (see Airspace/Ground Resource)
services  The En Route Controlled Airspace class performs the implicit services create, connect, and release. An En Route Controlled Airspace object performs the implicit service access.

service  Update Demand

• Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service  Update Load

• Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input  None

external output  None

notes  A Route object and an Airspace Volume object which are parts of the same En Route Controlled Airspace object might define parts of the same physical airspace, but both objects reflect common usage.
**specification**  Fix/Waypoint - a fix or a waypoint. A fix is a geographical position determined by visual reference to the surface, by reference to one or more radio NAVAIDs, by celestial plotting, or by another navigational device. A waypoint is a predetermined geographical position used for route/instrument approach definition, or progress reporting purposes, that is defined relative to a VORTAC station or in terms of latitude/longitude coordinates. (AIM)

**structure**

*part of*  Track; Oceanic Route; Route; Terminal Route

**instance connections**

*instance connection*  Fix/Waypoint (0, m) <-> Flight (0, m)

*instance connection*  Fix/Waypoint (0, m) <-> Aircraft (0, m)

*instance connection*  Fix/Waypoint (0, m) <-> Flight Manager (0, m)

*instance connection*  Fix/Waypoint (0, m) <-> Air Traffic Manager (0, m)

*instance connection*  Fix/Waypoint (0, m) <-> Traffic Manager (0, m)

**message connection**  Intrinsic only

**object states**  None

**attributes**

*attribute*  Capacity  (see Airspace/Ground Resource)

*attribute*  Configuration  (see Airspace/Ground Resource)

*attribute*  Demand  (see Airspace/Ground Resource)

*attribute*  Load  (see Airspace/Ground Resource)

*attribute*  Location  (see Airspace/Ground Resource)

*attribute*  Name  (see Airspace/Ground Resource)

*attribute*  Saturation Threshold  (see Airspace/Ground Resource)

*attribute*  Separation Minima  (see Airspace/Ground Resource)

*attribute*  Service Volume - for a NAVAID, the area of coverage
attribute  Type - the type of fix or waypoint (e.g., VOR, latitude/longitude)

attribute  Usage Restrictions  (see Airspace/Ground Resource)

services  The Fix/Waypoint class performs the implicit services create, connect, and release. A Fix/Waypoint object performs the implicit service access.

service  Update Demand
•  Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service  Update Load
•  Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input  None

external output  None

notes  None
**specification**  Flight - the operation of an aircraft on the surface or in the air; a flight can have three logical components, which are a flight plan, a flight clearance, and one or more aircraft.

**structure**

*part of*  Air Traffic

*contains parts*  Flight Clearance; Flight Plan; Aircraft

**instance connections**

*instance connection*  Flight (0, m) <-> Flight Manager (0, 1)

*instance connection*  Flight (0, m) <-> airspace-type (0, m)

*instance connection*  Flight (0, m) <-> Flight (0, m)

**message connections**  None

**object states**  A Flight object has the following states: predeparture, departure, en route, arrival, and postarrival. The state changes are shown in figure 3-10. The following short description is copied from section 3.2.6.2 and applies to the usual flight; some details differ for particular flights (e.g., a VFR flight without a flight plan, a flight that files its flight plan after it is airborne). The predeparture state corresponds to that time before the aircraft makes use of the airport movement area of an airport; during this time period, the initial flight planning, including the receipt of weather briefings and the filing of the flight plan, is completed. The departure state corresponds to that time spent on the airport movement area for taxiing and takeoff and in the air until handed off to an en route controller. (There is no en route state for a tower en route flight.) The en route state lasts until the flight is given the arrival clearance for its destination airport. The arrival state lasts until the aircraft leaves the airport movement area of the destination airport. The postarrival state exists essentially to allow for flight plan closing (although an IFR flight plan is considered closed upon arrival at a controlled airport). The states of a flight are defined to correspond to the assignment of the manager rather than to the airspace/ground resource used (although this is a fine point, because the managers and the airspace/ground resources are also closely related). That is, a change in flight state is associated with a change in manager rather than with a change in resource usage.

**attributes**

*attribute*  Trajectory - the planned four-dimensional path of the aircraft.

*attribute*  Type - IFR or VFR or composite VFR/IFR.

**services**  The Flight class performs the implicit services create, connect, and release. A Flight object performs the implicit service access.
**service** Maintain Trajectory

- Ensure that the prediction of the four-dimensional path of the aircraft is kept up-to-date.

**external input** None

**external output** None

**notes**

1. For this model, a flight is synonymous with a flight segment: a departure by an aircraft after it has arrived at an airport is considered part of another flight. However, two or more Flight objects can be connected by instance connections to reflect a continuing flight. (2) When an aircraft is on the ground, it can be part of a flight and part of a vehicle.

3. During the predeparture state of a Flight object, an instance connection exists first between the Flight object and the Flight Manager object (e.g., a Flight Service Station preflight briefing specialist) with which initial flight planning is carried out. After the initial flight planning, and before the aircraft has received any flight clearance, an instance connection exists between the Flight object and the first Flight Manager object that can deliver any part of the departure or en route clearance (e.g., the clearance delivery position). After this point, the instance connection is established and broken as the aircraft is "handed off" from one flight manager (e.g., controller) to another. Finally, after the arrival state of the Flight object, an instance connection exists between the Flight object and the Flight Manager object that can assist in closing the flight plan (note that this flight manager could be another FSS specialist for a VFR flight or the local controller for an IFR flight, using the Flight Manager object's service Assist in Flight Planning). (4) Instance connections between a Flight object and an airspace/ground resource-type object exist based upon the planned use of the resource by the flight as reflected in the flight plan. Each instance connection is created when the flight plan is created, changed as the flight plan is changed, and deleted once the resource has been used by the flight. Each Flight object having an instance connection with a resource-type object contributes to the value of the Demand attribute of that resource.
**specification**  
Flight Clearance - the aggregate of the clearances issued to a flight, and not yet completely carried out, for the departure, en route, and arrival phases of flight; a class created for convenience, without a direct analogue in the ATC system

**structure**

- **specialization of**  Clearance
- **part of**  Flight
- **contains parts**  Departure Clearance; En Route Clearance; Approach Clearance

**instance connection**  None

**message connections**  None

**object states**  None

**attribute**  Flight Identification - an ATC-approved identification; could have the same value as the Aircraft Identification attribute of an Aircraft object which is part of the Flight object.

**services**  The Flight Clearance class performs the implicit services create, connect, and release. A Flight Clearance object performs the implicit service access.

**external input**  None

**external output**  None

**notes**  None
specification  Flight Manager - one who plans or supervises flights, including approving resource usage by individual flights.

structure

specialization of  Manager

instance connections

instance connection  Flight Manager (0, 1) <-> Flight (0, m)

instance connection  Flight Manager (0, m) <-> airspace-type resource (0, m)

instance connection  Flight Manager (0, m) <-> Air/Ground Communications System (0, m)

instance connection  Flight Manager (0, m) <-> Interfacility Communications System (0, m)

instance connection  Flight Manager (0, m) <-> Intrafacility Communications System (0, m)

message connections

message  Request Accept Control Responsibility, sent from Flight Manager object

message  Request Accept Control Responsibility, sent to Flight Manager object

message  Request Adherence to Clearance, sent to Pilot object

message  Request Aircraft Altitude, sent to Pilot object

message  Request Aircraft Identification, sent to Pilot object

message  Request Aircraft Position, sent to Pilot object

message  Request Amend Flight Plan & Clearance, sent to Pilot object

message  Request Flight Plan Amendment, sent from Pilot object

message  Request IFR Flight Separation, sent from Flight Manager object

message  Request IFR Flight Separation, sent to Flight Manager object

message  Request Preflight Briefing, sent from Pilot object

© 2000 The MITRE Corporation. ALL RIGHTS RESERVED.
message Request Resource, sent from Flight Manager object

message Request Resource, sent to Flight Manager object

message Request Transfer of Communications, sent to Pilot object

object states None

attribute Area of Responsibility (Inherited)

services The Flight Manager class performs the implicit services create, connect, and release. A Flight Manager object performs the implicit service access.

service Assist in Flight Planning
• Upon request, provide or update preflight briefing.
• Upon request, evaluate and accept flight plan.
• Upon request, evaluate and accept flight plan modifications.
• Open flight plan (automatically, or upon request).
• Close flight plan (automatically, or upon request).

service Assist in Weather Avoidance
• Collect (solicit and receive) weather reports from pilots.
• Upon request, provide weather advisories.

service Detect, Predict & Report Restriction Violation
• Determine whether a violation of a usage restriction for an airspace-type resource is happening or is planned to happen and notify the appropriate manager-type object.

service Detect, Predict & Report Separation Violation
• Determine whether a violation of separation minima (i.e., an aircraft-to-aircraft conflict or an aircraft-to-airspace conflict) is happening or is planned to happen and notify the appropriate manager-type object.

service Direct Transfer of Communications
• Inform a flight (pilot) of the frequency that should be used in future communications.
service Ensure Restriction Compliance

- When requested by a flight to use an airspace-type resource, determine whether a usage restriction exists for the resource and approve or disapprove the use of the resource, as appropriate.

- When it is discovered that a flight has planned to use an airspace-type resource contrary to usage restrictions, determine whether and how to change the resource usage of that flight.

service Generate & Deliver Clearance

- Translate instructions to a flight into clearance language and deliver via a communications medium.

service Separate IFR Flights

- When requested by a flight to change its flight plan, determine whether a problem (i.e., an aircraft-to-aircraft conflict or an aircraft-to-airspace conflict) would exist based upon the revised plan and approve or disapprove the plan, as appropriate.

- When it is discovered that following an existing flight plan would cause a problem, determine whether and how to change the plan.

service Transfer & Accept Control Responsibility

- When controlling a flight: determine the next appropriate flight manager and time of control transfer and offer control to that flight manager.

- When offered control of a flight: evaluate the appropriateness of receiving control of the flight and either accept or deny control transfer.

external input None

external output None

notes (1) The Flight Manager class does not represent a currently-staffed position; instead, it includes parts of the functionality of several ATC positions. (2) The services of a Flight Manager object might be performed either by humans or by automation, or shared.
specification  Flight Plan - Specified information relating to the intended flight of an aircraft that is filed orally or in writing with an FSS or an ATC facility, as amended.

structure

    part of  Flight

instance connections  None

message connections

message Amend Flight Plan, sent from Pilot object

object states  An IFR Flight Plan object has two states, open and closed (also called active and inactive). A flight plan is closed when the aircraft is on the ground before takeoff. An IFR flight plan is opened after takeoff as soon as the aircraft is detected by radar, or (in a nonradar environment) as soon as the aircraft reports entering controlled airspace. A VFR flight plan is activated as soon as the aircraft is airborne and has informed the Flight Service Station of the actual time of departure. An IFR flight plan is closed when the aircraft lands at an airport with a control tower, or (when there is no control tower) the aircraft notifies ATC or the Flight Service Station of the landing, or when the flight cancels its IFR flight plan. A VFR flight plan is closed when the aircraft so notifies the destination Flight Service Station. When a VFR flight plan is not activated within one hour of the proposed departure time, it is cancelled.

attributes  As required for FAA Form 7233-1 (Flight Plan). The values of the attributes reflect all amendments since flight plan filing.

services  The Flight Plan class performs the implicit services create, connect, and release. A Flight Plan object performs the implicit service access.

    service  Amend Flight Plan

    •  Determine the affected attributes of the Flight Plan, and the needed change

    •  Change the identified attributes to the new values

external input  Airline Dispatch Offices; Military Base Operations; DUAT

external output  Airline Dispatch Offices; Military Base Operations; DUAT

notes  The Flight Plan class represents the flight plan commonly known in ATC, as amended throughout the flight.
specification  Ground Clearance - the aggregate of the clearances issued to a ground vehicle, and not yet completely carried out, for movement on the airport movement area; a class created for convenience, without a direct analogue in the ATC system movement area

structure

specialization of  Clearance

part of  Vehicle

instance connection  None

message connections

message  Request Amend Ground Clearance, sent from the Ground Vehicle Operator object

object states  None

attribute  Ground Vehicle Identification - an ATC-approved identification of the Ground Vehicle object of which the Ground Clearance object is a part; could have the same value as the Ground Vehicle Identification attribute of the Ground Vehicle object which is also part of the Vehicle object.

attribute  Route - the route approved for use in traversing the airport movement area.

services  The Ground Clearance class performs the implicit services create, connect, and release. A Ground Clearance object performs the implicit service access.

service  Amend Clearance

• Determine the needed change to the Route attribute and carry out the change

external input  None

external output  None

notes  None
specification  Ground Surveillance System - a system for the tracking of aircraft and ground vehicle positions on an airport movement area and the management of the information provided.

structure

specialization of  Surveillance System

instance connection  Ground Surveillance System (0, m) <-> Vehicle Surveillance System (0, m)

message connections

message  Transmit & Receive Radar Pulse, sent from Ground Surveillance System

message  Transmit & Receive Radar Pulse, sent to Ground Surveillance System

object states  None

attributes

attribute  Area of Coverage (Inherited)

attribute  Equipment Type (Inherited)

attribute  Status (Inherited)

services  The Ground Surveillance System class performs the implicit services create, connect, and release. A Ground Surveillance System object performs the implicit service access.

service  Locate, Identify & Report Vehicle

- Locate each aircraft and ground vehicle in the area of coverage, determine its identification, and report its identification and location.

external input  None

external output  None

notes  In reality, a ground surveillance system might not be able to locate and identify all aircraft or ground vehicles in its area of coverage, due in part to interference from other objects and in part to capacity limitations of the surveillance system.
**specification**  Ground Traffic - a grouping of vehicles determined by a defined criterion or set of criteria.

**structure**

- **specialization of** Traffic

- **contains part** Vehicle

**instance connection**  Ground Traffic <-> Ground Traffic Manager (0, m)

**message connection**  None

**object states**  None

**attribute**  Selection Criteria (inherited from Traffic)

**services**  The Ground Traffic class performs the implicit services create, connect, and release. A Ground Traffic object performs the implicit service access.

- **service connect**
  - When requested by a Ground Traffic Manager object, create an object with identified Vehicle objects as parts

**external input**  None

**external output**  None

**notes**  The definition of ground traffic is unique to this model. The Ground Traffic class exists in this model to allow the creation of well-defined groupings of vehicles for specific purposes (e.g., all vehicles needing to cross runway 14 to get to runway 9 in the next half hour).
**specification**  Ground Traffic Manager - one who plans or supervises ground traffic.

**structure**

**specialization of**  Traffic Manager

**instance connections**

**instance connection**  Ground Traffic Manager (1) <-> Ground Traffic (0, m)

**instance connection**  Ground Traffic Manager (0, m) <-> ground-type resource (0, m)

**instance connection**  System (0, m) <-> Air/Ground Communications

**instance connection**  System (0, m) <-> Interfacility Communications

**instance connection**  System (0, m) <-> Intrafacility Communications

**message connections**

**message**  Create Ground Traffic Object, sent to Ground Traffic class service "create"

**message**  Request Usage Restrictions, sent from Air Traffic Manager object

**message**  Request Usage Restrictions, sent from Ground Traffic Manager object

**message**  Request Usage Restrictions, sent to Air Traffic Manager object

**message**  Request Usage Restrictions, sent to Ground Traffic Manager object

**object states**  None

**attribute**  Area of Responsibility (inherited)

**services**  The Ground Traffic Manager class performs the implicit services create, connect, and release. A Ground Traffic Manager object performs the implicit service access.

**service**  Detect, Predict & Report Saturation

- Determine whether saturation of a ground-type resource is happening (based on the Load attribute) or is planned to happen (based on the Demand attribute) and notify the appropriate manager-type object.
service  Determine Ground Capacity

• Determine the number of vehicles that can be accommodated by an ground resource-type resource per unit time or instantaneously.

service  Resolve Ground Saturation Problem

• Develop directives in response to present and predicted vehicle flow problems.

service  Set Ground Saturation Threshold

• Based upon the ground resource-type capacity and other factors, determine a utilization value which, once achieved, should cause an alert.

external input  None

external output  None

notes  (1) The Ground Traffic Manager class does not represent a currently-staffed position; instead, it includes parts of the functionality of positions such as the ATCT positions that determine ground capacity.  (2) The services of a Ground Traffic Manager object might be performed either by humans or by automation.
specification  Ground Vehicle - a vehicle, other than an aircraft, that traverses a part of the airport movement area in the performance of its duties (e.g., snow removal vehicle)

structure

part of  Vehicle

contains parts  Ground Vehicle Operator; Vehicle Surveillance System; Vehicle Navigation System; Vehicle Communications System

instance connection  None

message connection  Create Vehicle Object, sent to Vehicle class

object states  A Ground Vehicle object has two states, "active" and "inactive," depending on whether the ground vehicle is on or off of the airport movement area, respectively.

attributes

attribute  Ground Vehicle Identification - an ATC-approved identification; might be the radio call sign.

attribute  Location - the position of the ground vehicle as known by ATC, whether determined by ATC or reported to ATC by the ground vehicle itself and recorded by ATC. Location includes lateral dimension and time.

services  The Ground Vehicle class performs the implicit services create, connect, and release. A Ground Vehicle object performs the implicit service access.

service create

•  Create and initialize, as specified for the implicit service create.

•  When a Ground Vehicle object is created and there does not already exist a Vehicle object of which the Ground Vehicle object is to be a part, requests the Vehicle class to create a Vehicle object of which the Ground Vehicle object is a part.

external input  None

external output  None

notes  Instance connections occur higher (i.e., between a Vehicle object and another object) and lower (i.e., between a part object and another object) than a Ground Vehicle object in the structure.
specification  Ground Vehicle Operator - the operator of a ground vehicle; the pilot when an aircraft is still on the ground

structure

part of  Ground Vehicle

instance connection  None

message connection

message  Message Available, from a Vehicle Communications System object that is part of the same Ground Vehicle object as the Ground Vehicle Operator object

message  Perform Navigation, to the Vehicle Navigation System object

message  Request Adherence to Clearance, from a Vehicle Manager object

message  Request Amend Clearance, to the Ground Clearance object which is part of the Vehicle object

message  Request Amend Ground Vehicle Route, to the Ground Vehicle Route object which is part of the Vehicle object

message  Request Amend Ground Vehicle Route & Ground Clearance, from a Vehicle Manager object

message  Request Create Ground Vehicle Object, to the Ground Vehicle class

message  Request Ground Vehicle Route Amendment, to a Vehicle Manager object

message  Request Report Ground Vehicle Position, from a Vehicle Manager object

message  Request Transfer of Communications, from a Vehicle Manager object

message  Request Vehicle Position, sent to the Vehicle Navigation System object which is part of the same Ground Vehicle object as the Ground Vehicle Operator object

message  Send Message, to a Vehicle Communications System object that is part of the same Ground Vehicle object as the Ground Vehicle Operator object

message  Send Vehicle Identification, sent to the Vehicle Surveillance System object which is part of the same Ground Vehicle object as the Ground Vehicle Operator object

message  Set Frequency, sent to the Vehicle Communications System object which is part of the same Ground Vehicle object as the Ground Vehicle Operator object
object states None

attribute Position - the driver or pilot

services The Ground Vehicle Operator class performs the implicit services create, connect, and release. A Ground Vehicle Operator object performs the implicit service access.

service create
• Create and initialize, as specified for the implicit service create.
• When a Ground Vehicle Operator object is created and there does not already exist an Ground Vehicle object of which the Ground Vehicle Operator object is to be a part, requests the Ground Vehicle class to create an Ground Vehicle object of which the Ground Vehicle Operator object is a part.

service Avoid Other Ground Traffic
• Ensure that other ground traffic is avoided, whether based on personal observation or the direction of a vehicle manager.

service Execute Movement
• Perform the duties necessary to carry out the movement on the airport movement area, including (for example)
  – Ensuring that the aircraft or ground vehicle follows the planned ground plan.
  – Ensuring that clearances and instructions are carried out.

service Manage Equipment
• Perform the duties necessary to ensure the proper performance of the Vehicle Surveillance System object, the Vehicle Navigation System object, and the Vehicle Communications System object.

service Report Ground Vehicle Position
• When requested by a Vehicle Manager object, request a Vehicle Navigation System object which is part of the same Ground Vehicle object to determine the location of the ground vehicle.

service Request Movement
• Coordinate movement on the airport movement area with a vehicle manager.
external input  None

external output  None

notes  (1) The (non-implicit) services of a Ground Vehicle Operator object are performed by a human. (2) The service Manage Equipment is performed, for the most part, independently of other objects, but also responds to external requests, e.g., a manager-type object requesting a particular beacon code setting for the transponder. (3) The service Report Ground Vehicle Position gets information from the Vehicle Navigation System object, not the Vehicle Surveillance System object, which responds independently to ATC’s Surveillance System object.
**specification**  
Ground Vehicle Route - the planned route of movement for the ground vehicle across the airport movement area.

**structure**

*part of*  
Vehicle

**instance connection**  
None

**message connection**

*message*  
Request Amend Ground Vehicle Route, sent from Ground Vehicle Operator object

**object states**  
A Ground Vehicle Route has two states, active and inactive, depending on whether the ground vehicle location is on or off of the airport movement area, respectively. Once the ground vehicle has exited the airport movement area and has completed the planned route, the object can be deleted.

**attribute**  
Ground Route - the planned route on the airport movement area.

**services**  
The Ground Vehicle Route class performs the implicit services create, connect, and release. A Ground Vehicle Route object performs the implicit service access.

*service*  
Amend Ground Vehicle Route

*•*  
Determine the needed change to the Ground Route attribute and carry out the change.

**external input**  
None

**external output**  
None

**notes**  
None
specification  Holding Area - an area on the airport movement area, other than a taxiway or runway, where an aircraft might wait for further instructions.

structure

part of  Airport Movement Area

instance connections

instance connection  Holding Area (0, m) <-> Vehicle (0, m)
instance connection  Holding Area (0, m) <-> Aircraft (0, m)
instance connection  Holding Area (0, m) <-> Vehicle Manager (0, m)
instance connection  Holding Area (0, m) <-> Ground Traffic Manager (0, m)
instance connection  Holding Area (0, m) <-> Traffic Manager (0, m)

object states  None

attributes

attribute  Capacity  (see Airspace/Ground Resource)
attribute  Configuration  (see Airspace/Ground Resource)
attribute  Demand  (see Airspace/Ground Resource)
attribute  Load  (see Airspace/Ground Resource)
attribute  Location  (see Airspace/Ground Resource)
attribute  Name  (see Airspace/Ground Resource)
attribute  Saturation Threshold  (see Airspace/Ground Resource)
attribute  Separation Minima  (see Airspace/Ground Resource)
attribute  Status - active or inactive.
attribute  Surface Condition - the condition of the holding area surface (e.g., covered with ice, snow, or long grass).
attribute  Usage Restrictions  (see Airspace/Ground Resource)
services  The Holding Area class performs the implicit services create, connect, and release. A Holding Area object performs the implicit service access.

service  Update Demand

•  Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service  Update Load

•  Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input  None

external output  None

notes  None
specification  Holding Pattern Airspace - the airspace containing a holding pattern.

structure

specialization of  Airspace Volume

instance connections

instance connection  Holding Pattern Airspace (0, m) <-> Flight (0, m)
instance connection  Holding Pattern Airspace (0, m) <-> Aircraft (0, m)
instance connection  Holding Pattern Airspace (0, m) <-> Flight Manager (0, m)
instance connection  Holding Pattern Airspace (0, m) <-> Air Traffic Manager (0, m)
instance connection  Holding Pattern Airspace (0, m) <-> Traffic Manager (0, m)

message connection  Intrinsic only

object states  None

attributes

attribute  Capacity  (see Airspace/Ground Resource)
attribute  Configuration  (see Airspace/Ground Resource)
attribute  Demand  (see Airspace/Ground Resource)
attribute  Load  (see Airspace/Ground Resource)
attribute  Location  (see Airspace/Ground Resource)
attribute  Name  (see Airspace/Ground Resource)
attribute  Saturation Threshold  (see Airspace/Ground Resource)
attribute  Separation Minima  (see Airspace/Ground Resource)
attribute  Usage Restrictions  (see Airspace/Ground Resource)

services  The Holding Pattern Airspace class performs the implicit services create, connect, and release.  A Holding Pattern Airspace object performs the implicit service access.

service  Update Demand
• Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

**service**  Update Load

• Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

**external input**  None

**external output**  None

**notes**  None
specification  Interfacility Communications System - a system of communications media provided by the ATC system for the transfer of information among ATC specialists in different facilities and the management of the information.

structure

specialization of  Communications System

instance connection  Interfacility Communications System (0, m) <-> manager-type object (0, m)

message connections

message  Request Transmit Message, from manager-type object

message  Request Accept Message, to manager-type object

object states  None

attributes

attribute  Area of Coverage (Inherited)

attribute  Equipment Type (Inherited)

attribute  Status (Inherited)

services  The Interfacility Communications System class performs the implicit services create, connect, and release. An Interfacility Communications System object performs the implicit service access.

service  connect (Inherited)

service  Transmit Message (Inherited)

external input  None

external output  None

notes
**specification**  Intrafacility Communications System - a system of communications media provided by the ATC system for the transfer of information among ATC specialists in the same facility and the management of the information.

**structure**

**specialization of**  Communications System

**instance connection**  Intrafacility Communications System (0, m) <-> manager-type object (0, m)

**message connections**

**message**  Request Accept Message, to manager-type object

**message**  Request Transmit Message, from manager-type object

**object states**  None

**attributes**

**attribute**  Area of Coverage (Inherited)

**attribute**  Equipment Type (Inherited)

**attribute**  Status (Inherited)

**services**  The Intrafacility Communications System class performs the implicit services create, connect, and release. An Intrafacility Communications System object performs the implicit service access.

**service**  connect (Inherited)

**service**  Transmit Message (Inherited)

**external input**  None

**external output**  None

**notes**  None
specification  Jet Route -- a route designed to serve aircraft operations from 18,000 feet MSL up to and including flight level 450. The routes are referred to as ”J” routes with numbering to identify the designated route; e.g., J105. (AIM)

structure

specialization of  Route

contains parts (inherited)  Fix/Waypoint; Route Segment

instance connections

instance connection  Jet Route (0, m) <-> Flight (0, m)
instance connection  Jet Route (0, m) <-> Aircraft (0, m)
instance connection  Jet Route (0, m) <-> Flight Manager (0, m)
instance connection  Jet Route (0, m) <-> Air Traffic Manager (0, m)
instance connection  Jet Route (0, m) <-> Traffic Manager (0, m)

message connection  Intrinsic only

object states  None

attributes

attribute  Capacity (Inherited)
attribute  Configuration (Inherited)
attribute  Demand (Inherited)
attribute  Load (Inherited)
attribute  Location (Inherited)
attribute  Name (Inherited)
attribute  Saturation Threshold (Inherited)
attribute  Separation Minima (Inherited)
attribute  Usage Restrictions (Inherited)
services  The Jet Route class performs the implicit services create, connect, and release. A Jet Route object performs the implicit service access.

service  Update Demand

•  Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service  Update Load

•  Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input  None

external output  None

notes  None
specification  Landing Navigation System- a system of navigation aids to assist aircraft in approach navigation and the management of the information provided.

structure

specialization of  Navigation System

instance connection  Landing Navigation System (0, m) <-> Vehicle Navigation System (0, m)

message connections

message  Offer Altitude Information, to Vehicle Navigation System

message  Offer Azimuth Information, to Vehicle Navigation System

message  Request Distance Information, from Vehicle Navigation System

object states  None

attributes

attribute  Identifier Code (Inherited)

attribute  Frequency Channels (Inherited)

attribute  Location  (Inherited)

attribute  NAVAID Type (Inherited)

attribute  Area of Coverage (Inherited)

attribute  Status (Inherited)

services  The Landing Navigation System class performs the implicit services create, connect, and release.  A Landing Navigation System object performs the implicit service access.

service  connect (Inherited)

service  Transmit Azimuth Information (Inherited)

service  Transmit Elevation Information

•  As scheduled, transmit information with which a receiving aircraft can determine its elevation during an approach

service  Transmit NAVAID Identifier (Inherited)
service  Transmit Range Information (Inherited)

external input  None

external output  None

notes  The services Transmit Range Information, Transmit Azimuth Information, and Transmit Elevation Information may not be possible for some landing navigation systems; some specializations of landing navigation systems can provide additional services.
**specification**  Manager - one who handles or directs with a degree of skill; one who alters by manipulation; a supervisor, executor, or administrator.

**structure**

*generalization of*  Traffic Manager; Flight Manager; Vehicle Manager

**instance connection**  None

**message connection**  None

**object states**  None

**attribute**  Area of Responsibility - the users or resources over which the manager has management responsibilities

**services**  None

**external input**  None

**external output**  None

**notes**  The Manager class exists as a convenience; any object in the manager hierarchy is usually an object of a class lower in the user hierarchy than the Manager class.
specification  Military Training Route - airspace of defined vertical and lateral dimensions established for the conduct of military flights training at airspeeds in excess of 250 knots IAS.

structure

specialization of  Route

contains parts (inherited)  Fix/Waypoint; Route Segment

instance connections

instance connection  Military Training Route (0, m) <-> Flight (0, m)
instance connection  Military Training Route (0, m) <-> Aircraft (0, m)
instance connection  Military Training Route (0, m) <-> Flight Manager (0, m)
instance connection  Military Training Route (0, m) <-> Air Traffic Manager (0, m)
instance connection  Military Training Route (0, m) <-> Traffic Manager (0, m)

message connection  Intrinsic only

object states  None

attributes

attribute  Capacity (Inherited)
attribute  Configuration (Inherited)
attribute  Demand (Inherited)
attribute  Load (Inherited)
attribute  Location (Inherited)
attribute  Name (Inherited)
attribute  Saturation Threshold (Inherited)
attribute  Separation Minima (Inherited)
attribute  Usage Restrictions (Inherited)
services  The Military Training Route class performs the implicit services create, connect, and release. A Military Training Route object performs the implicit service access.

service  Update Demand

  • Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service  Update Load

  • Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input  None

external output  None

notes  None
**specification**  Navigation Resource - navigation aids provided by the ATC system to assist aircraft in navigating between points and the management of the information provided (e.g., the broadcasting of a NAVAID position by radio signal).

**structure**

* specializations of Resource
* contains parts Navigation System

**instance connection**  None

**message connection**  None

**object states**  None

**attribute**  None

**services**  The Navigation Resource class performs the implicit services create, connect, and release. A Navigation Resource object performs the implicit service access.

**external input**  None

**external output**  None

**notes**  (1) The Navigation Resource class exists only for convenience as the highest-level navigation resource. Navigation resource-type objects typically exist at a lower level in the navigation hierarchy. (2) The Navigation Resource class refers only to the navigation functionality of the ATC system, i.e., only those navigation systems certified by the Federal Aviation Administration to be primary navigation systems.
specification  
A system of navigation aids to assist aircraft in navigating between points and the management of the information provided.

structure  

part of  Navigation Resource 

generalization of  Landing Navigation System  

instance connection  Navigation System (0, m) <-> Vehicle Navigation System (0, m)  

message connections  

message  Request Distance Information, from Vehicle Navigation System  

message  Offer Azimuth Information, to Vehicle Navigation System  

object states  None  

attributes  

attribute  Area of Coverage - the range of the navigation equipment  

attribute  Frequency Channels - the frequencies used by the NAVAID to send and to receive signals (e.g., a four or five digit frequency for a VOR station)  

attribute  Identifier Code - the identification of the ground station associated with the navigation system (e.g., a three-letter identifier for a VOR station)  

attribute  Location - the geographical position of the ground station associated with the navigation system  

attribute  NAVAID Type - the type of navigation aid provided by the navigation system (e.g., VOR, VOR/DME, LORAN-C)  

attribute  Status - operational, not operational, or test  

services  
The Navigation System class performs the implicit services create, connect, and release. A Navigation System object performs the implicit service access.

service  connect  

•  Upon determination that an aircraft is within the Area of Coverage of the Navigation System object, create an instance connection between the Navigation System object and the Vehicle Navigation System object which is part of the Aircraft or Ground Vehicle object.
• Upon determination that an aircraft is no longer within the Area of Coverage of the Navigation System object, remove the instance connection between the Navigation System object and the Vehicle Navigation System object which is part of the Aircraft object.

**service** Transmit Azimuth Information

• As scheduled, transmit information with which a receiving aircraft can determine the direction of the NAVAID from the aircraft

**service** Transmit NAVAID Identifier

• Continuously transmit the identifier code on the assigned frequency

**service** Transmit Range Information

• As scheduled, transmit information with which a receiving aircraft can determine the distance from the aircraft to the NAVAID

**external input** None

**external output** None

**notes** The services Transmit Range Information and Transmit Azimuth Information might not be possible for some navigation systems; some specializations of navigation systems can provide additional services.
**specification**  Oceanic Airway - a control area or portion thereof established in the form of a corridor in oceanic airspace, the centerline of which is defined by radio navigational aids or in terms of latitude/longitude coordinates.

**structure**

*part of* Oceanic Controlled Airspace

*generalization of* Track; Oceanic Route

**instance connections**

*instance connection*  Oceanic Airway (0, m) <-> Flight (0, m)

*instance connection*  Oceanic Airway (0, m) <-> Aircraft (0, m)

*instance connection*  Oceanic Airway (0, m) <-> Flight Manager (0, m)

*instance connection*  Oceanic Airway (0, m) <-> Air Traffic Manager (0, m)

*instance connection*  Oceanic Airway (0, m) <-> Traffic Manager (0, m)

**message connection**  Intrinsic only

**object states**  None

**attributes**

*attribute*  Capacity  (see Airspace/Ground Resource)

*attribute*  Configuration  (see Airspace/Ground Resource)

*attribute*  Demand  (see Airspace/Ground Resource)

*attribute*  Load  (see Airspace/Ground Resource)

*attribute*  Location  (see Airspace/Ground Resource)

*attribute*  Name  (see Airspace/Ground Resource)

*attribute*  Primary Direction - the expected direction of flight (e.g., east or west)

*attribute*  Saturation Threshold  (see Airspace/Ground Resource)

*attribute*  Separation Minima  (see Airspace/Ground Resource)
**attribute** Time Period - the time during which the oceanic airway is in effect as defined

**attribute** Usage Restrictions (see Airspace/Ground Resource)

**services** The Oceanic Airway class performs the implicit services create, connect, and release. An Oceanic Airway object performs the implicit service access.

**service** Update Demand

- Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

**service** Update Load

- Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

**external input** None

**external output** None

**notes** None
specification  Oceanic Controlled Airspace - that part of controlled airspace used primarily for the en route phase of flights over (a large body of) water, such as over an ocean.

structure

    part of  Controlled Airspace

    contains parts  Oceanic Airway

instance connections

    instance connection  Oceanic Controlled Airspace (0, m) <-> Flight (0, m)

    instance connection  Oceanic Controlled Airspace (0, m) <-> Aircraft (0, m)

    instance connection  Oceanic Controlled Airspace (0, m) <-> Flight Manager (0, m)

    instance connection  Oceanic Controlled Airspace (0, m) <-> Air Traffic Manager (0, m)

    instance connection  Oceanic Controlled Airspace (0, m) <-> Traffic Manager (0, m)

message connection  Intrinsic only

object states  None

attributes

    attribute  Capacity  (see Airspace/Ground Resource)

    attribute  Configuration  (see Airspace/Ground Resource)

    attribute  Demand  (see Airspace/Ground Resource)

    attribute  Load  (see Airspace/Ground Resource)

    attribute  Location  (see Airspace/Ground Resource)

    attribute  Name  (see Airspace/Ground Resource)

    attribute  Saturation Threshold  (see Airspace/Ground Resource)

    attribute  Separation Minima  (see Airspace/Ground Resource)

    attribute  Usage Restrictions  (see Airspace/Ground Resource)
services The Oceanic Controlled Airspace class performs the implicit services create, connect, and release. An Oceanic Controlled Airspace object performs the implicit service access.

service Update Demand

• Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service Update Load

• Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input None

external output None

notes None
specification  Oceanic Route - a low-altitude one-way oceanic airway defined by NAVAIDs.

structure

specialization of  Oceanic Airway

contains parts  Fix/Waypoint; Route Segment

instance connections

instance connection  Oceanic Route (0, m) <-> Flight (0, m)
instance connection  Oceanic Route (0, m) <-> Aircraft (0, m)
instance connection  Oceanic Route (0, m) <-> Flight Manager (0, m)
instance connection  Oceanic Route (0, m) <-> Air Traffic Manager (0, m)
instance connection  Oceanic Route (0, m) <-> Traffic Manager (0, m)

message connection  Intrinsic only

object states  None

attributes

attribute  Capacity (see Airspace/Ground Resource)
attribute  Configuration (see Airspace/Ground Resource)
attribute  Demand (see Airspace/Ground Resource)
attribute  Load (see Airspace/Ground Resource)
attribute  Location (see Airspace/Ground Resource)
attribute  Name (see Airspace/Ground Resource)
attribute  Primary Direction (Inherited)
attribute  Saturation Threshold (see Airspace/Ground Resource)
attribute  Separation Minima (see Airspace/Ground Resource)
attribute  Time Period (Inherited)
attribute Usage Restrictions (see Airspace/Ground Resource)

services The Oceanic Route class performs the implicit services create, connect, and release. An Oceanic Route object performs the implicit service access.

service Update Demand

• Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service Update Load

• Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input None

external output None

notes None
**specification**  PAR - Preferential Arrival Route; a specific arrival route from an appropriate en route point to an airport or arrival area. It may be included in a Standard Terminal Arrival Route (STAR) or a Preferred IFR Route. (AIM)

**structure**

specialization of  Preferential Route

part of  Preferred IFR Route; STAR

**instance connections**

instance connection  PAR (0, m) <-> Flight (0, m)

instance connection  PAR (0, m) <-> Aircraft (0, m)

instance connection  PAR (0, m) <-> Flight Manager (0, m)

instance connection  PAR (0, m) <-> Air Traffic Manager (0, m)

instance connection  PAR (0, m) <-> Traffic Manager (0, m)

**message connection**  Intrinsic only

**object states**  None

**attributes**

attribute  Capacity (see Airspace/Ground Resource)

attribute  Configuration (see Airspace/Ground Resource)

attribute  Demand (see Airspace/Ground Resource)

attribute  Load (see Airspace/Ground Resource)

attribute  Location (see Airspace/Ground Resource)

attribute  Name (see Airspace/Ground Resource)

attribute  Saturation Threshold (see Airspace/Ground Resource)

attribute  Separation Minima (see Airspace/Ground Resource)

attribute  Usage Restrictions (see Airspace/Ground Resource)

© 2000 The MITRE Corporation. ALL RIGHTS RESERVED.
services  The PAR class performs the implicit services create, connect, and release. A PAR object performs the implicit service access.

service  Update Demand

• Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service  Update Load

• Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input  None

external output  None

notes  None
**specification** PDAR - Preferential Departure and Arrival Route; a route between two terminals which are within or immediately adjacent to one ARTCC's area. (AIM)

**structure**

specialization of  Preferential Route

**instance connections**

instance connection  PDAR (0, m) <-> Flight (0, m)

instance connection  PDAR (0, m) <-> Aircraft (0, m)

instance connection  PDAR (0, m) <-> Flight Manager (0, m)

instance connection  PDAR (0, m) <-> Air Traffic Manager (0, m)

instance connection  PDAR (0, m) <-> Traffic Manager (0, m)

**message connection**  Intrinsic only

**object states**  None

**attributes**

attribute  Capacity (see Airspace/Ground Resource)

attribute  Configuration (see Airspace/Ground Resource)

attribute  Demand (see Airspace/Ground Resource)

attribute  Load (see Airspace/Ground Resource)

attribute  Location (see Airspace/Ground Resource)

attribute  Name (see Airspace/Ground Resource)

attribute  Saturation Threshold (see Airspace/Ground Resource)

attribute  Separation Minima (see Airspace/Ground Resource)

attribute  Usage Restrictions (see Airspace/Ground Resource)

**services**  The PDAR class performs the implicit services create, connect, and release. A PDAR object performs the implicit service access.

© 2000 The MITRE Corporation. ALL RIGHTS RESERVED.
service Update Demand

• Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service Update Load

• Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input None

external output None

notes None
**specification**  PDR - a specific departure route from an airport or terminal area to an en route point where there is no further need for flow control. It may be included in a SID or Preferred IFR Route. (AIM)

**structure**

**specialization of**  Preferential Route

**part of**  Preferred IFR Route; SID

**instance connections**

**instance connection**  PDR (0, m) <=-=> Flight (0, m)

**instance connection**  PDR (0, m) <=-=> Aircraft (0, m)

**instance connection**  PDR (0, m) <=-=> Flight Manager (0, m)

**instance connection**  PDR (0, m) <=-=> Air Traffic Manager (0, m)

**instance connection**  PDR (0, m) <=-=> Traffic Manager (0, m)

**message connection**  Intrinsic only

**object states**  None

**attributes**

**attribute**  Capacity  (see Airspace/Ground Resource)

**attribute**  Configuration  (see Airspace/Ground Resource)

**attribute**  Demand  (see Airspace/Ground Resource)

**attribute**  Load  (see Airspace/Ground Resource)

**attribute**  Location  (see Airspace/Ground Resource)

**attribute**  Name  (see Airspace/Ground Resource)

**attribute**  Saturation Threshold  (see Airspace/Ground Resource)

**attribute**  Separation Minima  (see Airspace/Ground Resource)

**attribute**  Usage Restrictions  (see Airspace/Ground Resource)
The PDR class performs the implicit services create, connect, and release. A PDR object performs the implicit service access.

**service** Update Demand

- Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

**service** Update Load

- Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

**external input** None

**external output** None

**notes** None
specification  Pilot - an aircraft operator (not including any other flight crewmember) or an on-board automation system with decision-making authority which interacts with ATC

structure

part of Aircraft

instance connection  None

message connection

message  Message Available, from a Vehicle Communications System object that is part of the same Aircraft object as the Pilot object

message  Perform Navigation, sent to the Vehicle Navigation System object which is part of the same Aircraft object as the Pilot object

message  Request Adherence to Clearance, from a Flight Manager object

message  Request Adherence to Clearance, from a Vehicle Manager object

message  Request Aircraft Position, to the Vehicle Navigation System object which is part of the same Aircraft object as the Pilot object

message  Request Amend Clearance, to the Arrival Clearance object which is part of the Flight Clearance object which is part of the same Aircraft object as the Pilot object

message  Request Amend Clearance, to the Departure Clearance object which is part of the Flight Clearance object which is part of the same Aircraft object as the Pilot object

message  Request Amend Clearance, to the En Route Clearance object which is part of the Flight Clearance object which is part of the same Aircraft object as the Pilot object

message  Request Amend Clearance, to the Ground Clearance object which is part of the same Vehicle object as the Aircraft object of which the Pilot object is part

message  Request Amend Flight Plan, to the Flight Plan object which is part of the same Flight object as the Aircraft object of which the Pilot object is part

message  Request Amend Flight Plan & Clearance, from a Flight Manager object

message  Request Amend Ground Vehicle Route, to the Ground Vehicle Route object which is part of the same Vehicle object as the Aircraft object of which the Pilot object is part
message  Request Amend Ground Vehicle Route & Ground Clearance, from a Vehicle Manager object

message  Request Flight Plan Amendment, to a Flight Manager object

message  Request Ground Vehicle Route Amendment, to a Vehicle Manager object

message  Request Report Aircraft Position, from a Flight Manager object

message  Request Report Aircraft Position, from a Vehicle Manager object

message  Request Transfer of Communications, from a Flight Manager object

message  Request Transfer of Communications, from a Vehicle Manager object

message  Send Aircraft Altitude, to a Vehicle Surveillance System object which is part of the same Aircraft object as the Pilot object

message  Send Aircraft Identification, to a Vehicle Surveillance System object which is part of the same Aircraft object as the Pilot object

message  Send Message, to a Vehicle Communications System object which is part of the same Aircraft object as the Pilot object

message  Set Frequency, to a Vehicle Communications System object which is part of the same Aircraft object as the Pilot object

object states  None

attribute  Position - when human, one of two aircraft operators: the pilot-in-command or the second-in-command

services  The Pilot class performs the implicit services create, connect, and release. A Pilot object performs the implicit service access.

service create

•  Create and initialize, as specified for the implicit service create.

•  When a Pilot object is created and there does not already exist an Aircraft object of which the Pilot object is to be a part, requests the Aircraft class to create an Aircraft object of which the Pilot object is a part.
service  Avoid Other Air Traffic

• Ensure that other air traffic is avoided, whether based on personal observation, guidance from on-board equipment, or the direction of a flight manager.

service  Avoid Other Ground Traffic

• Ensure that other ground traffic is avoided, whether based on personal observation or the direction of a vehicle manager.

service  Execute Flight

• Perform the duties necessary to carry out the flight, including (for example)
  – Ensuring that the aircraft follows the flight plan.
  – Ensuring that clearances and instructions are carried out.

service  Execute Movement

• Perform the duties necessary to carry out the movement on the airport movement area, including (for example)
  – Ensuring that the aircraft or ground vehicle follows the planned ground plan.
  – Ensuring that clearances and instructions are carried out.

service  Manage Equipment

• Perform the duties necessary to ensure the proper performance of the Vehicle Surveillance System object, the Vehicle Navigation System object, and the Vehicle Communications System object.

service  Plan Flight

• Perform the duties necessary to plan the flight, including (for example)
  – Requesting preflight briefings.
  – Filing the flight plan.
  – Coordinating changes to the flight plan.

service  Report Aircraft Altitude
• Upon request by a Flight Manager object, request a Vehicle Surveillance System object which is part of the same Aircraft object to report the altitude of the aircraft to the appropriate Surveillance System object.

**service** Report Aircraft Identification

• Upon request by a Flight Manager object, request a Vehicle Surveillance System object which is part of the same Aircraft object to report the identification of the aircraft to the appropriate Surveillance System object.

**service** Report Aircraft Position

• Upon request by a Flight Manager object, request a Vehicle Navigation System object which is part of the same Aircraft object to determine the location of the aircraft and return position information to the Flight Manager object.

**service** Request Movement

• Coordinate movement on the airport movement area with a vehicle manager.

*external input* None

*external output* None

**notes**

1. The (non-implicit) services of a Pilot object might be performed either by a human or by automation, depending on the particular aircraft.
2. The service Manage Equipment is performed, for the most part, independently of other objects, but also responds to external requests, e.g., a manager-type object requesting a particular beacon code setting for the transponder.
3. The service Report Aircraft Position gets information from the Vehicle Navigation System object, not the Vehicle Surveillance System object, which responds independently to ATC's Surveillance System object.
4. The service Plan Flight is not reserved exclusively to the Pilot object, e.g., the original flight plan might have been filed by an Airline Dispatch Office, but, throughout the flight, the pilot interacts with a manager when flight planning is involved.
specification  Preferential Route - a route, usually confined to one ARTCC’s area, which is adapted to accomplish inter/intrafacility controller coordination and to assure that flight data is posted at the proper control position.  (AIM)

structure

specialization of  Terminal Route

generalization of  PAR;  PDAR;  PDR

instance connections

instance connection  Preferential Route (0, m) <-> Flight (0, m)

instance connection  Preferential Route (0, m) <-> Aircraft (0, m)

instance connection  Preferential Route (0, m) <-> Flight Manager (0, m)

instance connection  Preferential Route (0, m) <-> Air Traffic Manager (0, m)

instance connection  Preferential Route (0, m) <-> Traffic Manager (0, m)

message connection  Intrinsic only

object states  None

attributes

attribute  Capacity  (see Airspace/Ground Resource)

attribute  Configuration  (see Airspace/Ground Resource)

attribute  Demand  (see Airspace/Ground Resource)

attribute  Load  (see Airspace/Ground Resource)

attribute  Location  (see Airspace/Ground Resource)

attribute  Name  (see Airspace/Ground Resource)

attribute  Saturation Threshold  (see Airspace/Ground Resource)

attribute  Separation Minima  (see Airspace/Ground Resource)

attribute  Usage Restrictions  (see Airspace/Ground Resource)
The Preferential Route class performs the implicit services create, connect, and release. A Preferential Route object performs the implicit service access.

**service**  Update Demand

- Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

**service**  Update Load

- Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

**external input**  None

**external output**  None

**notes**  None
Preferred IFR Route - a route established between busier airports to increase system efficiency and capacity. Preferred IFR Routes normally extend through one or more ARTCC areas and are designed to achieve balanced traffic flows among high density terminals. Preferred IFR Routes are correlated with SIDs and STARs and may be defined by airways, jet routes, direct routes between NAVAIDs, Waypoints, NAVAID radials/DME, or any combination thereof. (AIM)

structure
c contains parts Route; PDR; PAR

instance connections

instance connection Preferred IFR Route (0, m) <-> Flight (0, m)
instance connection Preferred IFR Route (0, m) <-> Aircraft (0, m)
instance connection Preferred IFR Route (0, m) <-> Flight Manager (0, m)
instance connection Preferred IFR Route (0, m) <-> Air Traffic Manager (0, m)
instance connection Preferred IFR Route (0, m) <-> Traffic Manager (0, m)

message connection Intrinsic only

object states None

attributes

attribute Capacity (see Airspace/Ground Resource)
attribute Configuration (see Airspace/Ground Resource)
attribute Demand (see Airspace/Ground Resource)
attribute Load (see Airspace/Ground Resource)
attribute Location (see Airspace/Ground Resource)
attribute Name (see Airspace/Ground Resource)
attribute Saturation Threshold (see Airspace/Ground Resource)
attribute Separation Minima (see Airspace/Ground Resource)
attribute Usage Restrictions (see Airspace/Ground Resource)

© 2000 The MITRE Corporation. ALL RIGHTS RESERVED.
services  The Preferred IFR Route class performs the implicit services create, connect, and release. A Preferred IFR Route object performs the implicit service access.

service  Update Demand

• Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service  Update Load

• Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input  None

eexternal output  None

notes  Although the Military Training Route class is a specialization of the Route class, a preferred IFR route would probably not be correlated with a military training route.
specification  Protected Airspace Volume - special use airspace (i.e., alert area, controlled firing area, military operations area, prohibited area, restricted area, or warning area) or other airspace (e.g., heavy weather area) in or near which a manager might want to restrict aircraft activity.

structure

specialization of  Airspace Volume

instance connections

instance connection  Protected Airspace Volume (0, m) <-> Flight (0, m)
instance connection  Protected Airspace Volume (0, m) <-> Aircraft (0, m)
instance connection  Protected Airspace Volume (0, m) <-> Flight Manager (0, m)
instance connection  Protected Airspace Volume (0, m) <-> Air Traffic Manager (0, m)
instance connection  Protected Airspace Volume (0, m) <-> Traffic Manager (0, m)

message connection  Intrinsic only

object states  None

attributes

attribute  Capacity  (see Airspace/Ground Resource)
attribute  Configuration  (see Airspace/Ground Resource)
attribute  Demand  (see Airspace/Ground Resource)
attribute  Load  (see Airspace/Ground Resource)
attribute  Location  (see Airspace/Ground Resource)
attribute  Name  (see Airspace/Ground Resource)
attribute  Saturation Threshold  (see Airspace/Ground Resource)
attribute  Separation Minima  (see Airspace/Ground Resource)
attribute  Usage Restrictions  (see Airspace/Ground Resource)

services  The Protected Airspace Volume class performs the implicit services create, connect, and release. A Protected Airspace Volume object performs the implicit service access.
service  Update Demand

- Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service  Update Load

- Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input  None

external output  None

notes  None
**specification**  Resource - a source of supply or support; a source of information or expertise

**structure**

*generalization of*  Airspace/Ground Resource; Surveillance Resource; Navigation Resource; Communications Resource; Aviation Weather Resource

**instance connection**  None

**message connection**  None

**object states**  None

**attribute**  None

**services**  None

**external input**  None

**external output**  None

**notes**  The Resource class is included as a convenient highest class in the resource hierarchy; objects typically occur at a lower level within the specialization classes.
**specification**  Route - a defined path, consisting of one or more courses in a horizontal plane, which aircraft traverse over the surface of the earth. (AIM) In this case, the route is contained within en route airspace.

**structure**

*part of*  En Route Controlled Airspace;  Preferred IFR Route

*contains parts*  Fix/Waypoint;  Route Segment

*generalization of*  Airway;  Area Route;  Jet Route;  Military Training Route

**instance connections**

*instance connection*  Route (0, m) <-> Flight (0, m)

*instance connection*  Route (0, m) <-> Aircraft (0, m)

*instance connection*  Route (0, m) <-> Flight Manager (0, m)

*instance connection*  Route (0, m) <-> Air Traffic Manager (0, m)

*instance connection*  Route (0, m) <-> Traffic Manager (0, m)

**message connection**  Intrinsic only

**object states**  None

**attributes**

*attribute*  Capacity (see Airspace/Ground Resource)

*attribute*  Configuration (see Airspace/Ground Resource)

*attribute*  Demand (see Airspace/Ground Resource)

*attribute*  Load (see Airspace/Ground Resource)

*attribute*  Location (see Airspace/Ground Resource)

*attribute*  Name (see Airspace/Ground Resource)

*attribute*  Saturation Threshold (see Airspace/Ground Resource)

*attribute*  Separation Minima (see Airspace/Ground Resource)
**attribute** Usage Restrictions (see Airspace/Ground Resource)

**services** The Route class performs the implicit services create, connect, and release. A Route object performs the implicit service access.

**service** Update Demand

- Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

**service** Update Load

- Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

**external input** None

**external output** None

**notes** The specializations of the Route class do not add attributes or services to distinguish them from the generalization class, but are included because they are commonly differentiated from each other in the ATC world on the basis of aircraft equipage and performance.
Specification

Route Segment - as used in ATC, a part of a route that can be defined by two navigational fixes, two NAVAIDs, or a fix and a NAVAID. (AIM) For the ATC model, it is assumed that a route segment can also be defined with a waypoint as an endpoint.

Structure

Part of Track; Oceanic Route; Route; Terminal Route

Instance connections

Instance connection Route Segment (0, m) <-> Flight (0, m)
Instance connection Route Segment (0, m) <-> Aircraft (0, m)
Instance connection Route Segment (0, m) <-> Flight Manager (0, m)
Instance connection Route Segment (0, m) <-> Air Traffic Manager (0, m)
Instance connection Route Segment (0, m) <-> Traffic Manager (0, m)

Message connection Intrinsic only

Object states None

Attributes

Attribute Capacity (see Airspace/Ground Resource)
Attribute Configuration (see Airspace/Ground Resource)
Attribute Demand (see Airspace/Ground Resource)
Attribute Endpoints - the fix/waypoint at each end of the route segment
Attribute Lateral & Altitude Limits - the boundaries of the route segment; an aircraft within the boundaries of the route segment is assumed to be on the route segment
Attribute Load (see Airspace/Ground Resource)
Attribute Location (see Airspace/Ground Resource)
Attribute Name (see Airspace/Ground Resource)
Attribute Saturation Threshold (see Airspace/Ground Resource)
Attribute Separation Minima (see Airspace/Ground Resource)
attribute  Usage Restrictions  (see Airspace/Ground Resource)

services  The Route Segment class performs the implicit services create, connect, and release. A Route Segment object performs the implicit service access.

service  Update Demand

•  Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service  Update Load

•  Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input  None

external output  None

notes  None
Runway - a defined rectangular area on a land airport prepared for the landing and takeoff run of aircraft along its length. Runways are normally numbered in relation to their magnetic direction rounded off to the nearest 10 degrees; e.g., Runway 01, Runway 25. (AIM)

structure

part of  Airport Movement Area

instance connections

instance connection  Runway (0, m) <-> Vehicle (0, m)
instance connection  Runway (0, m) <-> Aircraft (0, m)
instance connection  Runway (0, m) <-> Vehicle Manager (0, m)
instance connection  Runway (0, m) <-> Ground Traffic Manager (0, m)
instance connection  Runway (0, m) <-> Traffic Manager (0, m)

object states  None

attributes

attribute  Capacity (see Airspace/Ground Resource)
attribute  Configuration (see Airspace/Ground Resource)
attribute  Demand (see Airspace/Ground Resource)
attribute  Load (see Airspace/Ground Resource)
attribute  Location (see Airspace/Ground Resource)
attribute  Name (see Airspace/Ground Resource)
attribute  Saturation Threshold (see Airspace/Ground Resource)
attribute  Separation Minima (see Airspace/Ground Resource)
attribute  Status - active or inactive.

attribute  Surface Condition - the condition of the runway surface (e.g., covered with ice, snow, or long grass).

attribute  Usage Restrictions (see Airspace/Ground Resource)
services  The Runway class performs the implicit services create, connect, and release. A Runway object performs the implicit service access.

service  Update Demand

• Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service  Update Load

• Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input  None

external output  None

notes  None
specification  Sector - the smallest unit of airspace assigned to a Flight Manager.

structure

specialization of  Airspace Volume

part of  ACF Airspace

instance connections

instance connection  Sector (0, m) <-> Flight (0, m)
instance connection  Sector (0, m) <-> Aircraft (0, m)
instance connection  Sector (0, m) <-> Flight Manager (0, m)
instance connection  Sector (0, m) <-> Air Traffic Manager (0, m)
instance connection  Sector (0, m) <-> Traffic Manager (0, m)

message connection  Intrinsic only

object states  None

attributes

attribute  Capacity  (see Airspace/Ground Resource)
attribute  Configuration  (see Airspace/Ground Resource)
attribute  Demand  (see Airspace/Ground Resource)
attribute  Load  (see Airspace/Ground Resource)
attribute  Location  (see Airspace/Ground Resource)
attribute  Name  (see Airspace/Ground Resource)
attribute  Saturation Threshold  (see Airspace/Ground Resource)
attribute  Separation Minima  (see Airspace/Ground Resource)
attribute  Usage Restrictions  (see Airspace/Ground Resource)

services  The Sector class performs the implicit services create, connect, and release. A Sector object performs the implicit service access.
service  Update Demand

• Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service  Update Load

• Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input  None

external output  None

notes  None
specification  SID - Standard Instrument Departure; a preplanned IFR ATC departure procedure printed for pilot use in graphic and/or textual form. SIDs provide transition from the terminal to the appropriate en route structure. (AIM)

structure

specialization of  Terminal Route

contains parts  PDR

instance connections

instance connection  SID (0, m) <-> Flight (0, m)

instance connection  SID (0, m) <-> Aircraft (0, m)

instance connection  SID (0, m) <-> Flight Manager (0, m)

instance connection  SID (0, m) <-> Air Traffic Manager (0, m)

instance connection  SID (0, m) <-> Traffic Manager (0, m)

message connection  Intrinsic only

object states  None

attributes

attribute  Capacity (see Airspace/Ground Resource)

attribute  Configuration (see Airspace/Ground Resource)

attribute  Demand (see Airspace/Ground Resource)

attribute  Load (see Airspace/Ground Resource)

attribute  Location (see Airspace/Ground Resource)

attribute  Name (see Airspace/Ground Resource)

attribute  Saturation Threshold (see Airspace/Ground Resource)

attribute  Separation Minima (see Airspace/Ground Resource)

attribute  Usage Restrictions (see Airspace/Ground Resource)
services  The SID class performs the implicit services create, connect, and release. A SID object performs the implicit service access.

service  Update Demand

• Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service  Update Load

• Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input  None

external output  None

notes  None
**specification**  STAR - Standard Terminal Arrival; a preplanned IFR ATC arrival procedure published for pilot use in graphic and/or textual form. STARs provide transition from the en route structure to an outer fix or an instrument approach fix/arrival waypoint in the terminal area.

**structure**

*specialization of*  Terminal Route

*contains parts*  PAR

**instance connections**

*instance connection*  STAR (0, m) <-> Flight (0, m)

*instance connection*  STAR (0, m) <-> Aircraft (0, m)

*instance connection*  STAR (0, m) <-> Flight Manager (0, m)

*instance connection*  STAR (0, m) <-> Air Traffic Manager (0, m)

*instance connection*  STAR (0, m) <-> Traffic Manager (0, m)

**message connection**  Intrinsic only

**object states**  None

**attributes**

*attribute*  Capacity  (see Airspace/Ground Resource)

*attribute*  Configuration  (see Airspace/Ground Resource)

*attribute*  Demand  (see Airspace/Ground Resource)

*attribute*  Load  (see Airspace/Ground Resource)

*attribute*  Location  (see Airspace/Ground Resource)

*attribute*  Name  (see Airspace/Ground Resource)

*attribute*  Saturation Threshold  (see Airspace/Ground Resource)

*attribute*  Separation Minima  (see Airspace/Ground Resource)

*attribute*  Usage Restrictions  (see Airspace/Ground Resource)
services  The STAR class performs the implicit services create, connect, and release. A STAR object performs the implicit service access.

service  Update Demand

• Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service  Update Load

• Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input  None

external output  None

notes  None
**specification**  Surveillance Resource - equipment for the tracking of aircraft positions and the management of the information provided.

**structure**

  **specialization of** Resource

  contains parts  Surveillance System

  instance connection  None

  message connection  None

  object states  None

  attribute  None

**services**  The Surveillance Resource class performs the implicit services create, connect, and release. A Surveillance Resource object performs the implicit service access.

  external input  None

  external output  None

**notes**  (1) The Surveillance Resource class exists only for convenience as the highest-level surveillance resource. Surveillance resource-type objects typically exist at a lower level in the surveillance hierarchy. (2) The Surveillance Resource class refers only to the surveillance functionality of the ATC system.
specification  Surveillance System - a system for the tracking of vehicle (i.e., aircraft and/or ground vehicle) positions and the management of the information provided.

structure

   part of  Surveillance Resource

   generalization of  Airspace Surveillance System;  Ground Surveillance System

instance connection  Surveillance System (0, m) <-> Vehicle Surveillance System (0, m)

message connections

   message  Request Altitude Report, sent from Surveillance System

   message  Transmit & Receive Radar Pulse, sent from Surveillance System

   message  Transmit & Receive Radar Pulse, sent to Surveillance System

object states  None

attributes

   attribute  Area of Coverage - the range of the surveillance equipment.

   attribute  Equipment Type - the type of equipment used by the surveillance system, e.g., a particular air route surveillance radar (ARSR) series or airport surveillance radar (ASR) series.

   attribute  Status - operational or not operational.

services  The Surveillance System class performs the implicit services create, connect, and release.  A Surveillance System object performs the implicit service access.

service  connect

   •  Upon determination that a vehicle is within the Area of Coverage of the Surveillance System object (i.e., upon receiving responses to interrogations), create an instance connection between the Surveillance System object and the Vehicle Surveillance System object which is part of the Aircraft or Ground Vehicle object.

   •  Upon determination that a vehicle is no longer within the Area of Coverage of the Surveillance System object (i.e., responses to interrogations are no longer received), remove the instance connection between the Surveillance System object and the Vehicle Surveillance System object which is part of the Aircraft or Ground Vehicle object.
service  Locate, Identify & Report Flight

- Locate each aircraft in the area of coverage, determine the identification of the flight of which the aircraft is a part, and report its identification and location.

external input  None

external output  None

notes  (1) Instance connections and message connections exist almost exclusively for objects of specializations of surveillance system classes (not for objects of the Surveillance System class itself). (2) In reality, a surveillance system might not be able to locate and identify all vehicles in its area of coverage, due in part to a lack of vehicle equipage and in part to capacity limitations of the surveillance system itself.
specification  Taxiway - paved areas of the airport used by aircraft to proceed to or from the runways. (Nolan, 1990)

structure

part of  Airport Movement Area

instance connections

instance connection  Taxiway (0, m) <-> Vehicle (0, m)
instance connection  Taxiway (0, m) <-> Aircraft (0, m)
instance connection  Taxiway (0, m) <-> Vehicle Manager (0, m)
instance connection  Taxiway (0, m) <-> Ground Traffic Manager (0, m)
instance connection  Taxiway (0, m) <-> Traffic Manager (0, m)

object states  None

attributes

attribute  Capacity  (see Airspace/Ground Resource)
attribute  Configuration  (see Airspace/Ground Resource)
attribute  Demand  (see Airspace/Ground Resource)
attribute  Load  (see Airspace/Ground Resource)
attribute  Location  (see Airspace/Ground Resource)
attribute  Name  (see Airspace/Ground Resource)
attribute  Saturation Threshold  (see Airspace/Ground Resource)
attribute  Separation Minima  (see Airspace/Ground Resource)
attribute  Status - active or inactive.
attribute  Surface Condition - the condition of the taxiway surface (e.g., covered with ice, snow, or long grass).
attribute  Usage Restrictions  (see Airspace/Ground Resource)
services  The Taxiway class performs the implicit services create, connect, and release. A Taxiway object performs the implicit service access.

service  Update Demand

- Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service  Update Load

- Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input  None

external output  None

notes  None
**specification**  Terminal Airspace Volume - any volume of terminal airspace.

**structure**

*part of*  Terminal Controlled Airspace

*generalization of*  Terminal Control Area;  Airport Radar Services Area;  Airport Traffic Area

**instance connections**

*instance connection*  Terminal Airspace Volume (0, m) <-> Flight (0, m)

*instance connection*  Terminal Airspace Volume (0, m) <-> Aircraft (0, m)

*instance connection*  Terminal Airspace Volume (0, 1) <-> Flight Manager (0, m)

*instance connection*  Terminal Airspace Volume (0, m) <-> Air Traffic Manager (0, m)

*instance connection*  Terminal Airspace Volume (0, m) <-> Traffic Manager (0, m)

**message connection**  Intrinsic only

**object states**  None

**attributes**

*attribute*  Capacity  (see Airspace/Ground Resource)

*attribute*  Configuration  (see Airspace/Ground Resource)

*attribute*  Demand  (see Airspace/Ground Resource)

*attribute*  Load  (see Airspace/Ground Resource)

*attribute*  Location  (see Airspace/Ground Resource)

*attribute*  Name  (see Airspace/Ground Resource)

*attribute*  Saturation Threshold  (see Airspace/Ground Resource)

*attribute*  Separation Minima  (see Airspace/Ground Resource)

*attribute*  Usage Restrictions  (see Airspace/Ground Resource)
services  The Terminal Airspace Volume class performs the implicit services create, connect, and release. A Terminal Airspace Volume object performs the implicit service access.

service  Update Demand

• Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service  Update Load

• Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input  None

external output  None

notes  (1) The Terminal Airspace Volume class was introduced to permit the inclusion of any arbitrarily-defined volume of airspace. (2) The list of specialization class is not exhaustive, but includes commonly-defined types of airspace. Other specializations are possible.
specification  Terminal Control Area - TCA; controlled airspace extending upward from the surface or higher to specified altitudes, within which all aircraft are subject to operating rules and pilot and equipment requirements specified in FAR Part 91. (AIM)

structure

specialization of  Terminal Airspace Volume

instance connections

instance connection  Terminal Control Area (0, m) <-> Flight (0, m)
instance connection  Terminal Control Area (0, m) <-> Aircraft (0, m)
instance connection  Terminal Control Area (0, 1) <-> Flight Manager (0, m)
instance connection  Terminal Control Area (0, m) <-> Air Traffic Manager (0, m)
instance connection  Terminal Control Area (0, m) <-> Traffic Manager (0, m)

message connection  Intrinsic only

object states  None

attributes

attribute  Capacity (see Airspace/Ground Resource)
attribute  Configuration (see Airspace/Ground Resource)
attribute  Demand (see Airspace/Ground Resource)
attribute  Load (see Airspace/Ground Resource)
attribute  Location (see Airspace/Ground Resource)
attribute  Name (see Airspace/Ground Resource)
attribute  Saturation Threshold (see Airspace/Ground Resource)
attribute  Separation Minima (see Airspace/Ground Resource)
attribute  Usage Restrictions (see Airspace/Ground Resource)

services  The Terminal Control Area class performs the implicit services create, connect, and release. A Terminal Control Area object performs the implicit service access.
service  Update Demand

- Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service  Update Load

- Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input  None

external output  None

notes  None
specification  Terminal Controlled Airspace - that part of controlled airspace used primarily for the arrival and departure phases of flights.

structure

part of  Controlled Airspace

contains parts  Terminal Route;  Terminal Airspace Volume

instance connections

instance connection  Terminal Controlled Airspace (0, m) <-> Flight (0, m)
instance connection  Terminal Controlled Airspace (0, m) <-> Aircraft (0, m)
instance connection  Terminal Controlled Airspace (0, 1) <-> Flight Manager (0, m)
instance connection  Terminal Controlled Airspace (0, m) <-> Air Traffic Manager (0, m)
instance connection  Terminal Controlled Airspace (0, m) <-> Traffic Manager (0, m)

message connection  Intrinsic only

object states  None

attributes

attribute  Capacity  (see Airspace/Ground Resource)
attribute  Configuration  (see Airspace/Ground Resource)
attribute  Demand  (see Airspace/Ground Resource)
attribute  Load  (see Airspace/Ground Resource)
attribute  Location  (see Airspace/Ground Resource)
attribute  Name  (see Airspace/Ground Resource)
attribute  Saturation Threshold  (see Airspace/Ground Resource)
attribute  Separation Minima  (see Airspace/Ground Resource)
attribute  Usage Restrictions  (see Airspace/Ground Resource)
services  The Terminal Controlled Airspace class performs the implicit services create, connect, and release. A Terminal Controlled Airspace object performs the implicit service access.

service  Update Demand

• Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service  Update Load

• Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input  None

external output  None

notes  A Terminal Route object and a Terminal Control Volume object may define parts of the same airspace, but both objects reflect common usage.
specification  Terminal Route - a defined path, within terminal airspace, consisting of one or more courses in a horizontal plane, which aircraft traverse over the surface of the earth. Some terminal routes may also specify the altitude transitions.

structure

part of  Terminal Controlled Airspace
contains parts  Fix/Waypoint; Route Segment
generalization of  Preferential Route; SID; STAR

instance connections

instance connection  Terminal Route (0, m) <-> Flight (0, m)
instance connection  Terminal Route (0, m) <-> Aircraft (0, m)
instance connection  Terminal Route (0, m) <-> Flight Manager (0, m)
instance connection  Terminal Route (0, m) <-> Air Traffic Manager (0, m)
instance connection  Terminal Route (0, m) <-> Traffic Manager (0, m)

message connection  Intrinsic only

object states  None

attributes

attribute  Capacity  (see Airspace/Ground Resource)
attribute  Configuration  (see Airspace/Ground Resource)
attribute  Demand  (see Airspace/Ground Resource)
attribute  Load  (see Airspace/Ground Resource)
attribute  Location  (see Airspace/Ground Resource)
attribute  Name  (see Airspace/Ground Resource)
attribute  Saturation Threshold  (see Airspace/Ground Resource)
attribute  Separation Minima  (see Airspace/Ground Resource)
attribute  Usage Restrictions  (see Airspace/Ground Resource)

services  The Terminal Route class performs the implicit services create, connect, and release. A Terminal Route object performs the implicit service access.

service  Update Demand
- Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

service  Update Load
- Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

external input  None
external output  None

notes  A Terminal Route might be the route followed by an instrument approach procedure, which is a series of predetermined maneuvers for the orderly transfer of an aircraft under instrument flight conditions from the beginning of the initial approach to a landing or to a point from which a landing may be made visually (AIM). In this case, the route segments are the Initial Approach Segment, the Intermediate Approach Segment, the Final Approach Segment, and the Missed Approach Segment.
specification  Track - a high-altitude one-way oceanic airway defined by whole degrees of latitude and longitude.

structure

specialization of  Oceanic Airway

contains parts  Fix/Waypoint; Route Segment

instance connections

instance connection  Track (0, m) <-> Flight (0, m)
instance connection  Track (0, m) <-> Aircraft (0, m)
instance connection  Track (0, m) <-> Flight Manager (0, m)
instance connection  Track (0, m) <-> Air Traffic Manager (0, m)
instance connection  Track (0, m) <-> Traffic Manager (0, m)

message connection  Intrinsic only

object states  None

attributes

attribute  Capacity (see Airspace/Ground Resource)
attribute  Configuration (see Airspace/Ground Resource)
attribute  Demand (see Airspace/Ground Resource)
attribute  Load (see Airspace/Ground Resource)
attribute  Location (see Airspace/Ground Resource)
attribute  Name (see Airspace/Ground Resource)
attribute  Primary Direction (Inherited)
attribute  Saturation Threshold (see Airspace/Ground Resource)
attribute  Separation Minima (see Airspace/Ground Resource)
attribute  Time Period (Inherited)
**attribute** Usage Restrictions (see Airspace/Ground Resource)

**services** The Track class performs the implicit services create, connect, and release. A Track object performs the implicit service access.

**service** Update Demand

- Update the value of the Demand attribute to reflect the addition, change, or deletion of an instance connection.

**service** Update Load

- Update the value of the Load attribute to reflect the addition, change, or deletion of an instance connection.

**external input** None

**external output** None

**notes** None
**Specification**  Traffic - a grouping of flights or vehicles determined by a defined criterion or set of criteria.

**Structure**

*generalization of*  Air Traffic; Ground Traffic

**Instance connection**  Traffic (0, m) <-> Traffic Manager (0, m)

**Message connection**  None

**Object states**  None

**Attribute**  Selection Criteria - a criterion or set of criteria for selection of part objects

**Services**  The Traffic class performs the implicit services create, connect, and release. A Traffic object performs the implicit service access.

**External input**  None

**External output**  None

**Notes**  The definition of traffic is unique to this model, and was created for convenience.
**specification**  Traffic Manager - one who plans or supervises traffic.

**structure**

*specialization of* Manager

*generalization of* Air Traffic Manager;  Ground Traffic Manager

**instance connection**

<table>
<thead>
<tr>
<th>instance connection</th>
<th>Traffic Manager (0, m) &lt;-&gt; Traffic (0, m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>instance connection</td>
<td>Traffic Manager (0, m) &lt;-&gt; Interfacility Communications System (0, m)</td>
</tr>
<tr>
<td>instance connection</td>
<td>Traffic Manager (0, m) &lt;-&gt; Intrafacility Communications System (0, m)</td>
</tr>
<tr>
<td>instance connection</td>
<td>Traffic Manager (0, m) &lt;-&gt; airspace/ground-type resource (0, m)</td>
</tr>
</tbody>
</table>

**message connection**  None

**object states**  None

**attribute**  Area of Responsibility (inherited)

**services**  The Traffic Manager class performs the implicit services create, connect, and release.

A Traffic Manager object performs the implicit service access.

**external input**  None

**external output**  None

**notes**  The Traffic Manager class does not represent a currently-staffed position; instead, it exists for convenience as a specialization of the Air Traffic Manager class and the Ground Traffic Manager class.
specification  Transponder - the airborne radar beacon receiver/transmitter portion of the Air Traffic
Control Radar Beacon System (ATCRBS) which automatically receives radio signals from
interrogators on the ground, and selectively replies with a specific reply pulse or pulse group only to
those interrogations being received on the mode to which it is set to respond. (AIM)

structure

specialization of  Vehicle Surveillance System

instance connections  (inherited)

message connections  (inherited)

object states  (inherited)

attributes  (inherited)

services  (inherited)

external input  (inherited)

external output  (inherited)

notes  The Transponder class is included in the model because the term "transponder" is the
common term for vehicle surveillance system equipment.
specification  User - a person or inanimate object which avails itself of a resource, or carries out a purpose or action by means of a resource.

structure

generalization of  Traffic; Manager

instance connection  None

message connection  None

object states  None

attribute  None

services  None

external input  None

external output  None

notes  The User class was created as a convenient way to show that managers in the ATC system are also users of the ATC system. Any object in the user hierarchy is typically an object of a class lower in the user hierarchy than the User class.
**specification**  Vehicle - the operation of an aircraft or ground vehicle (e.g., baggage cart, fuel truck) on the surface of an airport; a vehicle can have three logical components, which are either an aircraft or a ground vehicle, a ground vehicle route, and a ground clearance.

**structure**

*part of* Ground Traffic

*contains parts* Ground Clearance; Ground Vehicle Route; Aircraft or Ground Vehicle

**instance connections**

*instance connection*  Vehicle object (0, m) <-> Vehicle Manager object (1)

*instance connection*  Vehicle object (0, m) <-> ground resource-type object (0, m)

**message connections**  None

**object states**  None

**attribute**  Type - the type of vehicle, e.g., aircraft, gasoline truck, baggage cart.

**services**  The Vehicle class performs the implicit services create, connect, and release. A Vehicle object performs the implicit service access.

**external input**  None

**external output**  None

**notes**  (1) An instance connection exists between the Vehicle object and the first Vehicle Manager object that can deliver any part of the ground clearance (e.g., the ground controller). After this point, the instance connection can be established and broken if the vehicle is “handed off” from one vehicle manager (e.g., controller) to another. (2) Instance connections between a Vehicle object and a ground resource-type object exist based upon the planned use of the resource by the vehicle as reflected in the planned ground vehicle route. Each instance connection is created when the ground vehicle route is created, changed as the ground vehicle route is changed, and deleted once the resource has been used by the vehicle. Each Vehicle object having an instance connection with a resource-type object contributes to the value of the Demand attribute of that resource.
specification  Vehicle Communications System - the communications capabilities of an individual vehicle (aircraft or ground vehicle) used in interacting with ATC's air/ground communications systems or ground/ground communications systems, e.g., the ability of a vehicle to receive the radio transmission of clearance information

structure

part of  Aircraft;  Ground Vehicle

instance connection  Vehicle Communications System object (0, m) <-> Communications System object (0, m)

message connections

message  Request Accept Message, to the Ground Vehicle Operator object which is part of the same Ground Vehicle object as the Vehicle Communications System object

message  Request Accept Message, to the Pilot object which is part of the same Aircraft object as the Vehicle Communications System object

message  Request Transmit Message, from the Ground Vehicle Operator object which is part of the same Ground Vehicle object as the Vehicle Communications System object

message  Request Transmit Message, from the Pilot object which is part of the same Aircraft object as the Vehicle Communications System object

message  Request Transmit Message, from a Communications System object with which an instance connection exists

message  Request Transmit Message, to a Communications System object with which an instance connection exists

object states  The states of a Vehicle Communications System object (from an ATC point of view) are "active" and "inactive."

attributes

attribute  Active Frequency - the ATC frequency;  the frequency that the vehicle has been instructed to use in communicating with ATC;  the frequency that the manager-type object expects to use in communicating with the vehicle.

attribute  Equipment Type - the type of communications system, i.e., radio or data link.
services The Vehicle Communications System class performs the implicit services create, connect, and release. A Vehicle Communications System object performs the implicit service access.

service Accept Message

• Upon request by a Communications System object, a Pilot object, or a Ground Vehicle Operator object, accept the transmitted message.

service Send Message

• Upon request by a Communications System object, a Pilot object, or a Ground Vehicle Operator object, transmit the message.

external input Communications with other aircraft or with company radios for the purpose of relaying a clearance from ATC.

external output Communications with other aircraft or with company radios for the purpose of relaying a clearance from ATC.

notes Communications capabilities of a vehicle for other than ATC purposes are not included in this model. However, communications capabilities which can be used for ATC (e.g., transmission of a clearance via another aircraft or via the Aeronautical Radio., Inc. (ARINC) Communications Addressing and Reporting System (ACARS) terrestrial digital data link) should be considered part of the model as external input or output.
specification  Vehicle Manager - one who plans or supervises vehicle movement.

structure

specialization of  Manager

instance connections

instance connection  Vehicle Manager (1) <-> Vehicle (0, m)

instance connection  Vehicle Manager (0, m) <-> ground-type resource (0, m)

instance connection  Vehicle Manager (0, m) <-> Air/Ground Communications System (0, m)

instance connection  Vehicle Manager (0, m) <-> Interfacility Communications System (0, m)

instance connection  Vehicle Manager (0, m) <-> Intrafacility Communications System (0, m)

message connections

message  Request Accept Control Responsibility, sent from Vehicle Manager object

message  Request Accept Control Responsibility, sent to Vehicle Manager object

message  Request Adherence to Clearance, sent to Ground Vehicle Operator object

message  Request Adherence to Clearance, sent to Pilot object

message  Request Amend Ground Vehicle Route & Clearance, sent to Ground Vehicle Operator object

message  Request Amend Ground Vehicle Route & Clearance, sent to Pilot object

message  Request Resource, sent from Vehicle Manager object

message  Request Resource, sent to Vehicle Manager object

message  Request Transfer of Communications, sent to Pilot object

message  Request Transfer of Communications, sent to Ground Vehicle Operator object

message  Request Vehicle Separation, sent from Vehicle Manager object
message Request Vehicle Separation, sent to Vehicle Manager object

object states None

attribute Area of Responsibility (Inherited)

services The Vehicle Manager class performs the implicit services create, connect, and release. A Vehicle Manager object performs the implicit service access.

service Assist in Route Planning

• Upon request, evaluate and accept, disapprove, or replan the route, as appropriate.

service Detect, Predict & Report Restriction Violation

• Determine whether a violation of a usage restriction for a ground-type resource is happening or is planned to happen and notify the appropriate manager-type object.

service Detect, Predict & Report Separation Violation

• Determine whether a violation of separation minima is happening or is planned to happen and notify the appropriate manager-type object.

service Direct Transfer of Communications

• Inform a vehicle of the frequency that should be used in future communications.

service Ensure Restriction Compliance

• When requested by a vehicle to use a resource, determine whether a restriction exists for the resource and approve or disapprove the use of the resource, as appropriate.

• When it is discovered that a vehicle has planned to use a resource contrary to usage restrictions, determine whether and how to change the resource usage of that vehicle.

service Generate & Deliver Clearance

• Translate instructions to a vehicle into clearance language and deliver via a communications medium.

service Separate Vehicles

• When requested by a vehicle to change its route, determine whether a separation violation would exist based upon the revised route and approve or disapprove the route, as appropriate.
• When it is discovered that following an existing route would cause a separation violation, determine whether and how to change the route.

service Transfer & Accept Control Responsibility

• When controlling a vehicle: determine the next appropriate vehicle manager and time of control transfer and offer control to that vehicle manager.

• When offered control of a vehicle: evaluate the appropriateness of receiving control of the vehicle and either accept or deny control transfer.

external input None

external output None

notes (1) The Vehicle Manager class does not represent a currently-staffed position; instead, it includes parts of the functionality of several ATC positions. (2) The services of a Vehicle Manager object might be performed either by humans or by automation, or shared.
**specification**  Vehicle Navigation System - the navigational functionality of an individual vehicle used in interacting with a Navigation System object (e.g., the ability of an aircraft VOR/DME-based RNAV unit to receive, interpret, and use radial and distance information from a VOR/DME facility).

**structure**

*part of* Aircraft; Ground Vehicle

**instance connection**  Vehicle Navigation System object (0, m) <-> Navigation System object (0, m)

**message connections**

*message*  Perform Navigation, sent from the Pilot object which is part of the same Aircraft object as the Vehicle Navigation System object

*message*  Perform Navigation, sent from the Ground Vehicle Operator object which is part of the same Ground Vehicle object as the Vehicle Navigation System object

*message*  Request Accept Navigational Guidance, sent from the Navigational System object with which an instance connection exists

*message*  Request Navigational Data, sent to the Navigational System object with which an instance connection exists

**object states**  The states of a Vehicle Navigation System object (from an ATC point of view) are "active" and "inactive." Unless informed otherwise, a manager-type object must assume that the state of a Vehicle Navigation System object is "active."

**attributes**

*attribute*  Equipment Type - the type of navigation system, e.g., VOR/DME-based RNAV system.

*attribute*  Status - whether or not the Vehicle Navigation System object is available for use, e.g., is "active" or "inactive."

**services**  The Vehicle Navigation System class performs the implicit services create, connect, and release. A Vehicle Navigation System object performs the implicit service access.

*service*  Accept Navigational Guidance

- When the aircraft is within the Area of Coverage of a Navigational System object, and the Equipment Type of the Vehicle Navigation System object is compatible with the NAVAID Type of the Navigational System object, and both the Vehicle
Navigation System and the Navigation System have Status "active," accept navigational data from that Navigational System object.

**service Interrogate Navigational Aid**

- When the aircraft is within the Area of Coverage of a Navigational System object, and the Equipment Type of the Vehicle Navigation System object is compatible with the NAVAID Type of the Navigational System object, and both the Vehicle Navigation System and the Navigation System have Status "active," request navigational data from that Navigational System object.

**service Navigate**

- Upon request of the Ground Vehicle Operator object which is part of the same Vehicle object as the Vehicle Navigation System object, guide the ground vehicle.

**external input** None

**external output** None

**notes** A Ground Vehicle object does not have an instance connection with a Navigation System object, nor does a ground vehicle use an ATC navigation system to navigate.
specification  Vehicle Surveillance System - for an aircraft: the airborne radar beacon receiver/transmitter portion of the Air Traffic Control Radar Beacon System (ATCRBS) which automatically receives radio signals from interrogators on the ground, and selectively replies with a specific reply pulse or pulse group only to those interrogations being received on the mode to which it is set to respond (AIM definition of "transponder"); for a ground vehicle, a transponder analogous to the aircraft transponder, possibly available in the timeframe of interest.

structure

part of  Aircraft; Ground Vehicle

generalization of  Transponder

instance connection  Vehicle Surveillance System object (0, m) <-> Surveillance System object (0, m)

message connection

message  Request Altitude Report, sent from a Surveillance System object with which an instance connection exists

message  Request Identification, sent from a Surveillance System object with which an instance connection exists

message  Send Aircraft Altitude, sent from the Pilot object which is part of the same Aircraft object as the Vehicle Surveillance System object

message  Send Aircraft Identification, sent from the Pilot object which is part of the same Aircraft object as the Vehicle Surveillance System object

message  Send Ground Vehicle Identification, sent from the Ground Vehicle Operator object which is part of the same Ground Vehicle object as the Vehicle Surveillance System object

object states  The states of a Vehicle Surveillance System object are specific to the particular object; e.g., the states (modes) of a Mode A transponder (from an ATC point of view) might be "off," "number," "ident," and "number and ident." (See note [1] below.)

attribute  Equipment Type - the type of system, e.g., Mode A.

services  The Vehicle Surveillance System class performs the implicit services create, connect, and release. A Vehicle Surveillance System object performs the implicit service access.
**service**  Report Altitude

- Upon request by a Surveillance System object or a Pilot object, return the aircraft altitude.

**service**  Report Identification

- Upon request by a Surveillance System object or a Pilot object, return the aircraft identification (e.g., the beacon code).
- Upon request by a Surveillance System object or a Ground Vehicle Operator object, return the aircraft identification (e.g., the beacon code).

**external input**  None

**external output**  None

**notes**  (1) The object states given in the example above are defined from an ATC viewpoint, and do not represent all the functionality of the transponder.  (2) The object classes are not well-defined because the model structure stops at a high level. Further specializations of the Vehicle Surveillance System class could provide classes with well-defined object states.  (3) The Report Altitude service may not be applicable to some objects, e.g., an object with Equipment Type of Mode A.
specification Wind/Weather Product - wind and weather products available to the users of the ATC system.

structure

part of Aviation Weather System

instance connection None

message connection Intrinsic only

object states None

attributes

attribute Product Type - the type of wind/weather product (e.g., area forecast).

attribute Time Period - the period of time covered by the report.

attribute Report Contents - the contents of the report.

services The Wind/Weather Product class performs the implicit services create, connect, and release. A Wind/Weather Product object performs the implicit service access.

external input All Wind/Weather Product objects are created outside the ATC system.

external output None

notes None
APPENDIX B

EXPERIMENT MODEL SPECIFICATIONS

This appendix contains the object specifications for the Experiment model in alphabetical order. All classes in the Experiment model are identified in this specification; however, since most of the high-level classes are borrowed from the ATC model, and many of the attributes and services of the domain-unique classes are inherited from classes in the ATC model, only domain-unique components are defined. References are made to the ATC model and the AERA 2 model where necessary.

The format used is as follows:

- **class**: name (Identify referenced model where applicable)
- **structure**: an identification of the classes and objects directly above and beneath this class and its objects in a generalization-specialization or whole-part structure; unless otherwise stated, a generalization-specialization structure involves classes and a whole-part structure involves objects.
  - **part of**
  - **contains parts**
  - **specialization of**
  - **generalization of**
- **domain-unique instance connections**: an identification of each instance connection which is unique to the Experiment model.
- **domain-unique attributes**: name and a textual description of each attribute which is unique to the Experiment model.
- **domain-unique services**: name and a description (by bulleted list, service chart, or other device) of processing performed by each domain-unique service.
- **notes**: information about why this class was included in the Experiment model, or any other additional information about the class or object which is not already provided.
specification  90-Minute Flying-Time Wedge

structure

specialization of Airspace Volume

domain-unique attributes

attribute  Boundary - wedged-shaped airspace volume extending a full 90 minutes flying time starting outside the New York TRACON and extending southeast toward and past the Washington area airports.

attribute  Demand (Inherited)

attribute  Load (Inherited)

attribute  Capacity (Inherited)

domain-unique services  None

notes  One of the six categories of airspace/ground-type resources identified as a testbed requirement in the experiment definition.

---

specification  Aircraft (Borrowed from ATC model)

structure

part of  Flight; Vehicle

contains parts  Pilot; Vehicle Communications System; Vehicle Navigation System

domain-unique attributes  None

domain-unique services  None

notes  None

---

specification  Aircraft-Aircraft Conflict (Refer to AERA 2 model)

structure

specialization of  Problem

domain-unique attributes  None
domain-unique services  None

notes  For the experiment, this class identifies one type of problem that AERA 2 will have to contend with in the en route airspace.

specification  Aircraft-Airspace Conflict (Refer to AERA 2 model)
structure
    specialization of  Problem

domain-unique attributes  None
domain-unique services  None

notes  For the experiment, this class identifies one type of problem that AERA 2 will have to contend with in the en route airspace.

specification  Air/Ground Communications System (Borrowed from ATC model)
structure
    specialization of  Communications System

    generalization of  Data Link-Like Communications System; Voice Radio-Like Communications System

domain-unique attributes  None
domain-unique services  None

notes  None

specification  Airport Movement Area (Borrowed from ATC model)
structure
    part of  Airspace/Ground Resource

    generalization of  La Guardia Airport; Washington Area Airport

domain-unique attributes  None
domain-unique services  None

notes  For the experiment, airport movement areas were identified as sources and sinks of flights.

specification  Airspace/Ground Resource (Borrowed from ATC model)
structure

  specialization of  Resource

  contains parts  Controlled Airspace; Airport Movement Area

domain-unique attributes  None

domain-unique services  None

notes  None

specification  Airspace Surveillance System (Borrowed from ATC model)
structure

  specialization of  Surveillance System

  generalization of  ARTS-Like Terminal Surveillance System; RDP-Like En Route Surveillance System

domain-unique attributes  None

domain-unique services  None

notes  None

specification  Airspace Volume (Borrowed from ATC model)
structure

  part of  En Route Controlled Airspace

  generalization of  90-Minute Flying-Time Wedge

domain-unique attributes  None
domain-unique services  None

notes  None

specification  Air Traffic ("Traffic Pattern")

structure

specialization of  Traffic

generalization of  Experiment Scenario Traffic Pattern

domain-unique attributes  None

domain-unique services  None

notes  (1) The experiment is only interested in interactions associated with traffic in the air. Air traffic is defined in the experiment definition as traffic patterns descriptive of the AERA 2 time-frame. (2) The experiment model definition of the term "traffic pattern" is more general than the definition from the Federal Aviation Regulations: "the traffic flow that is prescribed for aircraft landing at, taxiing on, or taking off from an airport."

specification  Air Traffic Manager (Borrowed from ATC model)

structure

specialization of  Traffic Manager

generalization of  Regional Air Traffic Manager

domain-unique attributes  None

domain-unique services  None

notes  None

specification  Arrival Clearance (Borrowed from ATC model)

structure

part of  Flight Clearance
domain-unique attributes  None

domain-unique services  None

notes  Upon handoff from an en route controller to a terminal controller and receipt of the arrival clearance, the flight changes state from en route to arrival.

specification  ARTS-Like Terminal Surveillance System

structure

specialization of  Airspace Surveillance System

domain-unique attributes  None

domain-unique services

service  Provide User Interface - provide the Plan View Display (PVD) with data blocks and conflict alert.

service  Provide TAAS Capabilities

notes  Prior to the start of the experiment, the necessary radar data processing automation must be established in the terminal area. Terminal automation with TAAS capabilities and an ARTS-like user interface is specifically required by the experiment.

specification  Clearance (Borrowed from ATC model)

structure

generalization of  Flight Clearance

domain-unique attributes  None

domain-unique services  None

notes  None
specification  Communications Resource (Borrowed from ATC model)

structure

specialization of  Resource

contains parts  Communications System

domain-unique attributes  None

domain-unique services  None

notes  None

specification  Communications System (Borrowed from ATC model)

structure

part of  Communications Resource

generalization of  Air/Ground Communications System

domain-unique attributes  None

domain-unique services  None

notes  None

specification  Controlled Airspace (Borrowed from ATC model)

structure

part of  Airspace/Ground Resource

contains parts  Enroute Controlled Airspace; Terminal Controlled Airspace

domain-unique attributes  None

domain-unique services  None

notes  None
**specification**  Data Link-Like Communications System

**structure**

**specialization of**  Air/Ground Communications System

**domain-unique attributes**  None

**domain-unique services**

**service**  Provide Data Link-Like Services

**notes**  A Data Link-Like Communications System object provides a mechanism for electronic communications between the controller and the aircraft (pilot).

---

**specification**  Departure Clearance (Borrowed from ATC model)

**structure**

**part of**  Clearance

**domain-unique attributes**  None

**domain-unique services**  None

**notes**  Upon receipt of the departure clearance, the flight changes state from predeparture to departure.

---

**specification**  En Route Clearance (Borrowed from ATC model)

**structure**

**part of**  Clearance

**domain-unique attributes**  None

**domain-unique services**  None

**notes**  Upon handoff from a terminal controller to an en route controller and receipt of the en route clearance, the flight changes state from departure to en route.

---

**specification**  En Route Controlled Airspace (Borrowed from ATC model)

B-8
structure
    part of  Controlled Airspace
    contains parts  Route; Airspace Volume

domain-unique attributes  None

domain-unique services  None

notes  The Location attribute is of interest to the experiment since the meter fix points will be
located on the boundary between the en route controlled airspace and the terminal
controlled airspace.

specification  En Route Controller

structure
    specialization of  Flight Manager

domain-unique attributes  None

domain-unique services

  service  Identify Conflicts - AERA 2 service; three types of problems identified of interest
to the experiment are aircraft-aircraft conflict, flow instruction non-compliance
(meter fix violations only), and aircraft-airspace conflict.

  service  Propose Resolutions - AERA2 service; the highest-ranked resolution will always
be delivered to and accepted by the aircraft.

  service  Compute Delay-Absorbing Maneuvers - this AREA 2 service will compute the
delay-absorbing maneuvers.

  service  Handoff Aircraft to Terminal Controller - the automated en route controller will
perform this service.

  service  Issue Delay-Absorbing Maneuvers - the automated en route controller will issue
delay-absorbing maneuvers.

notes  For the experiment, the En Route Controller will be automated. The services described are
identified in the experiment definition; however, the service names do not indicate whether
the service is performed by the automated human controller or AERA 2.
**specification**  Experiment Scenario Traffic Pattern

**structure**

**specialization of**  Air Traffic ("Traffic Pattern")

**domain-unique attributes**

**attribute**  Demand - a unique number which represents the projected concentration of flights.

**domain-unique services**  None

**notes**  Varying the Demand attribute throughout the experiment allows the experiment interactions to vary: heavy demand results in many interactions, less demand results in less interactions.

---

**specification**  Fix/Waypoint (Borrowed from ATC model)

**structure**

**part of**  Route

**generalization of**  Meter Fix

**domain-unique attributes**  None

**domain-unique services**  None

**notes**  None

---

**specification**  Flight (Borrowed from the ATC model)

**structure**

**part of**  Air Traffic ("Traffic Pattern")

**contains parts**  Aircraft; Flight Plan; Flight Clearance

**domain-unique attributes**  None

**domain-unique services**  None

**notes**  None
specification  Flight Clearance (Borrowed from the ATC model)

structure

specialization of  Clearance

part of  Flight

contains parts  Departure Clearance;  Enroute Clearance;  Arrival Clearance

domain-unique attributes  None

domain-unique services  None

notes  None

specification  Flight Manager (Borrowed from ATC model)

structure

generalization of  Enroute Controller;  Terminal Controller;  Tower Controller

specialization of  Manager

domain-unique attributes  None

domain-unique services  None

notes  None

specification  Flight Plan (Borrowed from ATC model)

structure

part of  Flight

domain-unique attributes

attribute  Expected Meter Fix Time  - expected time the flight will cross a meter fix.

attribute  Expected Airport Arrival Time - expected time the flight will arrive at the destination airport.
attribute  Flow Restriction Time - the time at which the flow restriction was issued.

domain-unique services  None

notes  The flight plan attributes identified are arrival times at the meter fix and at the destination airport. By running the metering program, these times are adjusted. The AERA 2 model contains a much more detailed description of the flight plan.

specification  Flow Instruction

structure  None

domain-unique instance connections

instance connection  Flow Instruction (1) <-> Meter Fix (1)

domain-unique attributes  None

domain-unique services  None

notes  For the experiment, the only type of flow instruction is produced by metering.

specification  Flow Instruction Non-Compliance (Refer to AERA 2 model)

structure

specialization of  Problem

domain-unique attributes  None

domain-unique services  None

notes  For the experiment, the only type of flow instruction non-compliance is a meter fix time violation.

specification  Highest-Ranked Resolution (HRR) (Refer to AERA 2 model)

structure

specialization of  Resolution

domain-unique attributes  None
domain-unique services  None

notes  For the experiment, the highest-ranked resolution is identified by AERA 2 and is always delivered to and accepted by the aircraft.

specification  ILS-Like Landing Navigation System
structure
specialization of  Landing Navigation System
domain-unique attributes  None
domain-unique services
service  Provide ILS Data to Aircraft.

notes  ILS navigational equipment on the ground is needed to communicate with the ILS navigational equipment on the aircraft, at La Guardia airport only.

specification  ILS Route
structure
specialization of  Terminal Route
domain-unique attributes  None
domain-unique services  None

notes  Predefined ILS routes will be identified in the La Guardia terminal area.

specification  La Guardia Airport
structure
specialization of  Airport Movement Area
domain-unique attributes
attribute  Arrival Capacity
attribute  Departure Capacity

attribute  Runway Configuration

attribute  Expected Departure Rate

domain-unique services  None

notes  La Guardia Airport must be modelled with sufficient detail to permit a tower controller to coordinate departures with the terminal area arrival controller(s) so that the arrival stream is spaced appropriately to allow for departures. Aircraft on the ground will request departures at a rate consistent with the airport arrival capacity used by the metering program.

specification  Landing Navigation System (Borrowed from ATC model)

structure

specialization of  Navigation System

generalization of  ILS-Like Landing Navigation System;  MLS-Like Landing Navigation System

domain-unique attributes

attribute  Type - ILS-Like or MLS-Like

domain-unique services  None

notes  None

specification  Manager (Borrowed from ATC model)

structure

generalization of  Traffic Manager; Flight Manager

domain-unique attributes  None

domain-unique services  None

notes  Base-level class of the Manager subject.
**specification**  Meter Fix

**structure**

*specialization of*  Fix/Waypoint

**domain-unique instance connections**

*instance connection*  Meter Fix (1) <-> Flow Instruction (1)

**domain-unique attributes**

*attribute*  Location - somewhere on the en route and terminal area boundary

**domain-unique services**  None

**notes**  For the experiment, a meter fix is a fix or waypoint which resides at the boundary between the en route airspace and the terminal airspace to which aircraft are metered.

**specification**  MLS-Like Landing Navigation System

**structure**

*specialization of*  Landing Navigation System

**domain-unique attributes**  None

**domain-unique services**

*service*  Provide MLS data to aircraft.

**notes**  MLS navigational equipment on the ground is needed to communicate with the MLS navigational equipment on the aircraft, at La Guardia airport only.

**specification**  MLS Route

**structure**

*specialization of*  Terminal Route

**domain-unique attributes**  None

**domain-unique services**  None
notes  Predefined MLS routes will be defined in the La Guardia terminal area.

---

specification  Multiple Problem (Refer to AERA 2 model)
structure
    contains parts  Problem
    domain-unique attributes  None
    domain-unique services  None
    notes  None

---

specification  Navigation Resource (Borrowed from ATC model)
structure
    specialization of  Resource
    contains parts  Navigation System
    domain-unique attributes  None
    domain-unique services  None
    notes  None

---

specification  Navigation System (Borrowed from ATC model)
structure
    part of  Navigation Resource
    generalization of  Landing Navigation System
    domain-unique attributes  None
    domain-unique services  None
    notes  None
specification  New York TRACON

structure

specialization of  Terminal Airspace Volume

domain-unique attributes  Boundary

domain-unique services  None

notes  The selected terminal area of interest is the New York TRACON. Meter fixes are located on the boundary, and arrival and departures of interest occur at La Guardia airport.

specification  Pilot (Borrowed from ATC model)

structure

part of  Aircraft

domain-unique attributes  None

domain-unique services

service  Receive/Implement Vectoring Instructions - this service can either be automated or performed by a simulated pilot (sim-pilot).

notes  En route pilot services are automated; terminal area services can either be automated or performed by a sim-pilot.

specification  Point (Refer to AREA 2 model)

structure

part of  Trajectory

domain-unique attributes  None

domain-unique services  None

notes  A point is a part of the trajectory created by AERA 2 to check for conflicts and meter fix time violations.

specification  Problem (Refer to AERA 2 model)

B-17
structure

part of  Multiple Problem

generalization of  Aircraft-Aircraft Conflict; Flow Instruction Non-Compliance; Aircraft-Airspace Conflict

domain-unique attributes

attribute  Subject Aircraft - the aircraft for which the problem was detected.

domain-unique services  None

notes  None

specification  RDP-Like En Route Surveillance System

structure

specialization of  Airspace Surveillance System

domain-unique attributes  None

domain-unique services

service  Provide Radar Data Processing (RDP) Services for En Route Airspace

notes  None

specification  Regional Air Traffic Manager

structure

specialization of  Air Traffic Manager

domain-unique attributes  None

domain-unique services

service  Adjust Flight Schedule - perform the metering program function of adjusting the schedule in response to problems. Adjustments may occur to allow for the inability of a flight to meet its meter fix time. Adjustments may include causing the metering
program to create a new schedule based on the latest estimates of undelayed arrival times.

service  Create Metering Flow Restriction - perform the metering program function of specifying the time a flight should cross a meter fix.

service  Impose Flow Instruction - perform the metering program function of passing the meter fix times to AERA 2 as flow instructions.

service  Sequence/Resequence Flight - perform the metering program function of flight sequencing, based on undelayed arrival times; resequencing may occur when adjustments are made to allow for the inability of a flight to meet its meter fix time.

service  Monitor - performed by TMC

notes  The services are described from the experiment definition; however, the service names do not indicate whether the service is performed by a human or by automation - as far as the model is concerned, who or what performs the services is not important.

---

specification  Resolution (Refer to AERA 2 model)

structure

  generalization of  Highest-Ranked Resolution (HRR)

  domain-unique attributes  None

  domain-unique services  None

notes  None

---

specification  Resource

structure

  generalization of  Airspace/Ground Resource; Surveillance Resource; Communications Resource

  domain-unique attributes  None

  domain-unique services  None

notes  Base-level class of the Resource subject.
**specification**  Route (Borrowed from ATC model)

**structure**

- **part of**  En Route Controlled Airspace
- **contains parts**  Fix/Waypoint

**domain-unique attributes**  None

**domain-unique services**  None

**notes**  The experiment definition states that the current concept for traffic management in the AAS is that there will be a series of time control points (Fix/Waypoints in the ATC model) along a flights path (Route in the ATC model).

---

**specification**  Segment (Refer to AERA 2 model)

**structure**

- **part of**  Trajectory

**domain-unique attributes**  None

**domain-unique services**  None

**notes**  A segment is a part of the trajectory created by AERA 2 to check for conflicts and meter fix time violations.

---

**specification**  Surveillance Resource (Borrowed from ATC model)

**structure**

- **specialization of**  Resource
- **contains parts**  Surveillance System

**domain-unique attributes**  None

**domain-unique services**  None

**notes**  None
specification  Surveillance System (Borrowed from ATC model)

structure

    part of  Surveillance Resource

    generalization of  Airspace Surveillance System

domain-unique attributes  None

domain-unique services  None

notes  None

---

specification  Terminal Airspace Volume (Borrowed from ATC model)

structure

    part of  Terminal Controlled Airspace

    generalization of  New York TRACON

domain-unique attributes  None

domain-unique services  None

notes  None

---

specification  Terminal Controlled Airspace (Borrowed from ATC model)

structure

    part of  Controlled Airspace

contains parts  Terminal Route; Terminal Airspace Volume

domain-unique attributes  None

domain-unique services  None

notes  The Location attribute is of interest to the experiment since the meter fix point will be located on the boundary between the en route controlled airspace and the terminal controlled airspace.
specification  Terminal Controller

structure

specialization of  Flight Manager

domain-unique attributes  None

domain-unique services

service  Final Arrival Sequencing and Spacing

service  Coordinate Arrivals With Departures

service  Accept/Initiate Handoffs

notes  For the experiment, one or two of the arrival positions will be staffed.

specification  Terminal Route

structure

part of  Terminal Controlled Airspace

generalization of  ILS Route; MLS Route

domain-unique attributes  None

domain-unique services  None

notes  None

specification  Tower Controller

structure

specialization of  Flight Manager

domain-unique attributes  None

domain-unique services
service  Coordinate Arrivals with Terminal Controller

service  Coordinate Departures with Terminal Controller

notes  A tower controller is identified at La Guardia Airport to coordinate departures with the TRACON arrival controller (terminal controller in this model), so that the arrival stream is spaced appropriately to allow for departures.

---

specification  Traffic (Borrowed from ATC model)

structure

  specialization of  User

  generalization of  Air Traffic ("Traffic Pattern")

domain-unique attributes  None

domain-unique services  None

notes  None

---

specification  Traffic Manager (Borrowed from ATC model)

structure

  specialization of  Manager

  generalization of  Air Traffic Manager

domain-unique attributes  None

domain-unique services  None

notes  None

---

specification  Trajectory

structure

  contains parts  Point; Segment

domain-unique attributes  None
domain-unique services  None

notes  Trajectories are the projected aircraft paths determined by AERA 2 from flight plan information for the purposes of identifying conflicts and meter fix time violations. A trajectory is composed of a series of points and segments.

specification  User (Borrowed from ATC model)
structure

    generalization of  Manager; Traffic

domain-unique attributes  None
domain-unique services  None

notes  Base class of the User subject.

specification  Vehicle Communications System (Borrowed from ATC model)
structure

    part of  Aircraft

domain-unique attributes  None
domain-unique services  None

notes  None

specification  Vehicle Navigation System (Borrowed from ATC model)
structure

    part of  Aircraft

domain-unique attributes  None
domain-unique services  None
notes None

specification Voice Radio-Like Communications System

structure

specialization of Air/Ground Communications System
domain-unique attributes None
domain-unique services None

notes Voice radio-like communications are necessary in the terminal area for communication between terminal area controllers and sim-pilots.

specification Washington Area Airport

structure

specialization of Airport Movement Area
domain-unique attributes

attribute Runway Configuration

attribute Expected Departure Rate
domain-unique services None

notes Washington area airports are contained in the 90-minute flying-time wedge and several of the other airspace categories defined in the experiment definition. These are identified as sources and sinks of flights.
APPENDIX C
TASF MODEL SPECIFICATIONS

This appendix contains the class and object specifications for the TASF model in alphabetical order. All classes in the TASF model are identified in this specification; however, since most of the high-level classes are borrowed from the ATC model, and many of the attributes and services of the domain-unique classes are inherited from classes in the ATC model, only domain-unique components are defined. References are made to the ATC model and the AERA 2 model where necessary.

The format used is as follows:

class - name (Identify referenced model where applicable)

structure - an identification of the classes and objects directly above and beneath this class and its objects in a generalization-specialization or whole-part structure; unless otherwise stated, a generalization-specialization structure involves classes and a whole-part structure involves objects.

part of
contains parts
specialization of
generalization of

domain-unique instance connections - an identification of each instance connection which is unique to the TASF model.

domain-unique attributes - a list of the names of the attributes which are unique to the TASF model.

domain-unique services - a list of the names of the services which are unique to the TASF model.

notes - information about why this class was included in the TASF model, or any other additional information about the class or object which is not already provided.
**specification**  Aircraft

**structure**

part of  Flight

contains parts  Pilot; Vehicle Navigation System; Vehicle Communications System

domain-unique attributes

Altitude  
Bank Angle  
Climb Rate  
Descent Rate  
Final Approach Speed  
Final Approach Slowdown Point  
Heading  
Speedup Acceleration  
Slowdown Acceleration  
Speed  
Turn Rate  
Velocity in x Direction  
Velocity in y Direction  
Velocity in z Direction  
Weight Class  
x-Position  
y-Position  
z-Position

domain-unique services

Accept Aircraft Point Position  
Accept Aircraft Route Information  
Accept Aircraft Route Segment Information  
Calculate Aircraft Point Position  
Send Aircraft Point Position to ARTS

**notes**  The attributes listed above are data items found in an Aircraft data structure; the services are abstracted from the trajectory modeler functions.

---

**specification**  Aircraft ILS Navigation System

**structure**
specialization of  Vehicle Navigation System

domain-unique attributes

Altitude Lock
Altitude Phase
Heading Lock
Heading Phase
Localizer Detected
Speed Lock
Speed Phase
Time of Reading

domain-unique services

Accept/Implement Maneuver Information

notes  Attributes are found in a data structure. This class is included in the model since ILS navigational equipment exists on those aircraft flying an ILS approach to La Guardia.

specification  Aircraft MLS Navigation System

structure

specialization of  Vehicle Navigation System

domain-unique attributes  None

domain-unique services

Receive MLS position data

notes  This class is included in the model since MLS navigational equipment exists on those aircraft flying an MLS approach to La Guardia.

specification  Air/Ground Communications System  (Borrowed from ATC model)

structure

specialization of  Communications System

generalization of  Data Link Communications System; Voice Radio Communications System
specification  Airport Movement Area  (Borrowed from ATC model)

structure

   part of  Airspace/Ground Resource

   contains parts  Runway

   generalization of  La Guardia Airport; Teterboro Airport

domain-unique attributes  None

domain-unique services  None

notes  None

specification  Airspace/Ground Resource  (Borrowed from ATC model)

structure

   specialization of  Resource

   contains parts  Controlled Airspace; Airport Movement Area

domain-unique attributes  None

domain-unique services  None

notes  None

specification  Airspace Surveillance System  (Borrowed from ATC model)

structure

   specialization of  Surveillance System

   generalization of  ARTS III
domain-unique attributes None
domain-unique services None
notes None

specification Airspace Volume (Borrowed from ATC model)
structure
  part of En Route Controlled Airspace
  generalization of Holding Pattern Airspace ("Holding Pattern")
domain-unique attributes None
domain-unique services None
notes None

specification Air Traffic (Borrowed from ATC model)
structure
  specialization of Traffic
  contains parts Flight
domain-unique attributes None
domain-unique services None
notes None

specification Arrival Clearance (Borrowed from ATC model)
structure
  part of Flight Clearance
domain-unique attributes None
domain-unique services None

notes Upon handoff from an en route controller to one of the arrival controllers (North or South) and receipt of the arrival clearance, the flight changes state from en route to arrival.

---

**specification** ARTS III

**structure**

specialization of Airspace Surveillance System
generalization of ARTS III With Ghosting
domain-unique attributes None
domain-unique services

Display Aircraft Location

notes One of the simulated models of TASF. It displays the actual location of the aircraft as calculated by the trajectory modeler (for the purposes of the TASF model, the trajectory modeler functions are abstracted as services in the Aircraft object).

---

**specification** ARTS III With Ghosting

**structure**

specialization of ARTS III
domain-unique attributes None
domain-unique services

Project MLS Flight Onto ILS Route

notes The ghosting aid provides the final approach controllers with a tool for spacing and sequencing flights for landing at La Guardia Airport.

---

**specification** Clearance (Borrowed from ATC model)

**structure**

generalization of Flight Clearance
domain-unique attributes None
domain-unique services None
notes None

---

specification Communications Resource (Borrowed from ATC model)
structure
   specialization of Resource
domain-unique attributes None
domain-unique services None
notes None

---

specification Communications System (Borrowed from ATC model)
structure
   part of Communications Resource
generalization of Air/Ground Communications System
domain-unique attributes None
domain-unique services None
notes None

---

specification Controlled Airspace (Borrowed from ATC model)
structure
   part of Airspace/Ground Resource
contains parts En Route Controlled Airspace; Terminal Controlled Airspace
domain-unique attributes None
domain-unique services None

notes None

specification Data Link Communication System

structure

specialization of Air/Ground Communications System

domain-unique attributes

Comment
Message Type
Status
Type (Mode S/ACARS)
Time of Message

domain-unique services None

notes Attribute found in defined data structure.

---

specification Departure Clearance (Borrowed from ATC model)

structure

part of Flight Clearance

domain-unique attributes None

domain-unique services None

notes A departure clearance is given while the aircraft is on the ground; however, the cleared route extends into the terminal area.

---

specification Departure Controller

structure

specialization of Terminal Flight Manager

domain-unique attributes None

---

C-8
**domain-unique services**

Accept Flights from Local Controller (no coordination)
Space Aircraft on Departure Routes

**notes** No coordination existed between the local controller and the departure controller in the six-month illustration. Aircraft departed from the ground and, within a certain time parameter, appeared on the final approach/departure display in the terminal area. For the six-month illustration, the departure controller position was automated.

**specification** En Route Controlled Airspace (Borrowed from ATC model)

**structure**

**part of** Controlled Airspace

**contains parts** Route ("En Route Airway"); Airspace Volume

**domain-unique attributes**

Boundary Code
Description Code
En Route Waypoint Identification
ICAO Code
NAVAID Identification
Route Identification
Sequence Number
Waypoint Identification

**domain-unique services** None

**notes** En route airspace was modelled to some degree in TASF for testing interface issues in the stand-alone mode.

**specification** En Route Flight Manager

**structure**

**specialization of** Flight Manager

**domain-unique attributes** None

**domain-unique services** None
notes None

specification Final Approach Controller

structure

specialization of Terminal Flight Manager

domain-unique attributes None

domain-unique services

Issue Clearance Directives to Pilot
Space Aircraft On Arrival Route

notes This was the only staffed position in the six-month illustration. The final approach controller used the ghosting aid to space and sequence flights for landing at La Guardia. Clearance directives were issued to ILS aircraft only.

specification Fix/Waypoint ("En Route Waypoint")

structure

part of Route ("En Route Airway")

domain-unique attributes

Holding Pattern Identification
ICAO Code
Magnetic Variation
Region Code
Waypoint Identification
Waypoint Type
Waypoint Usage

domain-unique services None

notes None

specification Flight

structure
part of Air Traffic

contains parts Flight Clearance; Flight Plan; Aircraft

domain-unique attributes

Airline
Flight Number

domain-unique services None

notes None

specification Flight Clearance (Borrowed from ATC model)

structure

specialization of Clearance

contains parts Departure Clearance; Arrival Clearance

domain-unique attributes None

domain-unique services None

notes None

specification Flight Manager (Borrowed from ATC model)

structure

specialization of Manager

generalization of Terminal Flight Manager; En Route Flight Manager

domain-unique attributes None

domain-unique services None

notes None

specification Flight Plan (Borrowed from ATC model)
structure

    part of  Flight

domain-unique attributes

    Arrival Airport
    Departure Airport
    Departure Fix
    Flight Type
    Runway

domain-unique services  None

notes  The attributes listed above were items in a Flight Plan data structure. TASF does not use proposed route data to predict future positions of the flight.

specification  Holding Pattern Airspace ("Holding Pattern")

structure

    specialization of  Airspace Volume

domain-unique attributes

    ICAO Code
    Inbound Holding Course
    Leg Length
    Leg Time
    Maximum Altitude
    Region Code
    Turn
    Waypoint Identification

domain-unique services  None

notes  None

specification  ILS Marker Beacon

structure

    part of  Instrument Landing System (ILS)
domain-unique attributes

- Airport Identification
- Facility Character
- Facility Elevation
- Frequency
- ICAO Code
- Localizer Identification
- Locator Location
- Magnetic Variation
- Marker Location
- Marker Type - Outer Marker Beacon or Middle Marker Beacon
- Minor Axis
- NAVAID Class
- Runway Identification

domain-unique services  None

notes  TASF models ILS marker beacons with a data structure which contains the attributes listed above. Marker beacons provide additional position information to pilots conducting an ILS approach.

specification  ILS Route

structure

specialization of  Terminal Route

domain-unique attributes  None

domain-unique services  None

notes  For the 6-MITF, TASF defined an ILS route at La Guardia Airport for landing on runway 13.

specification  Instrument Landing System (ILS)

structure

specialization of  Landing Navigation System

contains parts  ILS Marker Beacon

domain-unique attributes
Glide Slope Angle
Glide Slope Elevation
Glide Slope Latitude
Glide Slope Longitude
Glide Slope x
Glide Slope y
Localizer Bearing
Localizer Latitude
Localizer Longitude
Localizer Name
Localizer Width
Localizer x (nmi)
Localizer y (nmi)

domain-unique services

Provide Distance to Aircraft Avionics
Provide Glide Slope to Aircraft Avionics

notes None

specification La Guardia Airport

structure

specialization of Airport Movement Area
contains parts La Guardia Runway 13
domain-unique attributes None
domain-unique services None

notes For the 6-MITF, La Guardia Airport was the center of operations in the TRACON.

specification La Guardia Airspace Volume

structure

specialization of Terminal Airspace Volume
domain-unique attributes None
domain-unique services  None

notes  None

specification  La Guardia Runway 13

structure

specialization of  Runway

part of  La Guardia Airport

domain-unique attributes  None

domain-unique services  None

notes  Runway 13 was the only active runway in the six-month illustration.

specification  Landing Navigation System (Borrowed from ATC model)

structure

specialization of  Navigation System

generalization of  Instrument Landing System (ILS); Microwave Landing System (MLS)

domain-unique attributes  None

domain-unique services  None

notes  None

specification  Manager (Borrowed from ATC model)

structure

specialization of  User

generalization of  Flight Manager

domain-unique attributes  None

domain-unique services  None
specification  Microwave Landing System (MLS)

structure

  specialization of  Landing Navigation System

domain-unique attributes  None

domain-unique services

  Provide x-y-z Positions to Aircraft Avionics

notes  None

_____________________________________________________

specification  MLS Route

structure

  specialization of  Terminal Route

domain-unique attributes  None

domain-unique services  None

notes  For the 6-MITF, TASF defined an MLS route at La Guardia Airport for landing on runway 13.

_____________________________________________________

specification  Navigation Resource  (Borrowed from ATC model)

structure

  specialization of  Resource

  contains parts  Navigation System

domain-unique attributes  None

domain-unique services  None

notes  None
specification  Navigation System  (Borrowed from ATC model)

structure

  part of  Navigation Resource

  generalization of  Landing Navigation System

domain-unique attributes  None

domain-unique services  None

notes  None

specification  New York TRACON

structure

  specialization of  Terminal Airspace Volume

domain-unique attributes  None

domain-unique services  None

notes  None

specification  North Arrival Controller

structure

  specialization of  Terminal Flight Manager

domain-unique attributes  None

domain-unique services

Accept North Arrival Flights from En Route Flight Manager (AGD)

notes  En route flights were handed off to TASF from the en route part of the Adaptive Ground Delay program (AGD) software with no coordination.

C-17
specification  Pilot

structure

part of  Aircraft

domain-unique attributes  None

domain-unique services

Accept Clearance Directives From Controller
Maneuver Aircraft

notes  TASF had the capability to accept input from a simulated pilot ("sim-pilot") which was
used to maneuver flights for landing at La Guardia Airport.

specification  Resource

structure

generalization of  Airspace/Ground Resource; Surveillance Resource; Navigation
Resource; Communications Resource

domain-unique attributes  None

domain-unique services  None

notes  None

specification  Route ("En Route Airway")

structure

part of  En Route Controlled Airspace

contains parts  Fix/Waypoint ("En Route Waypoint"); Route Segment

domain-unique attributes

Boundary Code
Description Code
En Route Waypoint Identification
ICAO Code
NAVAID Identification
Route Identification
Sequence Number
Waypoint Identification

domain-unique services None

notes None

specification Route Segment

structure

part of Route ("En Route Airway")

domain-unique attributes None

domain-unique services None

notes None

specification Runway

structure

part of Airport Movement Area

generalization of La Guardia Runway 13

domain-unique attributes

Airport Identification
Approach Identification
Description
ILS Identification
Landing Threshold Elevation
Length
Localizer Identification
Magnetic Bearing
Runway Identification
Stopway
Threshold Displace Distance
Threshold Cross Height
Width
domain-unique services None

notes None

specification SID (Borrowed from the ATC model)

structure

specialization of Terminal Route

domain-unique attributes

Distance to Waypoint
SID Identification

domain-unique services None

notes TASF identifies SIDs. The attributes listed above are a subset of the data items found in a defined data structure.

specification South Arrival Controller

structure

specialization of Terminal Flight Manager

domain-unique attributes None

domain-unique services

Accept South Arrival Flights From En Route Flight Manager (AGD)

notes En route flights were handed off to TASF from the en route part of the Adaptive Ground Delay program (AGD) software with no coordination.

specification STAR (Borrowed from the ATC model)

structure

specialization of Terminal Route

domain-unique attributes
STAR Identification

domain-unique services

notes TASF identifies STARs. The attributes listed above are a subset of the data items found in a defined data structure.

specification Surveillance Resource (Borrowed from ATC model)
structure

specialization of Resource
contains parts Surveillance System
domain-unique attributes None
domain-unique services None

notes None

specification Surveillance System (Borrowed from ATC model)
structure

part of Surveillance Resource
generalization of Airspace Surveillance System
domain-unique attributes None
domain-unique services None

notes None

specification Terminal Airspace Volume (Borrowed from ATC model)
structure

part of Terminal Controlled Airspace
generalization of La Guardia Airspace Volume; Newark Airspace Volume

C-21
domain-unique attributes None
domain-unique services None
notes None

specification Terminal Controlled Airspace (Borrowed from ATC model)
structure
part of Controlled Airspace
contains parts Terminal Route; Terminal Airspace Volume

domain-unique attributes None
domain-unique services None
notes None

specification Terminal Flight Manager
structure
specialization of Flight Manager
generalization of North Arrival Controller; South Arrival Controller; Departure Controller; Final Approach Controller

domain-unique attributes None
domain-unique services None
notes None

specification Terminal Route (Borrowed from ATC model)
structure
part of Terminal Controlled Airspace
generalization of SID; STAR; ILS Route; MLS Route

C-22
domain-unique attributes

Airport Identification
ICAO Code
Route Type
Speed Limit
Transition Identification
Vertical Angle

domain-unique services  None

notes  None

specification  Teterboro Airport

structure

specialization of  Airport Movement Area

domain-unique attributes  None

domain-unique services  None

notes  None

specification  Traffic

structure

specialization of  User

generalization of  Air Traffic

domain-unique attributes  None

domain-unique services  None

notes  None
specification  User

structure

generalization of  Manager; Traffic

domain-unique attributes  None
domain-unique services  None

notes  None

specification  Vehicle Communications System  (Borrowed from ATC model)

structure

part of  Aircraft

domain-unique attributes  None
domain-unique services  None

notes  None

specification  Vehicle Navigation System  (Borrowed from ATC model)

structure

part of  Aircraft

generalization of  Aircraft ILS Navigation System; Aircraft MLS Navigation System
domain-unique attributes  None
domain-unique services  None

notes  None

specification  Voice Radio Communication System

structure

specialization of  Air/Ground Communications System
domain-unique attributes None

domain-unique services None

notes None
APPENDIX D

AERA 2 MODEL SPECIFICATIONS

This appendix contains the object specifications for the AERA 2 model in alphabetical order. All classes in the AERA 2 model are identified in this specification; however, since most of the high level classes are borrowed from the ATC model, and many of the attributes and services of the domain-unique classes are inherited from classes in the ATC model, only domain-unique components are defined. References are made to the ATC model where necessary.

The format used is as follows:

*class* - name (Identify referenced model where applicable).

*structure* - an identification of the classes and objects directly above and beneath this class and its objects in a generalization-specialization or whole-part structure; unless otherwise stated, a generalization-specialization structure involves classes and a whole-part structure involves objects.

- part of
- contains parts
- specialization of
- generalization of

*domain-unique instance connections* - an identification of each instance connection which is unique to the AERA 2 model.

*domain-unique attributes* - name and a textual description of each attribute which is unique to the AERA 2 model.

*domain-unique services* - name and a description (by bulleted list, service chart, or other device) of processing performed by each domain-unique service.

*notes* - information about why this class was included in the AERA 2 model, or any other additional information about the class or object which is not already provided.
specification  ACF Airspace ("Planning Region") (Borrowed from ATC model)

structure

specialization of  Airspace Volume

contains part  Sector

domain-unique instance connections

ACF Airspace ("Planning Region") (1) <-> Restriction (0,m)

domain-unique attributes

Sector of Control

domain-unique services

Automated Problem Detection
Automated Resolution Generation

notes  None

specification  Aircraft (Borrowed from ATC model)

structure

part of  Flight

domain-unique instance connections

Aircraft (1) <-> Track (0,m)
Aircraft (1,m) <-> Aircraft Class (1)

domain-unique attributes

Assigned Altitude
Callsign - Aircraft identifier
Emergency Status
Ground Speed - the speed at which the aircraft is moving with respect to the ground.
  Known only to the controller.
Reported Altitude

domain-unique services  None
specification  Aircraft-Aircraft Conflict

structure

specialization of  Problem

domain-unique instance connections  None

domain-unique attributes

Altitude
End of Delay
End Time
Geometry
Minimum Horizontal Separation Time
Notify Sector
Object Aircraft
Object Aircraft End Location
Object Aircraft Start Location
Object Aircraft Start Sector
Separation
Start of Delay
Start Time
Subject Aircraft End Location
Subject Aircraft Start Location
Subject Aircraft Start Sector
Trailing Aircraft

domain-unique services  None

notes  None

specification  Aircraft-Airspace Conflict

structure

specialization of  Problem

domain-unique instance connections  None

domain-unique attributes
Airspace Name
Airspace Type
End Location
Minimum Miss Distance
Separation
Start Location

domain-unique services None

notes None

----------------------------------------

specification Aircraft Class

structure None

domain-unique instance connections

Aircraft Class (1) <-> Aircraft (1,m)
Aircraft Class (1) <-> Climb Profile (1,10)
Aircraft Class (1) <-> Descent Profile (1,10)
Aircraft Class (1) <-> Long Range Cruise Profile (1,7)

domain-unique attributes

Aircraft Model
Aircraft Types
Cruise Acceleration - Cruise acceleration rate (TAS)
Heavy Aircraft Indicator - Indicates heavy aircraft (Jumbo Jet)
Maximum Altitude
Maximum Endurance Speed
Maximum Mach Speed
Maximum Speed
Minimum Speed
Nominal Descent Profile Index
Reference Altitude (for Maximum Speed)
Reference Altitude (for Minimum Speed)

domain-unique services None

notes None

----------------------------------------

specification Aircraft Descent Altitude Profile

structure
specialization of  Terminal Route

domain-unique instance connections

Aircraft Descent Altitude Profile (1,4) <-> Airport (1)

domain-unique attributes

AERA Arrival Fix
Approach Fix
Destination Fix
Required Altitude at Restriction Fix
Restriction Fix
Target Altitude

domain-unique services None

notes None

specification  Air/Ground Communications System (Borrowed from ATC model)

structure

specialization of  Communications System

generalization of  Data Link Communications System

domain-unique instance connections None

domain-unique attributes None

domain-unique services None

notes None

specification  Airport

structure

specialization of  Airport Movement Area

domain-unique instance connections

Airport (1) <-> Aircraft Descent Altitude Profile (1,4)
### Domain-Unique Attributes

- Barometric Pressure
- Elevation
- Name

### Domain-Unique Services

- None

### Notes

- None

---

### Specification

**Airport Movement Area** (Borrowed from ATC model)

**Structure**

- **Part of**: Airspace/Ground Resource
- **Generalization of**: Airport

**Domain-Unique Instance Connections**: None

**Domain-Unique Attributes**: None

**Domain-Unique Services**: None

**Notes**: None

---

### Specification

**Airspace/Ground Resource** (Borrowed from ATC model)

**Structure**

- **Specialization of**: Resource

- **Contains parts**: Controlled Airspace; Airport Movement Area

**Domain-Unique Instance Connections**: None

**Domain-Unique Attributes**: None

**Domain-Unique Services**: None

**Notes**: None
specification  Airspace Volume (Borrowed from ATC model)

structure

   part of  En Route Controlled Airspace

   generalization of  ACF Airspace ("Planning Region"); Sector; Protected Airspace Volume ("Blocked Airspace"); Holding Pattern Airspace ("Holding Pattern")

domain-unique instance connections  None

domain-unique attributes  None

domain-unique services  None

notes  None

specification  APDIA

structure

   specialization of  Terminal Airspace Volume

domain-unique instance connections

   APDIA (0,1) <-> Airport (1)

domain-unique attributes

   Boundry
   End Time
   Maximum Altitude
   Minimum Altitude
   Start Time

domain-unique services  None

notes  The Automated Problem Detection Inhibited Area (APDIA) is a volume of airspace in which automated problem detection is not performed.
specialization of Terminal Airspace Volume

domain-unique instance connections None

domain-unique attributes

Boundary Points (latitude/longitude)
End Time
Maximum Altitude
Minimum Altitude
Name
Start Time

domain-unique services None

notes None

specification Automated Replan Amendment

structure None

domain-unique instance connections

Automated Replan Amendment (1) <-> Automated Replan Plan (1)

domain-unique attributes

Aircraft Identification
Amendment Template

domain-unique services

Request Trial Plan Creation

notes None

specification Automated Replan Plan

structure

specialization of Trial Plan

domain-unique instance connections
Automated Replan Plan (1) ↔ Automated Replan Amendment (1)

domain-unique attributes  None

domain-unique services  None

notes  None

---

specification  Aviation Weather Resource (Borrowed from ATC model)

structure

specialization of  Resource

contains parts  Aviation Weather System

domain-unique instance connections  None

domain-unique attributes  None

domain-unique services  None

notes  None

---

specification  Aviation Weather System (Borrowed from ATC model)

structure

part of  Aviation Weather Resource

contains parts  Wind/Weather Product

domain-unique instance connections  None

domain-unique attributes  None

domain-unique services  None

notes  None

---

specification  Clearance Directive
structure

part of Plan

domain-unique instance connections None

domain-unique attributes

Activation Points
Beginning of Maneuver Points
CD Type
Parameter Values
Source

domain-unique services None

notes None

___________________________________________________________________________

specification Climb Profile

structure None

domain-unique instance connections

Climb Profile (1,10) <-> Aircraft Class (1)

domain-unique attributes

Climb Gradient
Speed
Upper Altitude Bound

domain-unique services None

notes None

___________________________________________________________________________

specification Communications Resource (Borrowed from ATC model)

structure

specialization of Resource

contains parts Communications System
domain-unique instance connections  None

domain-unique attributes  None

domain-unique services  None

notes  None

specification  Communications System (Borrowed from ATC model)

structure

  part of  Communications Resource

  generalization of  Air/Ground Communications System

domain-unique instance connections  None

domain-unique attributes  None

domain-unique services  None

notes  None

specification  Controlled Airspace (Borrowed from ATC model)

structure

  part of  Airspace/Ground Resource

  contains parts  En Route Controlled Airspace; Terminal Controlled Airspace

domain-unique instance connections  None

domain-unique attributes  None

domain-unique services  None

notes  None
specification  Controller Reminder

structure  None

domain-unique instance connections

  Controlled Reminder (0,m) <-> Trajectory (1)

domain-unique attributes

  Aircraft Identification
  Message
  Type

domain-unique services

  Deliver Controller Reminder

notes  None

specification  Current Flight Plan

structure

  specialization of  Plan

domain-unique instance connections  None

domain-unique attributes  None

domain-unique services

  Reconformance Aid
  Limited Resolution Aid

notes  None

specification  Data Link Communications System

structure

  specialization of  Air/Ground Communications System

domain-unique instance connections  None
**domain-unique attributes**  None

**domain-unique services**  None

**notes**  None

__________________________

**specification**  Descent Profile

**structure**  None

**domain-unique instance connections**

Descent Profile (1,10) <-> Aircraft Class (1)

**domain-unique attributes**

Descent Gradient
Profile Index
Speed
Upper Altitude Bound

**domain-unique services**  None

**notes**  None

__________________________

**specification**  En Route Controlled Airspace (Borrowed from ATC model)

**structure**

*part of* Controlled Airspace

*contains parts*  Route; Airspace Volume

**domain-unique instance connections**  None

**domain-unique attributes**  None

**domain-unique services**  None

**notes**  None

__________________________

**specification**  En Route Controller
structure

specialization of Flight Manager

domain-unique instance connections None

domain-unique attributes

Sector Identification

domain-unique services

Request Automated Coordination
Request Automated Replan
Request Limited Resolution Aid
Request New Plan Creation
Request Plan Amendment
Request Reconformation Aid

notes None

 specification Fix/Waypoint ("Fix") (Borrowed from ATC model)

structure

part of Route

domain-unique instance connections

Fix/Waypoint ("Fix") (1) <-> Metering Restriction (0,1)

domain-unique attributes

Position (lat/long)
Minimum Altitude
Maximum Altitude
Current Altimeter Setting
Altitude Indicator (H/L)
Magnetic Declination
Flow Instruction

domain-unique services None

notes None
specification  Flight (Borrowed from ATC model)

structure

specialization of  User

contains parts  Aircraft; Flight Plan

domain-unique instance connections  None

domain-unique attributes

Aircraft Identification

domain-unique services  None

notes  None

specification  Flight Manager (Borrowed from ATC model)

structure

specialization of  Manager

generalization of  En Route Controller

domain-unique instance connections  None

domain-unique attributes  None

domain-unique services  None

notes  None

specification  Flight Plan (Borrowed from ATC model)

structure

part of  Flight; Plan

domain-unique instance connections
Flight Plan (1) <-> Flight Plan Amendment (1)

**domain-unique attributes**

- Aircraft Callsign
- Assigned Altitude at Initial Time
- Assigned Speed at Initial Time
- Departure Fix
- Destination Fix
- Filed Cruise Altitude
- Filed TAS at Cruise Altitude
- Fix at Initial Time
- Full Route
- Navigation/Communications Equipage
- Initial Time
- Modelled Route
- Speed at Initial Time

**domain-unique services** None

**notes** None

____________________________________________________

**specification** Flight Plan Amendment

**structure** None

**domain-unique instance connections**

Flight Plan Amendment (0,m) <-> Flight Plan (1)

**domain-unique attributes**

- Type
  - Flight Plan Change

**domain-unique services** None

**notes** None

____________________________________________________

**specification** Flow Instruction Non-Compliance

**structure**

  **specialization of** Problem
domain-unique instance connections  None

domain-unique attributes  None

domain-unique services  None

notes  None

specification  Heavy Weather Area

structure

   specialization of  Protected Airspace Volume ("Blocked Airspace")

domain-unique instance connections  None

domain-unique attributes

   Bearing Radial
   Description
   Severity
   Time of Latitude Update
   Velocity
   (Inherited attributes also)

domain-unique services  None

notes  None

specification  Highest-Ranked Resolution (HRR)

structure

   specialization of  Resolution

domain-unique instance connections  None

domain-unique attributes  None

domain-unique services  None

notes  None
**specification**  Holding Pattern Airspace ("Holding Pattern")

**structure**

**specialization of**  Protected Airspace Volume ("Blocked Airspace");  Airspace Volume

**domain-unique instance connections**  None

**domain-unique attributes**

- Distance to VOR
- Hold Axis Direction
- Holding Fix Name
- Leg Length
- Reference Route Name
- Turn Direction
- VOR Radial (Leg)
  (Inherited attributes also)

**domain-unique services**  None

**notes**  None

---

**specification**  Horizontal Wind Layer

**structure**

**part of**  Wind

**domain-unique instance connections**  None

**domain-unique attributes**

- Upper Altitude Bound
- x-component of Velocity
- y-component of Velocity

**domain-unique services**  None

**notes**  None

---

**specification**  Long Range Cruise Profile
structure None

domain-unique instance connections

Long Range Cruise Profile (1,7) <-> Aircraft Class (1)

domain-unique attributes

Long Range Cruise Speed
Upper Altitude Bound

domain-unique services None

notes None

___________________________________________________________________________
specification Machine Plan

structure

specialization of Non-Current Flight Plan

domain-unique instance connections

Machine Plan (1, 2) <-> Resolution (1)

domain-unique attributes

attribute Objection

domain-unique services None

notes None

___________________________________________________________________________
specification Manager (Borrowed from ATC model)

structure

specialization of User

generalization of Flight Manager

domain-unique instance connections None

domain-unique attributes None
**Domain-unique services**  None

**Notes**  None

---

**Specification**  Metering Restriction

**Structure**

*Specialization of*  Restriction

*Contains parts*  Metering Slot

**Domain-unique instance connections**

Metering Restriction (0,1) <--> Fix/Waypoint ("Fix") (1)

**Domain-unique attributes**

- Active Indicator
- Arrival Rate
- Highest Altitude at Meter Fix
- Location of Meter Fix
- Lowest Altitude at Meter Fix
- Maximum Speed at Meter Fix
- Metered Destination
- Name of Meter Fix
- Target Altitude

**Domain-unique services**  None

**Notes**  None

---

**Specification**  Metering Slot

**Structure**

*Part of*  Metering Restriction

**Domain-unique instance connections**  None

**Domain-unique attributes**

- Aircraft Identification

D-20
Slot Time

domain-unique services: None

notes: None

---

**specification** Multiple Problem

**structure**

contains parts: Problem

domain-unique instance connections: None

domain-unique attributes: None

domain-unique services: None

notes: None

---

**specification** Non-Conformance

**structure**

specialization of: Problem

domain-unique instance connections: None

domain-unique attributes: None

domain-unique services: None

notes: None

---

**specification** Non-Current Flight Plan

**structure**

specialization of: Plan

generalization of: Trial Plan; Pending Plan; Machine Plan

domain-unique instance connections: None

D-21
domain-unique attributes

Changes to Current Plan
Creation Time
Expiration Time
Ground Speed
Owner
Plan Identification
Status

domain-unique services None

notes None

specification Pending Plan

structure

specialization of Non-Current Flight Plan

domain-unique instance connections None

domain-unique attributes None

domain-unique services None

notes None

specification Plan

structure

contains parts Flight Plan; Clearance Directive; Trajectory

generalization of Non-Current Flight Plan; Current Flight Plan

domain-unique instance connections

Plan (1,2) <-> Problem (0, m)

domain-unique attributes None

domain-unique services None
specification  Planning Region State Segment

structure

   specialization of  State Segment

domain-unique instance connections  None

domain-unique attributes  None

domain-unique services  None

notes  None

specification  Problem

structure

   part of  Multiple Problem

   generalization of  Non-Conformance; Aircraft-Aircraft Conflict; Aircraft-Airspace Conflict; Flow Instruction Non-Compliance

domain-unique instance connections

   Problem (0,m) <-> Plan (1,2)
   Problem (1) <-> Resolution (0,10)

domain-unique attributes

   Subject Aircraft

domain-unique services  None

notes  None

specification  Protected Airspace Volume ("Blocked Airspace") (Borrowed from ATC model)

structure
specialization of Airspace Volume

generalization of Restricted Area; Heavy Weather Area; Holding Pattern Airspace ("Holding Pattern")

domain-unique instance connections None

domain-unique attributes

Adjacent Fixes
Airspace Centroid
BAS Extreme Points
Boundary Points
Convex Point Indicator
End Time
Exempt Aircraft
Maximum Altitude
Minimum Altitude
Rerouted Aircraft
Start Time
Type

domain-unique services None

notes None

___________________________________________________________________________

specification Radar Weather Area

structure

specialization of Wind/Weather Product

domain-unique instance connections None

domain-unique attributes

Name
Boundary Points
Severity

domain-unique services None

notes None

___________________________________________________________________________
**specification**  Resolution

**structure**

*specialization of* Highest-Ranked Resolution (HRR)

**domain-unique instance connections**

Resolution (0,10) <-> Problem (1)
Resolution (1) <-> Machine Plan (1,2)

**domain-unique attributes**

Aircraft Identification
Expiration Time
Maneuver Parameters
Maneuver Type

**domain-unique services**  None

**notes**  None

---

**specification**  Resource (Borrowed from ATC model)

**structure**

*generalization of* Airspace/Ground Resource; Aviation Weather Resource; Communications Resource

**domain-unique instance connections**  None

**domain-unique attributes**  None

**domain-unique services**  None

**notes**  None

---

**specification**  Restricted Area

**structure**

*specialization of* Protected Airspace Volume ("Blocked Airspace")

**domain-unique instance connections**  None
domain-unique attributes  (Inherited)

domain-unique services  None

notes  None

---

**specification**  Restriction

**structure**

*generalization of*  Metering Restriction

**domain-unique instance connections**

Restriction (0,m) <-> ACF Airspace ("Planning Region") (1)

**domain-unique attributes**

- Destination Fix
- Exception Fixes
- Message
- Name
- Restriction Subtype
- Restriction Type
- Status (active, inactive)

**domain-unique services**  None

**notes**  None

---

**specification**  Route (Borrowed from ATC model)

**structure**

*part of*  En Route Controlled Airspace

*contains parts*  Fix/Waypoint ("Fix"); Route Segment ("Airway Segment")

*generalization of*  Victor/Jet Route

**domain-unique instance connections**  None

**domain-unique attributes**  None

D-26
domain-unique services  None

notes  None

specification  Route Segment ("Airway Segment") (Borrowed from ATC model)

structure

part of  Route

domain-unique instance connections  None

domain-unique attributes

DME
Geometric Type
Major/Minor Status
Maximum En Route Altitude
Minimum En Route Altitude
Minimum Obstacle Clearance
Minimum Reception Altitude
VOR1
VOR2
VOR3

domain-unique services  None

notes  None

specification  Sector (Borrowed from ATC model)

structure

part of  ACF Airspace ("Planning Region")

specialization of  Airspace Volume

domain-unique instance connections  None

domain-unique attributes

Boundary Points
Center Designation
Center Identification
Focal Point Fixes
Maximum Altitude
Minimum Altitude
Radio Frequency
Shelves

domain-unique services  None

notes  None

specification  STAR

structure

specialization of  Terminal Route

domain-unique instance connections

STAR (0,4) <-> Airport (1)

domain-unique attributes

Name
Route

domain-unique services  None

notes  None

specification  State Segment

structure

part of  Trajectory

generalization of  Planning Region State Segment

domain-unique instance connections

State Segment (1) <-> State Segment End (1)
State Segment (1) <-> State Segment Start (1)

domain-unique attributes
Bearing
Acceleration
Gradient

domain-unique services None

notes None

specification State Segment End

structure None

domain-unique instance connections

State Segment End (1) <-> State Segment (1)

domain-unique attributes

Altitude
Ground Speed
Indicated Airspeed
Mach
Node/Fix
Segment Length
Time
True Airspeed
x-Coordinate
y-Coordinate

domain-unique services None

notes None

specification State Segment Start

structure None

domain-unique instance connections

State Segment Start (1) <-> State Segment (1)

domain-unique attributes
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Altitude</td>
<td></td>
</tr>
<tr>
<td>Distance</td>
<td></td>
</tr>
<tr>
<td>Ground Speed</td>
<td></td>
</tr>
<tr>
<td>Indicated Airspeed</td>
<td></td>
</tr>
<tr>
<td>Mach</td>
<td></td>
</tr>
<tr>
<td>Node/Fix</td>
<td></td>
</tr>
<tr>
<td>True Airspeed</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td></td>
</tr>
<tr>
<td>x-Coordinate</td>
<td></td>
</tr>
<tr>
<td>y-Coordinate</td>
<td></td>
</tr>
</tbody>
</table>

*domain-unique services* None

*notes* None

**specification** Terminal Airspace Volume (Borrowed from ATC model)

**structure**

*part of* Terminal Controlled Airspace

*generalization of* Approach Control Area; APDIA

*domain-unique instance connections* None

*domain-unique attributes* None

*domain-unique services* None

*notes* None

**specification** Terminal Controlled Airspace (Borrowed from ATC model)

**structure**

*part of* Controlled Airspace

*contains parts* Terminal Route; Terminal Airspace Volume

*domain-unique instance connections* None

*domain-unique attributes* None

*domain-unique services* None

*notes* None
specification  Terminal Route (Borrowed from ATC model)

structure

  part of  Terminal Controlled Airspace

  generalization of  STAR; Aircraft Descent Altitude Profile

domain-unique instance connections  None

domain-unique attributes  None

domain-unique services  None

notes  None

specification  Track

structure  None

domain-unique instance connections

  Track (0,m) <-> Aircraft (1)

domain-unique attributes

  History Trail
  Position
  Velocity Vector

domain-unique services

  Update Track Position
  Out-of-Conformance Detection

notes  None
specification Trajectory

structure

part of Plan

contains parts State Segment

domain-unique instance connections

Trajectory (1) <-> Controller Reminder (0,m)

domain-unique attributes None

domain-unique services

Handoff
Request Track Creation
Resynchronization
Trajectory Estimation

notes None

___________________________________________________________________________

specification Trial Plan

structure

specialization of Non-Current Flight Plan

generalization of Automated Replan Plan

domain-unique instance connections None

domain-unique attributes None

domain-unique services None

notes None

___________________________________________________________________________

specification User (Borrowed from ATC model)

structure

generalization of Manager; Flight
<table>
<thead>
<tr>
<th>Specification</th>
<th>Victor/Jet Route</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure</td>
<td>Specialization of Route</td>
</tr>
<tr>
<td>Domain-unique instance connections</td>
<td>None</td>
</tr>
<tr>
<td>Domain-unique attributes</td>
<td>None</td>
</tr>
<tr>
<td>Domain-unique services</td>
<td>None</td>
</tr>
<tr>
<td>Notes</td>
<td>None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Specification</th>
<th>Wind</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure</td>
<td>Contains parts Horizontal Wind Layer</td>
</tr>
<tr>
<td>Specialization of Wind/Weather Product</td>
<td></td>
</tr>
<tr>
<td>Domain-unique instance connections</td>
<td>None</td>
</tr>
<tr>
<td>Domain-unique attributes</td>
<td>None</td>
</tr>
<tr>
<td>Domain-unique services</td>
<td>None</td>
</tr>
<tr>
<td>Notes</td>
<td>None</td>
</tr>
</tbody>
</table>

| Specification | Wind/Weather Product (Borrowed from ATC model) |

D-33
structure

part of Aviation Weather System

generalization of Wind; Radar Weather Area

domain-unique instance connections None

domain-unique attributes None
domain-unique services  None

notes  None
LIST OF REFERENCES


Springen, A. L., November 1989, National Airspace System Airport Movement Area Control Operational Concept, WP-89W181 (Also DOT/FAA/DS-89/33), The MITRE Corporation, McLean, VA.

U.S. Department of Transportation, Federal Aviation Administration, Airman's Information Manual.

U.S. Department of Transportation, Federal Aviation Administration, Federal Aviation Regulations Parts 91 and 135.

BIBLIOGRAPHY


© 2000 The MITRE Corporation. ALL RIGHTS RESERVED.


U.S. Department of Transportation, Federal Aviation Administration, *Airman's Information Manual*.

U.S. Department of Transportation, Federal Aviation Administration, *Federal Aviation Regulations Parts 91 and 135*.


U.S. Department of Transportation, Federal Aviation Administration, *Order 7110.65: Air Traffic Control*.

U.S. Department of Transportation, Federal Aviation Administration, *Order 7110.83: Oceanic Air Traffic Control*.

<table>
<thead>
<tr>
<th>ACRONYMS</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-MITF</td>
<td>six-month illustration of technical feasibility</td>
</tr>
<tr>
<td>AAS</td>
<td>Advanced Automation System</td>
</tr>
<tr>
<td>ACARS</td>
<td>ARINC Communications Addressing and Reporting System</td>
</tr>
<tr>
<td>ACF</td>
<td>area control facility</td>
</tr>
<tr>
<td>AERA</td>
<td>Automated En Route Air Traffic Control</td>
</tr>
<tr>
<td>AGD</td>
<td>Adaptive Ground Delay</td>
</tr>
<tr>
<td>AIM</td>
<td>Airman's Information Manual</td>
</tr>
<tr>
<td>APDIA</td>
<td>Automated Problem Detection Inhibited Area</td>
</tr>
<tr>
<td>ARINC</td>
<td>Aeronautical Radio, Inc.</td>
</tr>
<tr>
<td>ARSA</td>
<td>airport radar services area</td>
</tr>
<tr>
<td>ARSR</td>
<td>air route surveillance radar</td>
</tr>
<tr>
<td>ARTCC</td>
<td>air route traffic control center</td>
</tr>
<tr>
<td>ARTS</td>
<td>Automated Radar Terminal System</td>
</tr>
<tr>
<td>ATA</td>
<td>airport traffic area</td>
</tr>
<tr>
<td>ATC</td>
<td>air traffic control</td>
</tr>
<tr>
<td>ATCRBS</td>
<td>Air Traffic Control Radar Beacon System</td>
</tr>
<tr>
<td>ATCT</td>
<td>air traffic control tower</td>
</tr>
<tr>
<td>BRITE</td>
<td>bright radar indicator tower equipment</td>
</tr>
<tr>
<td>CHI</td>
<td>computer-human interface</td>
</tr>
<tr>
<td>DME</td>
<td>distance measuring equipment</td>
</tr>
<tr>
<td>DUAT</td>
<td>Direct User Access Terminal</td>
</tr>
<tr>
<td>DYSIM</td>
<td>dynamic simulator</td>
</tr>
<tr>
<td>FAA</td>
<td>Federal Aviation Administration</td>
</tr>
<tr>
<td>FAR</td>
<td>Federal Aviation Regulations</td>
</tr>
<tr>
<td>FSS</td>
<td>flight service station</td>
</tr>
<tr>
<td>HRR</td>
<td>highest-ranked resolution</td>
</tr>
<tr>
<td>IAS</td>
<td>indicated airspeed</td>
</tr>
<tr>
<td>ICAO</td>
<td>International Civil Aviation Organization</td>
</tr>
<tr>
<td>IFR</td>
<td>instrument flight rule</td>
</tr>
<tr>
<td>I-Lab</td>
<td>Integration and Interaction Laboratory</td>
</tr>
<tr>
<td>ILS</td>
<td>instrument landing system</td>
</tr>
<tr>
<td>INS</td>
<td>inertial navigation system</td>
</tr>
<tr>
<td>JFK</td>
<td>John F. Kennedy</td>
</tr>
</tbody>
</table>
LORAN  long-range navigation
MLS    microwave landing system
NAS    National Airspace System
NAVAID navigational aid
NDB    nondirectional radio beacon
NSL    National Simulation Laboratory
OOA    object-oriented analysis
PAR    preferential arrival route
PDAR   preferential departure and arrival route
PDR    preferential departure route
PVD    Plan View Display
RDP    radar data processing
RNAV   area navigation
SID    standard instrument departure
STAR   standard terminal arrival route
TAAS   Terminal Advanced Automation System
TASF   Terminal Area Simulation Facility
TCA    terminal control area
TRACON terminal approach control
UTC    coordinated universal time
VFR    visual flight rule
VHF    very high frequency
VLF    very low frequency
VOR    VHF omnidirectional range