NASA Contractor Report 201681

ASAC Executive Assistant Architecture Description Summary

Eileen Roberts and James A. Villani
Logistics Management Institute, McLean, Virginia

Contract NAS2-14361

April 1997

National Aeronautics and
Space Administration
Langley Research Center
Hampton, Virginia 23681-0001

Contents

INTRODUCTION 2

NASA's Role in Promoting Aviation Technology 2

NASA's Research Objective 3

Genesis of the Aviation System Analysis Capability 4

Goals of the ASAC Project: Identifying and Evaluating Promising Technologies 5

Approach to Analyzing the Integrated Aviation System 6

Defining the Aviation System Analysis Capability 6

COMPONENTS OF THE AVIATION SYSTEM ANALYSIS CAPABILITY 8

Overview 8

ASAC Executive Assistant 9

ASAC Quick Response System (QRS) 10

ASAC Related Web Sites 12

Models 14

Data Repositories 14

Document Server 15

ASAC MODELS 15

Schematic of ASAC models and data flows 15

Analyses Using ASAC Models 18

ASAC Model Integration Prototype 19

ARCHITECTURE METHODOLOGY 19

The DSSA Approach 21

DSSA Domain Models (System Architecture) 22

OBJECT MODEL 22

DYNAMIC MODEL 22

FUNCTIONAL MODEL 23

DSSA Reference Architecture 23
ASAC EA DOMAIN MODEL 23
DSSA Stage 1—Define the Scope of the Domain 23
DSSA SUB-STAGE 1-1: ASAC DESCRIPTION AND GENERAL USER NEEDS 24
DSSA SUB-STAGE 1-2: DEFINE ASAC REQUIREMENTS 24
DSSA SUB-STAGE 1-3: DEFINE THE ASAC EA DOMAIN 29
DSSA SUB-STAGE 1-4: DEFINE ASAC EA-SPECIFIC RESOURCES 31
DSSA SUB-STAGE 1-5: RE-DEFINE THE ASAC EA DOMAIN (NARROW THE DOMAIN OF INTEREST) 32
DSSA SUB-STAGE 1-6: DETERMINE VERIFICATION PROCEDURE 36
DSSA SUB-STAGE 1-7: REVIEW AND ITERATE 36
DSSA Stage 2—Define and Scope Domain-Specific Elements 36
DSSA SUB-STAGE 2-1: DEFINE AND REFINE AN ELEMENT 37
DSSA SUB-STAGE 2-2: DEFINE AND REFINE APPLICATION SERVICES 41
DSSA SUB-STAGE 2-3: REFINE THE CONTEXT DIAGRAM 48
DSSA SUB-STAGE 2-4: IDENTIFY RELATIONSHIPS AMONG ELEMENTS 48
DSSA Sub-stage 2-5: ASAC EA Domain Dictionary 61
DSSA Sub-stage 2-6: ASAC EA High-Level Requirements Specification 61
FUNCTIONAL REQUIREMENTS 61
NON-FUNCTIONAL REQUIREMENTS 61
DSSA Sub-stage 2-7: Define Roles 64
DSSA Sub-stage 2-8: Define Assumptions 65
DSSA Sub-stage 2.9: Define Issues 67
DSSA Sub-stage 2-10: Review and Iterate 68
DSSA STAGE 3—DEFINE AND REFINE DOMAIN-SPECIFIC DESIGN AND IMPLEMENTATION CONSTRAINTS 68
DSSA Sub-stage 3-1: Define Constraints on the Architecture 68
IMPLEMENTATION CONSTRAINTS 69
DESIGN CONSTRAINTS 69

SOFTWARE CONSTRAINTS 69
HARDWARE AND PHYSICAL CONSTRAINTS 70
PERFORMANCE CONSTRAINTS 71
DSSA Sub-stage 3-2: Review and Iterate 71
DSSA STAGE 4—DEVELOP DOMAIN ARCHITECTURE(S) 71
DSSA Sub-stage 4-1: Object Model 72
DSSA Sub-stage 4-2: User Operational Concept 77
DSSA Sub-stage 4-3: Specific User Scenario 87
SCENARIO DESCRIPTION AND ASSUMPTIONS 87
DSSA Sub-stage 4-4: Dynamic Model 90
EVENT TRACE DIAGRAM 90
STATE DIAGRAM 96
DSSA Sub-stage 4-5: Functional Model 101
DATA FLOW DIAGRAM 101
DSSA Sub-stage 4-6: Requirements Traceability 103
DSSA Sub-stage 4-7: Review and Iterate 106
CONCLUSION 106

ASAC EXECUTIVE ASSISTANT ARCHITECTURE DESCRIPTION SUMMARY

In this technical document, we describe the system architecture developed for the Aviation System Analysis Capability (ASAC) Executive Assistant (EA). The document is composed of the following sections:

- Introduction
- Components of the Aviation System Analysis Capability
- ASAC Models
- Architecture Methodology
- ASAC EA Domain Model (Architecture)
- Conclusion.

In the first section, Introduction, we describe the genesis and role of the ASAC system. In the second and third sections we discuss the objectives of the ASAC system and provide an overview of components and models within the ASAC system.

In Architecture Methodology, we discuss our choice for an architecture methodology, the Domain Specific Software Architecture (DSSA), and the DSSA approach to developing a system architecture.

The next section, ASAC EA Domain Model (Architecture), includes the development process and the ASAC EA system architecture description. This section contains four subsections, one for each DSSA stage:

- DSSA Stage 1—Define the Scope of the Domain
- DSSA Stage 2—Define and Scope Domain-Specific Elements
- DSSA Stage 3—Define and Refine Domain-Specific Design and Implementation Constraints

- DSSA Stage 4—Develop Domain Architecture(s).

The final section is the document conclusion.

This document has six appendices. They are

- Bibliography
- Appendix A—Acronyms
- Appendix B—Domain Dictionary
- Appendix C—Proceedings of the ASAC Architecture Meetings, 11–20 Mar 1996
- Appendix D—Client/Server architecture
- Appendix E—Preliminary ASAC EA Design.

The first four appendices are self-explanatory. Appendix D is a short tutorial on client/server systems. The final Appendix is a high-level view of a preliminary design for ASAC.

INTRODUCTION

NASA's Role in Promoting Aviation Technology

The United States has long been the world's leader in aviation technology for both civil and military aircraft. During the past several decades, U.S. firms have transformed this position of technological leadership into a thriving industry with large domestic and international sales of aircraft and related products. In 1992, sales of civil aircraft peaked at $39.9 billion, with exports of $24.3 billion. Exports of engines, parts, and related products totaled $12.4 billion in the same year. The comparable figures for 1995 were $23.6 billion, $12.8 billion, and $11.9 billion, respectively.

Despite the industry's historic record of success, the difficult business environment of the past several years has stimulated concerns about whether the U.S. aeronautics industry will maintain its worldwide leadership position. Increased competition, both technological and financial, from European and other non-U.S. aircraft manufacturers has reduced the global market share of U.S. producers of large civil transport aircraft and cut the number of U.S. airframe manufacturers to

only two. Order cancellations and stretch-outs of deliveries by airlines, forthcoming noise abatement requirements, and environmental concerns create additional challenges for U.S. producers and purchasers of aircraft.

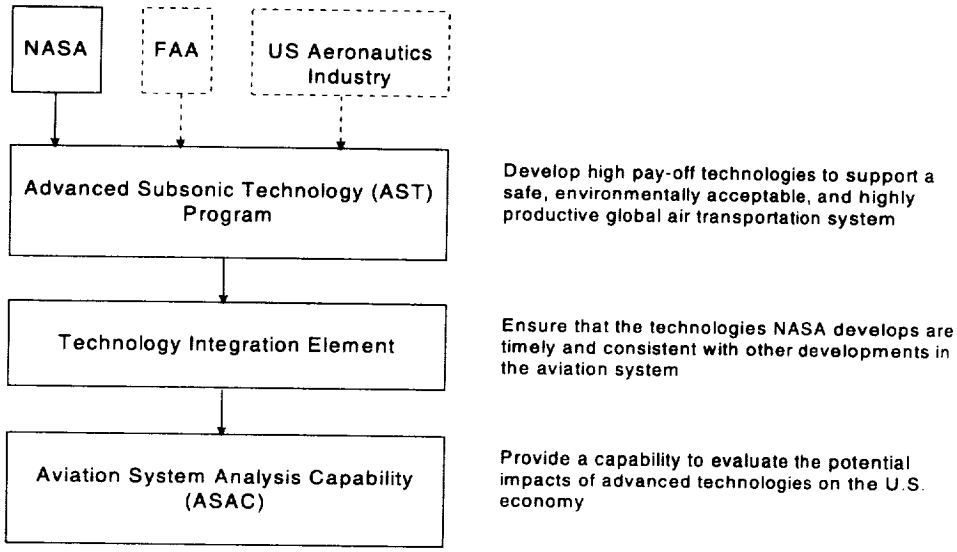
The primary role of the National Aeronautics and Space Administration (NASA) in supporting civil aviation is to develop technologies that improve the overall performance of the integrated air transportation system, making air travel safer and more efficient, while contributing to the economic welfare of the United States. NASA conducts much of the basic and early applied research that creates the advanced technology introduced into the air transportation system. Through its technology research program, NASA aims to maintain and improve the leadership role in aviation technology and air transportation held by the United States for the last half century.

The principal NASA program supporting subsonic transportation is the Advance Subsonic Technology (AST) program, managed by the Subsonic Transportation Division, Office of Aeronautics, NASA Headquarters. In cooperation with the Federal Aviation Administration and the U.S. aeronautics industry, the AST program develops high-payoff technologies that support the development of a safe, environmentally acceptable, and highly productive global air transportation system. NASA measures the long-term success of its AST program by how well it contributes to an increased market share for U.S. civil aircraft and aircraft component producers and the increased effectiveness and capacity of the national air transportation system.

NASA's Research Objective

To meet its objective of assisting the U.S. aviation industry with the technological challenges of the future, NASA must identify research areas that have the greatest potential for improving the operation of the air transport system. Therefore, NASA seeks to develop the ability to evaluate the potential impact of various advanced technologies. By thoroughly understanding the economic impact of advanced aviation technologies and by evaluating the use of new technologies in the integrated aviation system, NASA aims to balance its aeronautical research program and help speed the introduction of high-leverage technologies. Figure 1 illustrates NASA's research objective.

Figure 1. NASA's Research Objective

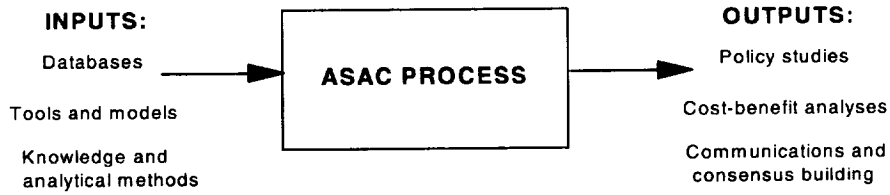


Genesis of the Aviation System Analysis Capability

Technology integration is the element of the AST program designed to ensure that the technologies NASA develops are timely and consistent with other developments in the aviation system. Developing an ASAC is one of the objectives of the technology integration element. With this analytical capability, NASA and other organizations in the aviation community can better evaluate the potential economic impacts of advanced technologies.

ASAC is envisioned primarily as a process for understanding and evaluating the impact of advanced aviation technologies on the U.S. economy. ASAC consists of a diverse collection of models and databases and analysts, and individuals from the public and private sectors brought together to work on issues of common interest to organizations within the aviation community. ASAC will also be a resource available to those same organizations to perform analyses; provide information; and assist scientists, engineers, analysts, and program managers in their daily work. ASAC will provide this assistance through information system resources, models, and analytical expertise and conducting and organizing large-scale studies of the aviation system and advanced technologies. Figure 2 displays this concept.

Figure 2. ASAC Process



Goals of the ASAC Project: Identifying and Evaluating Promising Technologies

Develop credible evaluations of the economic and technological impact of advanced aviation technologies on the integrated aviation system is the principal objective of ASAC. These evaluations will then be used to help NASA program managers select the most beneficial mix of technologies for NASA investment, both in broad areas, such as propulsion or navigation systems, and in more specific projects within the broader categories. Generally, engineering analyses of this kind require multidisciplinary expertise, use several models of different components and technologies, and consider multiple economic outcomes and technological alternatives. These types of analyses are most effective if they include information and inputs from organizations and analysts from different parts of the aviation community. In this way, the studies incorporate the expertise of people around the United States and build acceptance from the start of the research effort.

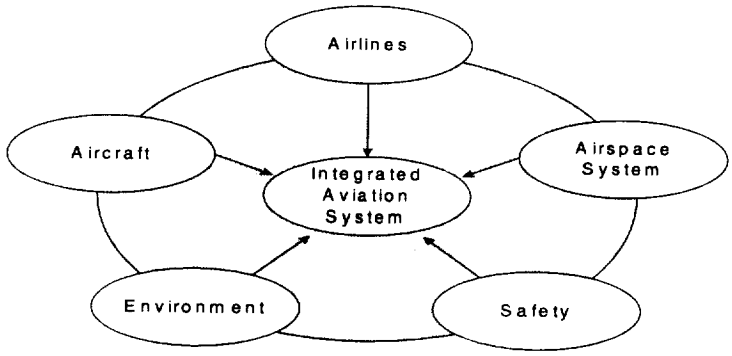
In addition to identifying broad directions for investments in technology, the program must also help researchers at NASA and elsewhere evaluate the economic potential of alternative technologies and systems. By better informing engineers about potential markets for technologies and data on how the current system works, ASAC will help NASA engineers incorporate their customers' needs more easily into their routine work. These types of problems most likely involve investigating technical designs for specific aircraft or subsystems that can readily replace existing equipment without requiring significant changes to other aviation components. With such information, researchers could more easily evaluate the utility of alternative designs and quickly estimate the value of their design concepts. Analysts from industry, government, and universities would also use ASAC in this way.

Approach to Analyzing the Integrated Aviation System

The most useful aviation technologies are not necessarily the most technically advanced. Rather, NASA and industry must invest in the technologies that have the most promising payoffs—those that clearly demonstrate a capacity for economically viable performance enhancements—from the perspective of those organizations that will purchase and operate the technologies.

Because new aviation technologies are introduced into a complex system, the potential impact of any proposed technology must be analyzed from a system-wide perspective. Otherwise, the potential impact may be over- or underestimated because of the unexamined interdependencies with other elements of the aviation system. Figure 3 shows the components of the integrated aviation system.

Figure 3. Components of the Integrated Aviation System



In summary, with the ASAC, users can develop credible evaluations of the economic and technological impact of advanced aviation technologies on all components of the integrated aviation system.

Defining the Aviation System Analysis Capability

Defining the purpose, goals, requirements, and components of an item in clear and easily understood terms is the first step in developing any item. To achieve this step, a series of ASAC architecture meetings were held from 11 to 20 March 1996

to set the groundwork for ASAC architecture development. Topics covered during the meetings included the following:

- What is ASAC?
- ASAC Users
- ASAC Goals and Attributes
- ASAC Terms
- ASAC Requirements
- ASAC Models and Data
- Information flow between ASAC models
- First Generation ASAC.

The products created at the architecture meetings are documented in *Proceedings of the ASAC Architecture Meetings, 11–20 Mar 1996*, included in Appendix C.

Based on the March ASAC architecture meetings, we defined the objectives of ASAC—providing a system that will

1. Perform analyses of fundamental importance to NASA, other government agencies, and industry, at both a macro and detailed level. This objective requires impeccable objectivity, credibility, and the complete absence of conflicts of interest.

2. Provide credible evaluations of the economic and technological impacts of advanced aviation technologies on the integrated aviation system.

3. Provide access to a collection of data and analytical tools with which researchers at NASA and elsewhere can quickly evaluate the economic potential of alternative technologies and systems.

4. Serve as a commonly accepted, credible vehicle for interaction and cooperation both within NASA and among NASA, other government agencies, and industry.

5. Model the decision-making of the air carriers who actually buy and operate aircraft and systems.

6. Allow for a hierarchy of models at varying levels of detail that are appropriate for the analytical task at hand. However, when such detailed information is not needed, reduce the computational burden by selecting only those critical items of data needed to pass the analysis on to the next step.

7. Allow independent and incremental development of the ASAC capability.

8. Maximize the enhanced analytical capability that can be achieved in the first two years of development.

9. Using available models as appropriate, incorporate modular operation of individual models whenever feasible.

10. Minimize NASA's risk during the development period by limiting the amount of model integration required, and by designing model improvements and development to reduce the number of tasks on critical paths.

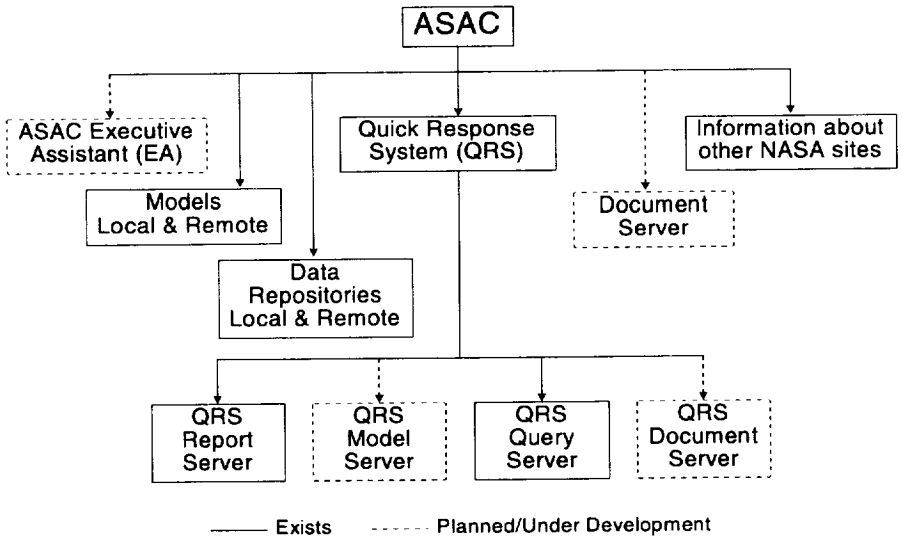
11. Protect proprietary data, commercial data, and intellectual property.

COMPONENTS OF THE AVIATION SYSTEM ANALYSIS CAPABILITY

Overview

ASAC is a diverse collection of models and databases and analysts and individuals from the public and private sectors brought together to work on the issues of common interest to organizations within the aviation community. Figure 4 shows the major system components of ASAC.

Figure 4. ASAC System Components



Some ASAC system components exist; others are under development. Two ASAC components, Information about other NASA sites and the Document Server will be available to the general public. All other ASAC components will be available on a restricted basis. The following sections provide a brief description of each ASAC system component as it exists today.

ASAC Executive Assistant

With the ASAC EA, researchers at NASA and elsewhere can quickly evaluate the economic potential of alternative technologies and systems. By providing inputs to and linking the many models and data that the ASAC system will comprise, the EA will provide an intelligent interface with which the user can perform detailed analyses. Definition of the ASAC Executive Assistant architecture is the focus of this document.

Table 1 outlines the tentative development schedule for the EA.

Table 1. Proposed Development Schedule for the ASAC Executive Assistant

Item	Year
Define ASAC EA requirements	1995
Define the ASAC EA	1996
Develop the ASAC EA Architecture	1996
Develop the Model Integration Prototype (First Generation ASAC)	1996–1997
Design the ASAC EA	1997
Develop an ASAC EA Proof-of-Concept	1997–1998
Develop and deploy the ASAC EA	1998–1999

The ASAC Model Integration Prototype (First Generation ASAC) will demonstrate integration of some of the First Generation ASAC models. Development of this prototype is the first step in providing a robust, fully functional, ASAC Executive Assistant.

The ASAC Model Integration Prototype (First Generation ASAC) will comprise a subset of the complete ASAC model network. NASA and others will use it to perform selected economic analysis of aircraft technology and air traffic management improvements.

The ASAC Model Integration Prototype (First Generation ASAC) will be available to authorized ASAC users (password protected). Users will use a World Wide Web (WWW) browser to access the system.

ASAC Quick Response System (QRS)

Users can use a forms-capable WWW browser such as Netscape or Mosaic to access the QRS. The QRS Report Server is operational and located at http://www.asac.lmi.org/access/rserver.html. The complete ASAC QRS will comprise four components:

- QRS Report Server
- QRS Model Server
- QRS Query Server
- QRS Document Server.

With the QRS Report Server, users can generate reports from information stored in the ASAC data repository. Ninety reports are available from the following categories:

- Airport data
- Carrier data
- Equipment data
- Flight segment data
- Jet engine data
- Origin and destination data
- Miscellaneous (includes airport and carrier codes).

At present, six models are available from the QRS Model Server. The models are listed in Table 2.

With the QRS Query Server, a user can query the following information that is stored in the ASAC data repository:

- Airport code
- Airport location
- Airport name
- Airport rundown
- Bearing between airports
- Carrier code
- Carrier name
- Distance between airports
- Equipment code
- Equipment name.

The fourth component, the QRS Document Server, will host QRS related documents. For example, *The ASAC QRS Report Server User's Guide*, LMI Report NS601RD1, is available for download.

ASAC Related Web Sites

Links to the following sites are currently available from the ASAC Related Web Sites page, located at http://www.asac.lmi.org/related.html, on the World Wide Web:

- NASA Sites
  - NASA Home Page
  - NASA Affiliated Institutes
  - NASA Ames Research Center
  - NASA Ames—DARWIN Distributed Remote Aeronautics Management
  - NASA Ames Systems Analysis Branch
  - NASA Dryden Flight Research Center
  - NASA Goddard Space Flight Center
  - NASA Johnson Space Center Space and Life Sciences Directorate
  - NASA Kennedy Space Center
  - NASA Langley Research Center
  - NASA Lewis Research Center
  - NASA Lyndon B. Johnson Space Center
  - NASA Marshall Space Flight Center
  - NASA Technical Report Server

- Aviation-Related Sites
  - Aerospace Industries Association (AIA)
  - AirNav: Airport Information
  - Air Transportation Association (ATA)
  - Center for Advanced Aviation System Development (CAASD)
  - Commercial Aviation Resource Center
  - Federal Aviation Administration (FAA)
  - FAA List of Other Aviation Internet Sites
  - FAA Office of Environment and Energy
  - General Aviation Related Server List
  - MIT Flight Transportation Laboratory
  - MIT Modeling Research Under NASA/AATT
  - NASAO Center for Aviation Research & Education 5010 Database
  - Regional Airline Association
- Other Related Sites
  - United States International Trade Commission

Additional links of interest to the ASAC community are periodically added to this page.

Models

Models support the QRS and its components as well as the EA. New models will be added to the repositories as they are developed. The following models currently reside on the ASAC system:

Table 2. Content of ASAC Model Repositories

Model	Operating System	Comment
ASAC Air Carrier Investment Model—Second Generation	Windows (Excel, Version 5.0) Macintosh (Excel,Version 5.0)	Available as a standalone model from the QRS Model Server
ASAC Air Carrier Network Cost Model	HP-UX 10.20	Available via a WWW interface
ASAC Airport Capacity Model	HP-UX 10.20	Available via a WWW interface
ASAC Airport Delay Model	HP-UX 10.20	Available via a WWW interface
ASAC Flight Segment Cost Model (Cost Translator)	HP-UX 10.20	Available via a WWW interface
ASAC Flight Segment Cost Model (Mission Generator)	HP-UX 10.20	Available via a WWW interface

Data Repositories

Like models, data repositories support the QRS and its components as well as the EA. New data are added to the repositories on an as-needed basis. The following data currently reside in the data repositories:

Table 3. Content of ASAC Data Repositories

Item	Years
DOT Airline Service Quality Performance (ASQP)	1993 and 1995
DOT Form 41 Financial	1989–1994
DOT Origin and Destination Matrices	1989–1994
DOT Schedule B-43 Airframe Inventory	1994
DOT T-100 Flight Segment	1989–1994
DOT T-3/T-100 Airport Rank	1989–1994
FAA Noise Certification	1996
FAA Terminal Area Forecast (TAF)	1976–1994 Historical 1995–2010 Forecast
Official Airlines Guide (OAG) North American and Worldwide Merge Files	1993
World Jet Inventory	1993 and 1995

Document Server

ASAC-related documents (other than those for the QRS) will be available for download from the Document Server.

ASAC MODELS

Schematic of ASAC models and data flows

Defining the ASAC model data flows was one of the major accomplishments of the March ASAC architecture meetings. These flows show the framework of the models that will be a part of the ASAC system, and using these flows, planners can develop models in a logical sequence. The flows also validate the planned usage of the ASAC system.

The models are grouped into the following four analytical areas:

- 1.0 Aircraft and System Technologies
- 2.0 FAA Air Traffic Management
- 3.0 Environment
- 4.0 General Aviation.

Each model has a unique number. The number designates the model's analytical area, e.g., all model numbers that begin with a *2* belong to the FAA Air Traffic Management analytical area. The number also designates a model's position in a logical stream. For example, a stream might comprise the following models:

2.3 ASAC Airport Capacity Model →,

2.3.2 ASAC Airport Delay Model →,

2.3.2.1 ASAC Flight Segment Cost Model—Cost Translator.

The model data flows for each of the four analytical areas are shown in Figures 5 through 8. Squares represent models that belong to the analytical area named in the figure title; circles represent models that belong to a different analytical area.

Figure 5. 1.0 Aircraft and System Technologies

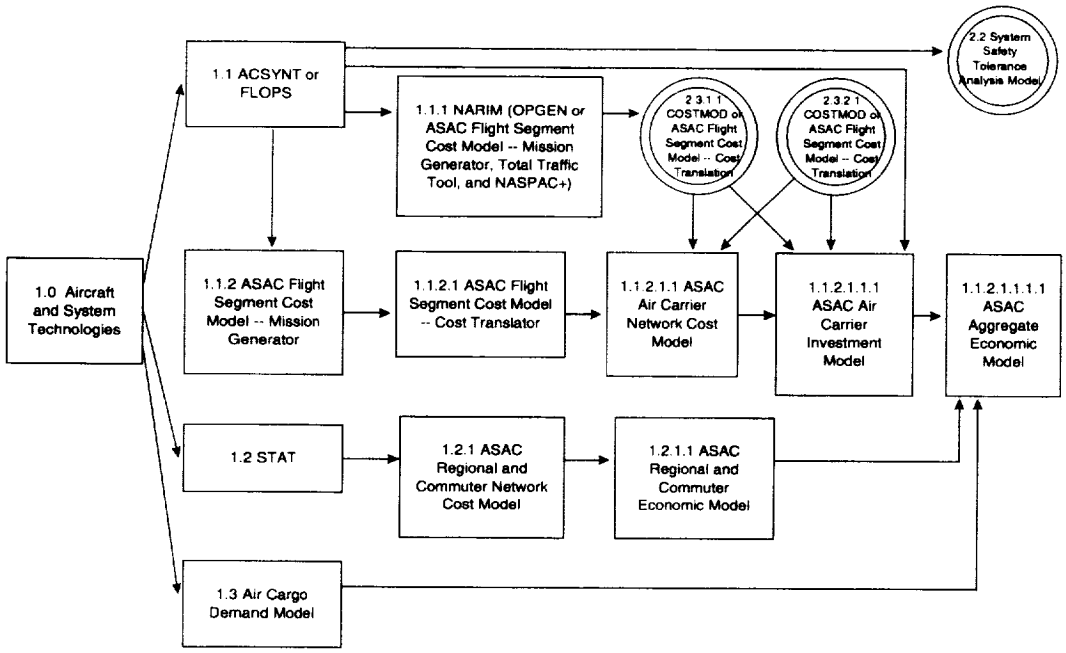


Figure 6. 2.0 FAA Air Traffic Management

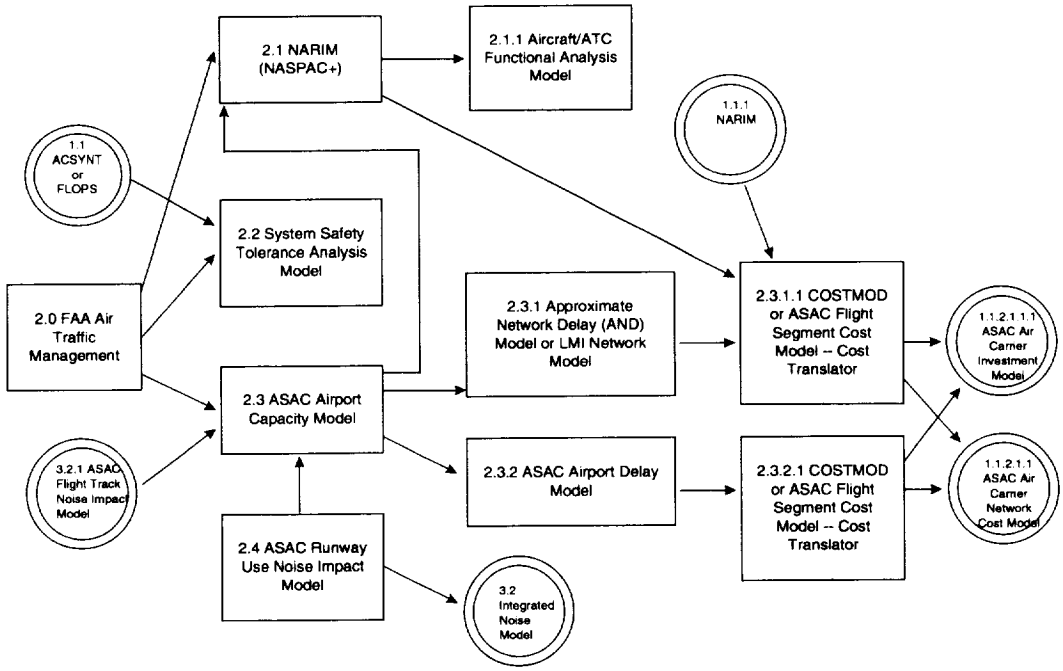


Figure 7. 3.0 Environment

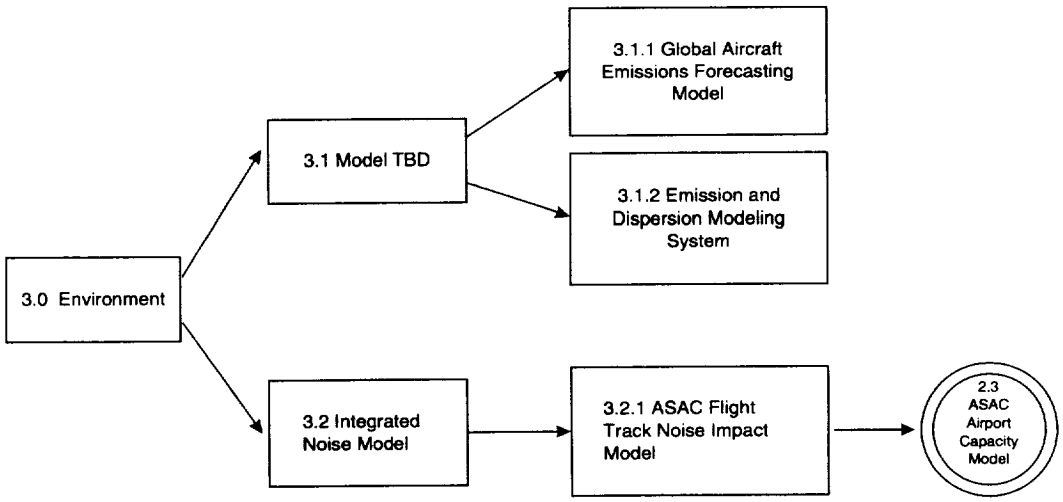
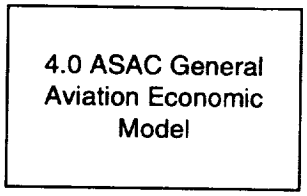


Figure 8. 4.0 General Aviation



Therefore, ASAC is a clearly defined domain that consists of numerous analytical models. Additional information on the inputs and outputs of each model can be found in *Proceedings of the ASAC Architecture Meetings, 11 - 20 Mar 1996*, included in Appendix C.

Analyses Using ASAC Models

The above represented models can be used either alone or in combination to analyze specific AST program elements. Table 4 shows representative collections of models relevant to these areas.

Table 4. ASAC Models Used to Analyze AST Program Elements

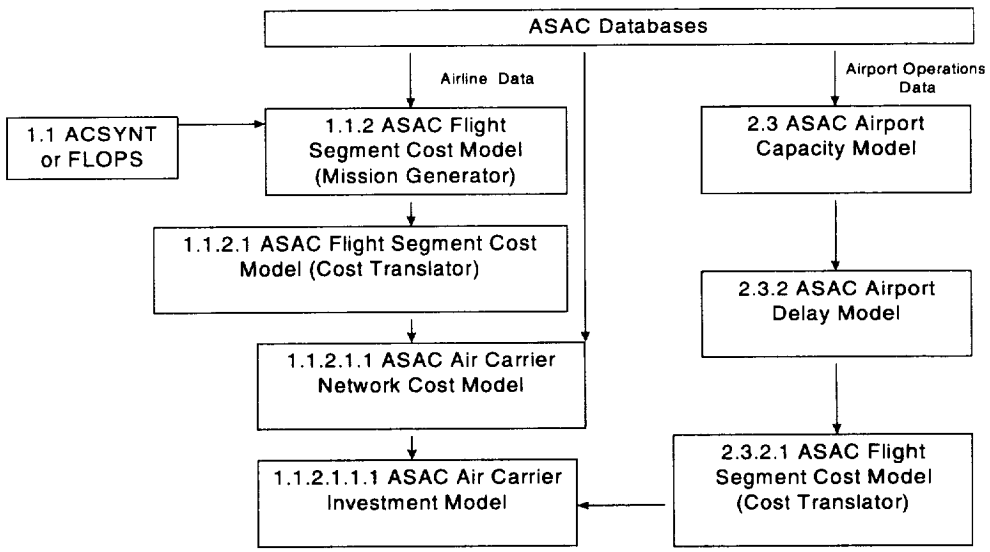
AST Program Element	ASAC Models
Advanced Air Transportation Technology	ASAC Flight Segment Cost Model, ASAC Airport Capacity Model, ASAC Air Carrier Investment Model, ASAC System Safety Tolerance Analysis Model, National Airspace Research and Investment Model
Aging Aircraft	ASAC Air Carrier Investment Model, ASAC Database
Civil Tiltrotor	National Airspace Research and Investment Model, ASAC Database
Composites	Flight Optimization System Model, Aircraft Synthesis Model, ASAC Air Carrier Investment Model
Environmental Assessment	ASAC Flight Segment Cost Model
Fly-by-Light and Power-by-Wire	Flight Optimization System Model, Aircraft Synthesis Model, ASAC Air Carrier Investment Model
General Aviation and Commuter Aviation	National Airspace Research and Investment Model, ASAC Database
Integrated Wing	Flight Optimization System Model, Aircraft Synthesis Model, ASAC Air Carrier Investment Model
Noise Reduction	ASAC Air Carrier Flight Track Efficiency Model, ASAC Airport Capacity Model, ASAC Air Carrier Investment Model
Propulsion	ASAC Flight Segment Cost Model, ASAC Air Carrier Investment Model, Flight Optimization System Model, Aircraft Synthesis Model
Terminal Area Productivity	ASAC Airport Capacity Model, ASAC Airport Delay Model, ASAC Air Carrier Investment Model, ASAC System Safety Tolerance Analysis Model, National Airspace Research and Investment Model

ASAC Model Integration Prototype

A subset of ASAC models will initially be implemented in the ASAC Model Integration Prototype (First Generation ASAC). As additional models are developed, they will be added to the ASAC system.

As shown in Figure 9, this collection of models will enable analyses of improvements in aircraft technology (the left-most chain in the figure) or improvements in air traffic management (the right-most chain in the figure).

Figure 9. Models Used in the ASAC Model Integration Prototype



ARCHITECTURE METHODOLOGY

In this section, we discuss our choice for an architecture methodology and the corresponding approach to developing a system architecture. The Defense Advanced Research Projects Agency (DARPA) is a research organization for the Department of Defense. We selected a component of DARPA's Evolutionary Software Development program, the Domain-Specific Software Architecture (DSSA), as the architecture methodology for the ASAC system. As an approach, DSSA is ideal for a system with a clearly identified domain such as the ASAC.

DSSA was a joint industry and university research effort designed to apply leading-edge basic research to real-world problems. Contributors include Lockheed-

Martin and Honeywell Corporations. DSSA is currently being used on such programs as the U.S. Army's Joint Simulation System (JSIMS).

DSSA is based on the Object Modeling Technique (OMT) methodology, which was developed by Jim Rumbaugh, et al. OMT is one of the most widely adopted object-oriented analysis and design methodologies. DSSA encompasses the development of

- a domain model (system architecture);
- a reference architecture (system design);
- its supporting infrastructure and environment, and
- a process and methodology to instantiate and refine and evaluate it.

Key ideas in DSSA are

- an iterative, architecture driven development process and
- component-based application development.

Goals of the DSSA are to

- accelerate development of technology for domain analysis techniques and architecture description methods and
- increase productivity and quality of applications developed in the domains of interest.

The benefits of using the DSSA include

- development of a scalable, adaptable, and evolvable system architecture;
- early validation;
- iterative refinement of requirements and design;
- facilitation of rapid prototyping;
- significant reduction in software development cost and schedule;
- reduced risk and improved system quality and reliability as systems are constructed from proven architectures and proven parts; and

- significant reduction in software maintenance costs because of lower error rate, improved component quality, and single-point maintenance of common parts.

The DSSA Approach

A domain engineering process is used to generate a DSSA. The goal of the process is to map user needs into system and software requirements that, based on a set of implementation constraints, eventually define a DSSA.

There are five stages in the domain engineering process. Each stage is further divided into steps or sub-stages. The process is concurrent, recursive, and iterative. Therefore, completion requires several passes through each stage. The five stages in the domain engineering process are described in Table 5.

Table 5. DSSA Stages

Stage	Title	Description	ASAC EA Phase
1	Define the scope of the domain	Definition of what can be accomplished with emphasis on user needs	Architecture
2	Define/refine domain-specific elements	Similar to requirements analysis with emphasis on the problem space	Architecture
3	Define/refine domain-specific design and implementation constraints	Similar to requirements analysis with emphasis on the solution space	Architecture
4	Develop domain models and architectures	Similar to high-level design with emphasis on defining module and model interfaces and semantics	Architecture and design
5	Produce and gather reusable work products	Implementation and collection of reusable artifacts such as code and documentation	Design and development

This architecture effort described in this report covers applicable parts of stages 1, 2, 3, and 4 (partial). Stages 4 (remainder) and 5 address system design and development. The ASAC design and development tasks will be a follow-on effort to this architecture definition effort.

DSSA Domain Models (System Architecture)

The DSSA domain model consists of three groups of models—object, dynamic, and functional (Table 6). The domain model shows the functions performed and the data, control information, and entities flow among those functions. It comprises several representations of the behavior of the entities in the domain.

A DSSA domain model does not assign the functions to specific components, nor does it specify the interfaces among functions; it documents the existence of the interfaces and the interactions that are involved. A domain model is, therefore, a logical model; it is not a physical model. A domain model represents the *what* of a system. It places the capabilities that the system architecture should support in context.

Table 6. DSSA Model Definitions

Model	Description	Diagram
Object	A description of the objects in a system and the inter-relationships among objects	Object
Dynamic	A description of the control of a system, particularly emphasizing the time-dependent processing and temporal ordering of functions.	Event Trace State
Functional	A depiction of the relationships among functions, usually within the problem domain.	Data Flow

OBJECT MODEL

The object model describes the static structure of the objects in a system and their relationships. The object model contains object diagrams. An object diagram is a graph in which nodes are object classes and arcs are relationships among classes.[1]

DYNAMIC MODEL

The dynamic model describes the aspects of a system that change over time. The dynamic model specifies and implements the control aspects of a system. The dynamic model contains event trace diagrams and state diagrams. An event trace diagram depicts a sequence of events and the objects exchanging the events. A

[1] Rumbaugh, J. et.al., Object-Oriented Modeling and Design.

state diagram is a graph in which nodes are states and arcs are transitions among states caused by events.[2]

FUNCTIONAL MODEL

The functional model describes the data value transformations within a system. The functional model contains data flow diagrams. A data flow diagram is a graph in which nodes are processes and arcs are data flows.[3]

DSSA Reference Architecture

A reference architecture is an extension of a domain model. It describes how the functions and interfaces occur or could occur. Functions are allocated to the components that implement them, and interface requirements are specified. Ideally, functions in the domain model are explicitly mapped or linked to components of the reference architecture. Reference architectures represent how a domain model will be implemented. The ASAC reference architecture will be generated during fiscal year 1997.

ASAC EA DOMAIN MODEL

We tailored the DSSA approach[4] to meet the needs of the ASAC architecture development effort. In this section, we discuss each of the applicable areas of the first four DSSA stages.

DSSA Stage 1—Define the Scope of the Domain

The first phase in the domain-engineering process focuses on the contents of the domain of interest (bound the problem). In this stage, we define what can be accomplished with an emphasis on the user's needs.

The sub-stages of DSSA Stage 1 are

- 1-1 Description and general user needs,
- 1-2 Requirements,

[2] Rumbaugh, J. et. al., Object-Oriented Modeling and Design.

[3] Rumbaugh, J. et. al., Object-Oriented Modeling and Design.

[4] Tracz, W, and L. Coglianese. *Domain-Specific Software Architecture Engineering Process Guidelines*, ADAGE-IBM-92-028.

- 1-3 Domain definition,
- 1-4 Resource list,
- 1-5 Re-defined domain definition,
- 1-6 Verification procedure, and
- 1-7 Review and iterate.

DSSA SUB-STAGE 1-1: ASAC DESCRIPTION AND GENERAL USER NEEDS

ASAC is envisioned primarily as a process for understanding and evaluating the impact of advanced aviation technologies on the U.S. economy. ASAC consists of a diverse collection of models and databases and analysts, and individuals from the public and private sectors brought together to work on the issues of common interest to organizations within the aviation community. ASAC also will be a resource available to those same organizations to perform analyses; provide information; and assist scientists, engineers, analysts, and program managers in their daily work. ASAC will provide this assistance through information system resources, models, and analytical expertise and by conducting and organizing large-scale studies of the aviation system and advanced technologies.

DSSA SUB-STAGE 1-2: DEFINE ASAC REQUIREMENTS

ASAC requirements, as identified by potential ASAC users at the March ASAC architecture meetings, are grouped into the following six areas:

1. General
2. General and EA
3. EA
4. Model
5. Database
6. QRS

The following are the requirements for this sub-stage:

- General requirements

  1. ASAC will be able to analyze aircraft, airspace, air carrier operations and investments, environmental issues, and safety.

  2. ASAC will be built to provide for model access; ASAC databases and tools must also be developed.

  3. ACSYNT and FLOPS will incorporated into ASAC.

  4. ASAC will contain tools developed to analyze the potential impact of airspace technologies and new aircraft.

  5. ASAC will quantify the impact of various technologies on the U.S. economy, employment, and trade.

  6. ASAC will provide design methods used to analyze the impact of aircraft operations on the atmosphere in both the terminal area and en-route air spaces.

  7. Analytical studies will be performed in parallel with ASAC development.

  8. ASAC analyses methods will be openly represented (documented and described) to the extent allowed (protect proprietary data and intellectual property rights). (ASAC documentation will provide descriptions of the model input and output files.)

  9. ASAC will provide on-line responses and shall provide immediate feedback on potential economic benefits when analytically appropriate.

  10. Selected data repositories will be periodically updated.

  11. ASAC may provide on-line access to NASA and FAA sites.

  12. ASAC may provide on-line access to the NASA Technical Report server.

  13. During development, ASAC may restrict Internet access to specific remote machines.

14. ASAC will provide system and data security in accordance with best commercial practices.

15. ASAC will provide the capability to exercise individual models.

16. ASAC will use existing models to the maximum extent possible (reusability).

17. ASAC will provide a link to all ASAC model log-in screens. A user must have access to the model and must log in as a user.

- General and EA requirements

18. Implementation of ASAC will remain flexible as analyses are performed in response to the changing need of its users (users can substitute models and data).

19. Implementation of ASAC will allow the addition of new models and tools to the system.

20. ASAC will use industry standards such as HTML 2.0.

21. ASAC development will be under configuration control.

22. ASAC will support users using Windows, Macintosh, HP-UX, SunOS, and SGI IRIX.

23. ASAC results will be consistent and reliable.

- EA requirements

24. With EA, users will have guided access (with uniform look and feel) to information now widely scattered in widely varying formats.

25. EA will accommodate operation of models and databases at remote sites.

26. With EA, the user will be able to evaluate a class of predefined representative problems.

27. With EA, the user will be able to view and modify data at intermediate steps (visually inspect and change data being transferred among models).

28. EA will provide an audit trail showing models used and input and output files.

29. EA will run analyses comprising one or more models.

30. EA model inputs will be predefined.

31. Users will be able to change EA model input values.

32. EA models will allow multiple inputs.

33. The EA will determine the number of times the model needs to run.

34. Invalid EA model inputs for a specific analysis will not be available.

35. EA model inputs may be a file.

36. Users will be able to view EA model outputs in raw and converted format.

37. EA analysis will stop at selected intermediate steps.

38. With the EA, users will be able to select the model inputs and outputs to view.

39. With the EA, users will be able to save viewed data.

40. The EA will mail a notification of completion to the user.

41. In an information message, the EA will indicate an estimate of the run time for selected models.

42. With the EA, users will be able to cancel analysis at all intermediate steps.

43. The EA will provide a general optimizing tool.

44. For each analysis, the EA will create an analysis document that includes elements needed to recreate the analysis.

- Model requirements

  45. ASAC will include a detailed cost-benefit model used to analyze the viability of advanced aircraft technologies.

  46. ASAC will include a model used to generate forecasts of air travel demand, airline costs, and required aircraft inventories.

  47. ASAC will include a model that estimates the cost to operate a representative (or actual) airline route structure with different aircraft.

  48. ASAC will estimate the noise characteristics of new aircraft and evaluate the noise impacts of aircraft operations on airport areas.

  49. ASAC will include a model to estimate system safety tolerance.

  50. Models may use multiple data repositories.

  51. External data will be available to models as input.

  52. ASAC will establish standards for use by model developers.

  53. With ASAC, legacy model developers will provide advance notice of and detailed information about changes and allow LMI to be a tester for changes.

- Database requirements

  54. Databases describing air traffic and airport operations and capacity will be constructed for ASAC.

  55. ASAC will provide access to searchable databases of aviation incidents and accidents.

  56. ASAC will provide raw and processed data on air carrier operations, aircraft utilization, and aircraft operating costs.

- QRS requirement

  57. A common query tool will be developed so that the interface to the databases appears seamless to the user.

DSSA SUB-STAGE 1-3: DEFINE THE ASAC EA DOMAIN

In this sub-stage, we identify our problem and identify systems with similar problems that have existing solutions. We can then leverage existing solutions in developing our architecture.

Our problem is distributed economic modeling with distributed users in a heterogeneous environment. This problem resembles

- distributed database problems, e.g., an ATM banking network;
- distributed simulation problems, e.g., Army Close Combat Tactical Trainer; and
- heterogeneous environment problems, e.g., the World Wide Web.

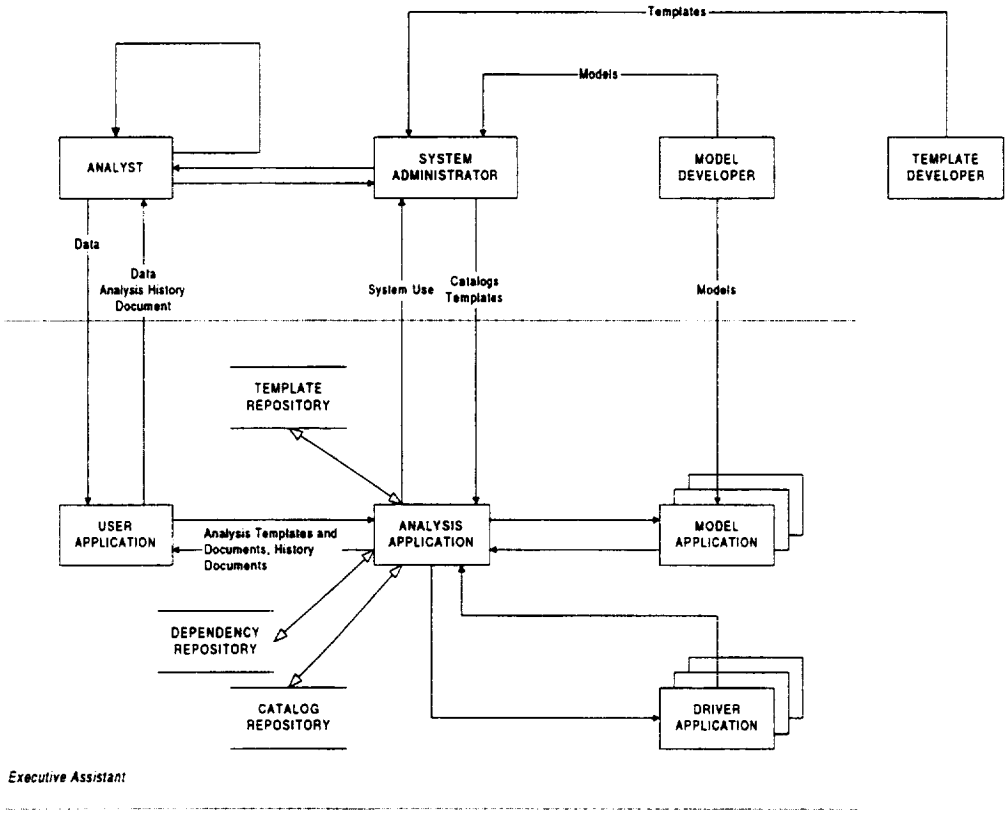
We can use solutions developed for the above problem areas as a starting point for ASAC architecture investigation.

The ASAC Context (block) Diagram, shown in figure 10, is the result of an initial investigation and discussion of the ASAC system. The diagram depicts the high-level relationships among the major components of the ASAC system.

Using the Context Diagram, we can define the following objects in the ASAC EA system:

- Analysis Application
- Driver Application
- Model Application
- User Application.

Figure 10. ASAC Architecture Context Diagram



We can also define items that are outside the ASAC EA domain (they will not be part of our solutions). They include the following:

- Spreadsheet applications
- Graphing applications
- Word processing applications
- Browser applications
- E-mail applications
- Standalone legacy models

- Other ASAC components, e.g., the QRS
- User Local Area Network (LAN)
- User Wide Area Network (WAN).

Inputs to the ASAC EA domain include the following:

- Data inputs
- Models
- Templates
- Catalogs.

Outputs from the ASAC EA domain include the following:

- Data outputs
- Analysis history document
- System use.

ASAC analysts, model and template developers, and ASAC EA system administrators generate inputs. Outputs go to ASAC analysts and ASAC EA system administrators.

DSSA SUB-STAGE 1-4: DEFINE ASAC EA-SPECIFIC RESOURCES

Domain experts

Domain experts are individuals whose expertise and experience in the domain can lend insight into various aspects of the domain. Domain experts have been identified in two broad areas. For the ASAC domain and future user needs for applications in the ASAC domain, the domain experts are members of the Technology Integration Steering Committee.

For system design details, hardware constraints and dependencies, future technology that may be incorporated into new applications in this domain, the domain

experts are LMI research staff and external DSSA experts, such as Dr. Will Tracz[5].

Domain Artifacts

Identifying domain artifacts brings to light systems and processes that should be considered in developing the ASAC architecture.

Resources that exist include

- legacy models, e.g., ACSYNT, FLOPS, NARIM, and the first generation ACIM and
- research tools developed for ASAC studies, e.g., wind optimization routines.

The current process, which the ASAC EA will replace, is largely a manual process that uses the following disjointed automated tools:

- Telephone
- PC tools
- Pen and paper
- E-mail
- Developed ASAC components, e.g., data repositories and the QRS.

DSSA SUB-STAGE 1-5: RE-DEFINE THE ASAC EA DOMAIN (NARROW THE DOMAIN OF INTEREST)

Once we have defined the application domain, exploring all the possible implementation and design trade-offs or developing all possible implementations in the solution space is not economically feasible. This section will narrow our problem space.

Technologies of interest are derived from user requirements, the ASAC domain, similar problems and solutions, and domain-specific resources. Technologies that

[5] We visited Dr. Will Tracz, one of the principal investigators on the DSSA, at Lockheed-Martin Corporation, Owego, NY on two occasions, 19 and 20 August 1996 and 19 and 20 December 1996. On the first visit, we reviewed our work and gained a fuller understanding of the DSSA methodology. On the second visit, we reviewed our draft ASAC architecture description.

we are interested in using to build applications in this domain include the following:

- UNIX
- C/C++
- FORTRAN
- Microsoft Windows
- X Window System
- Macintosh OS
- Java
- WWW browsers
- WWW servers
- Client/server application development tools
- Client/server
- IBM-compatible personal computers (PCs)
- Macintosh computers
- Hewlett-Packard workstations and servers
- Silicon Graphics workstations and servers
- RS/6000 workstations and servers
- Sybase database
- TCP/IP
- SQL
- CORBA
- DCE.

Identifying technologies that are not of interest is an equally important task. They include

- operating systems that are not requirements, e.g., OS/2, and
- development tools that are not specifically intended for our type of problem, e.g., Microsoft Visual Basic and Powersoft PowerBuilder.

The methodology we are interested in using to build the applications in the ASAC EA domain is the Object Modeling Technique and associated CASE tools.

The documentation standards we are interested in using to build applications in the ASAC EA domain are NASA report standards and LMI defined standards, e.g., software development documentation standards.

Applicable Standards

Our applicable standards, listed in Table 6, are largely a subset of the standards listed in the National Institute of Standards and Technology (NIST) Application Portability Profile (APP) Open System Environment Profile (OSE). As stated in the NIST APP OSE,

> An OSE encompasses the functionality needed to provide interoperability, portability, and scalability of computerized applications distributed across networks of heterogeneous, multivendor hardware/software/communications platforms. The OSE forms an extensible framework that allows services, interfaces, protocols, and supporting data formats to be defined in terms of nonproprietary specifications that evolve though open (public), consensus-based forums.

We feel this description, along with the appropriate standards, is applicable to the ASAC EA domain. Also, if we use these standards, services outside the ASAC EA domain will be interoperable with services within the ASAC EA domain.

Table 6. Applicable Standards

NIST APP Service	Specification Title
Operating System Service—Kernel Operations API	FIPS 151-2 Portable Operating System Interface (POSIX)—System Application Program Interface [C Language]
Operating System Service—Operating System Commands and Utilities	FIPS 189 Portable Operating System Interface (POSIX); Part 2: Shell and Utilities
Human/Computer Interface Service—Graphical User Interface	IBM Common User Access (CUA) Guidelines for Object-Oriented Interface Design[1]
Software Engineering Service—Programming Language C	FIPS 160 C
Software Engineering Service—Programming Language C++	TBD—Under development
Software Engineering Service—Integrated Software Engineering Environment Repository	ISO/IEC 13719-1 Portable Common Tool Environment (PCTE) Application Programmer's Interface
Data Management Service—Relational Database Management System	FIPS 127-2 Database Language SQL
Network Service—Communication Protocols	FIPS 146-2 Profiles for Open Systems Internetworking Technologies (POSIT)
Network Service—Remote Procedure Call	Open Software Foundation (OSF) Distributed Computing Environment (DCE) Remote Procedure Call (RPC) Component
Network Service—Object Request Broker	The Common Object Request Broker: Architecture and Specification (Revision 2.0)
Network Service—Object Request Broker	CORBAservices: Common Object Services Specification (Revised Edition)
Network Service—Electronic Messaging API	X.400 Based Electronic messaging Application Program Interface (API) IEEE 1224.1
Network Service—Directory Services API	Directory Services Application Program Interface (API) IEEE 1224.2
Network Service—Distributed Information Services	ANSI/NISO Z39.50 Information Retrieval Service and Protocol
Network Service—Data Encryption	FIPS 46-2 Data Encryption Standard (DES)
Network Service—Digital Signatures	FIPS 186 Digital Signature Standard (DSS)
Other Services	**Specification Title**
Software Engineering Service—Programming Language Java	Java 2.0
Data Interchange Service—Document Markup Language	HyperText Markup Language (HTML) 3.2

[1]The NIST APP standard for graphical user interface is FIPS 158-1 X Window System using OSF MOTIF. We chose to use the IBM CUA standard as a guide for ASAC graphical user interfaces. MOTIF is based on and similar to the IBM CUA, however, the IBM CUA is more general and better suited to our application.

DSSA SUB-STAGE 1-6: DETERMINE VERIFICATION PROCEDURE

The domain experts—members of the Technology Integration Steering Committee, LMI Research Staff, and DSSA experts—will review the architecture (domain model, requirements and solution).

DSSA SUB-STAGE 1-7: REVIEW AND ITERATE

During this stage, we will review and iterate the items defined in DSSA Stage 1.

DSSA Stage 2—Define and Scope Domain-Specific Elements

Compiling a dictionary of domain-specific terminology is the goal for this phase of the domain-engineering process. In this stage, control flow and data flow information is added to the context diagram created in Stage 1. Object diagrams are also created in this phase. This stage defines what can be accomplished with emphasis on the problem space.

DSSA stage 2 includes the following sub stages:

- 2-1 Define and refine an element
- 2-2 Define and refine application services
- 2-3 Refine the context diagram
- 2-4 Identify relationships between elements
- 2-5 Create a domain dictionary
- 2-6 Create a high-level requirements specification
- 2-7 Define roles
- 2-8 Define assumptions
- 2-9 Define issues
- 2-10 Review and iterate.

DSSA SUB-STAGE 2-1: DEFINE AND REFINE AN ELEMENT

Identify a Domain Element

The following major elements are identified in DSSA Stage 1:

- User Application
- Analysis Application
- Model Application
- Driver Application
- Repository
- Analysis Template/Document
- History Document.

Identify Attributes of Elements

User Application

The user application is the interface between the user and the analysis application. The user application accepts inputs from the user and sends requests to the analysis application.

User application responsibilities include the following:

- Display a graphical user interface (GUI)
- Accept input from the user
- Send requests to the analysis application
- Display results for the user
- Input and output data
- Control local disk input/output
- Process data

- Establish connection with the analysis application
- Provide user login for the analysis application
- Provide system administration and control.

Analysis Application

The analysis application will act as an integration agent, pull information from the various models, and coordinate the data flow between models and the user application. The user application will submit requests for information to the analysis application, which will then execute the appropriate model(s) and send the results back to the user application.

The analysis application also controls the drivers that perform such functions as optimization and sensitivity analysis.

Analysis application capabilities include the following:

- Accept inputs from the user application
- Accept execution control commands from the user application
- Pass data to and from the user application
- Know how to run a model
- Know how to run a driver
- Know the available models in the system
- Know the available drivers in the system
- Coordinate execution of multiple models, including scheduling, queuing, and load balancing
- Know model dependencies
- Know driver dependencies
- Process data
- Control disk input/output

- Query data
- Establish connection to models
- Authenticate users
- Configure model
- Configure driver.

Model Application

The models are the key analytical elements that provide information to the user. The model application controls the models. The analysis application coordinates and sends inputs to multiple models. A model can be a complex program that generates custom output or a database query that performs a simple look-up for a given set of inputs.

Model application capabilities include the following:

- Know how to execute a model
- Know which inputs are necessary
- Process data
- Query data
- Input and output data
- Control disk input/output.

Special Consideration for Legacy Models

Legacy (existing) models that operate from input files and generate output files may need a special interface to integrate with the ASAC EA. Support programs to introduce input information into the appropriate input files or extract output information from the necessary files may also be required.

Driver Application

Drivers provide the analytical capability necessary for optimization, sensitivity analysis, and response surface generation.

Driver application capabilities include the following:

- Set input information for a series of runs
- Process data
- Input and output data.

Repository

The repository is a database that contains the catalog, template, and dependency repositories. The catalog repository contains the registrations for all models and drivers in the ASAC EA. The template repository contains all global templates available in the ASAC EA. The dependency repository contains the dependencies for all models and drivers in the ASAC EA.

Analysis Template and Document

The analysis template, which is generic, provides an outline for performing specific types of analyses. When the user loads the template for a specific task, he or she uses the analysis template to create an analysis document.

History Document

The history document contains the result(s) of running an analysis using an analysis document.

Table 7 lists the seven elements described above.

Table 7. Element Definitions

Element	Description	
	Data	Functional
User Application		X
Analysis Application		X
Model Application		X
Driver Application		X
Repository	X	
Analysis Template/Document	X	
History Document	X	

DSSA SUB-STAGE 2-2: DEFINE AND REFINE APPLICATION SERVICES

The ASAC system comprises the following application services:

- Distributed Computing Service
- Presentation Service
- Data Service
- Management Service
- Communication Service
- Common Support Service.

In this sub-stage, we identify their attributes.

A single module or component may provide many of the services identified in this sub-stage. We will make this decision in the design phase of ASAC EA development.

Distributed Computing Service

The Distributed Computing Service provides specialized support for applications that may be dispersed among computer systems in the network but must maintain a cooperative processing environment.

It is composed of the following services:

- Remote Process Service
- Directory Service
- Data Interchange Service
- Analysis (Broker) Service
- Thread Management Service.

Remote Process Service

The Remote Process Service provides remote procedure call (RPC) capability to the system.

Directory Service

The Directory Service maintains a dynamic list of all application services, models, and drivers throughout the domain. When a client makes a request, the Directory Service locates an application service that can handle the request and tells the client how to handle the request.

Data Interchange Service

The Data Interchange Service provides support for applications that may be dispersed among computer systems in the network but must maintain a cooperative processing environment.

Analysis (Broker) Service

With the Analysis (Broker) Service, applications can use methods or objects that are remote. Objects can dynamically discover each other and interact across machines, operating systems, and networks.

Thread Management Service

The Thread Management Service provides thread management capabilities for applicable application services.

Presentation Service

The Presentation Service is the user interface layer. Through this service, the user formulates a request and receives a reply. This service also displays alert information to the user.

It is composed of the following services:

- User Interface Service
- Alert Notification Service

User Interface Service

The User Interface Service provides direct interaction with a user through windows, icons, menus, keyboard, mouse, or other means.

Alert Notification Service

The Alerts Notification Service presents user and application generated alerts and notifications. For example, if a requested model is unavailable, the system will alert the user.

Data Service

The Data Service provides data administration, management, input and output, and distribution services.

It is composed of the following services:

- Data Administration Service
- Data Management Service
- File input and output (I/O) Service
- Software Distribution Service
- Catalog Service.

Data Administration Service

The Data Administration Service allows a System Administrator to enter, maintain, change, and remove data in repositories.

Data Management Service

The Data Management Service provides DBMS connectivity and performs query transactions.

File I/O Service

The File I/O Service provides file management capability.

Software Distribution Service

The Software Distribution Service provides electronic software distribution capability.

Catalog Service

The Catalog Service registers and assembles model and driver descriptions, configurations, I/O, type, and average execution times.

Management Service

The Management Service provides system, application, error, performance, and security management.

It is composed of the following services:

- Security Administration Service
- System Administration Service
- Application Management Service
- System Management Service
- Audit Service
- Error Management Service

- Performance Monitoring Service
- Security Service

Security Administration Service

The Security Administration Service maintains a registry of all authorized users throughout the domain and tracks which functions each user or group may perform (authorization). This service assigns password protection, model security, groups, and permissions.

System Administration Service

With the System Administration Service users can configure, operate, maintain, and manage the local configuration.

Application Management Service

The Application Management Service maintains a dynamic configuration of application services. It starts the appropriate application services on appropriate machines. It also monitors application services to ensure they are available and performing correctly, restarts lost services, and starts and stops services.

System Management Service

The System Management Service provides EA system management and query governing and handles runaway queries.

Audit Service

The Audit Service provides a permanent record of system usage. It includes user access, model usage, error logging, and time stamping.

Error Management Service

The Error Management Service provides error management for system and application level errors. It preserves the integrity of the system.

Performance Monitoring Service

The Performance Monitoring Service provides routine diagnostics and performance monitoring for the EA system.

Security Service

The Security Service provides a single login service for all systems throughout the domain. It authenticates a user and provides logon transparency.

Communication Service

The Communication Service supports communication among all components within the domain. It provides translation if more than one protocol is used. It also provides facilities for receiving data external to the domain and sending data out of the domain.

It is composed of the following services:

- Communication Management Service
- Network Service
- Intra-application Communication Service
- Transaction Management Service
- Queuing Service
- Load Balancing Service

Communication Management Service

The Communication Management Service provides overarching communication services among system applications and other services.

Network Service

The Network Service connects the external communications network and the EA system.

Intra-applicaiton Communication Service

The Intra-application Communication Service provides communication among components in the EA system.

Transaction Management Service

The Transaction Management Service creates and manages transaction logs and processes transactions.

Queuing Service

The Queuing Service provides process queuing to ensure fair access to system resources.

Load Balancing Service

The Load Balancing Service balances the loads among replicated models to ensure no model is overloaded.

Common Support Service

The Common Support Service provides common system-level support across all components of the EA system.

It is composed of the following services:

- Alerts Service
- Message Service
- Help Service

Alerts Service

The Alerts Service provides functionality for user and application generated alerts and notifications. It routes and manages alert messages throughout the domain.

Message Service

The Message Service handles message traffic (parsing and distribution) among applications. It provides message queuing capability.

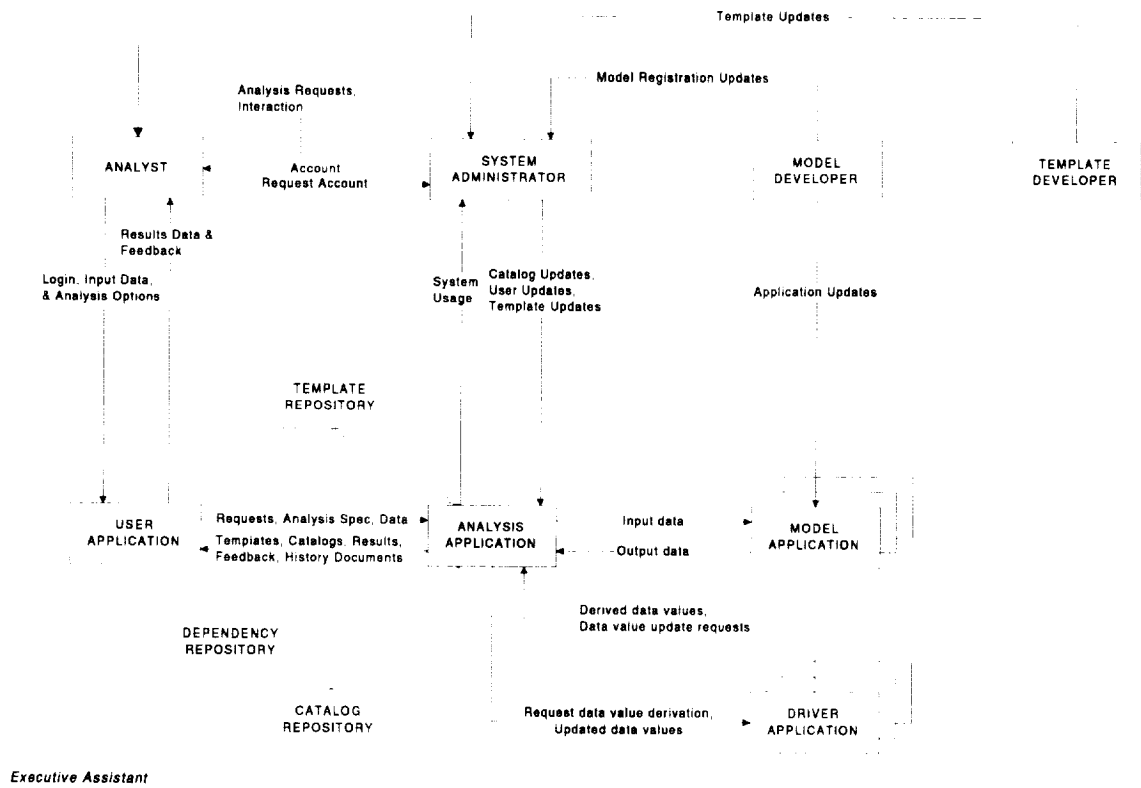
Help Service

The Help Service provides help to users.

DSSA SUB-STAGE 2-3: REFINE THE CONTEXT DIAGRAM

Having defined the above services, we can refine the Context Diagram so that it includes data and control flow information.

Figure 11. Revised ASAC EA Context Diagram



DSSA SUB-STAGE 2-4: IDENTIFY RELATIONSHIPS AMONG ELEMENTS

Creating object diagrams is one method of identifying relationships among elements. Object diagrams depict *is a/is a kind of* and *consists of/part of* relationships among elements.

The following eighteen object diagrams have been developed for the ASAC domain:

- High Level View
- User Application
- Repository
- Catalog Entry
- Analysis Template and Document
- History Document
- Model Data Element
- Execution Point
- Dependency
- Driver
- Application Services
- Distributed Computing Service
- Presentation Service
- Data Service
- Management Service
- Communication Service
- Common Support Service
- People.

We used the Rumbaugh OMT object-oriented methodology to develop the diagrams. The relationships of interest, *is a/is a kind of* and *consists of/part of*, correspond to two types of OMT relationships:

- Aggregation
- Generalization.

Aggregation is a *part of* relationship, represented by a diamond symbol; generalization is an *is a* relationship, represented by a triangle.

Other symbols used in the object diagrams are a rectangle and a rounded rectangle. The rectangle represents an object class; the rounded rectangle an object instance. The text inside both symbols is the class name. The number adjoining a rectangle represents an explicit object order relationship, e.g., 1+ represents a one or more than one to one relationship.

Relationships between object diagrams are denoted by a shaded triangle. The corresponding object diagram figure number(s) is located to the right of the triangle.

The eighteen object diagrams follow.

Figure 12. High Level View Object Diagram

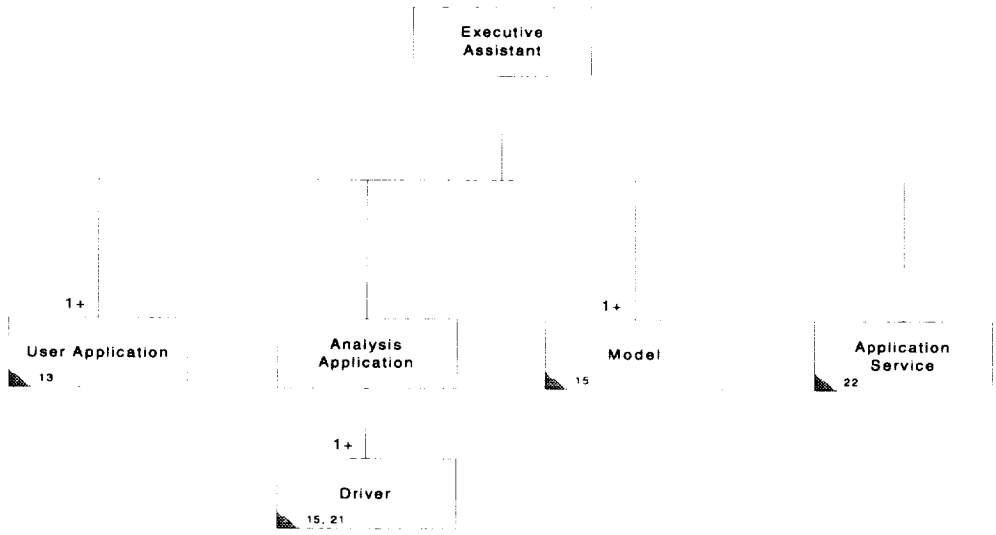


Figure 13. User Application Object Diagram

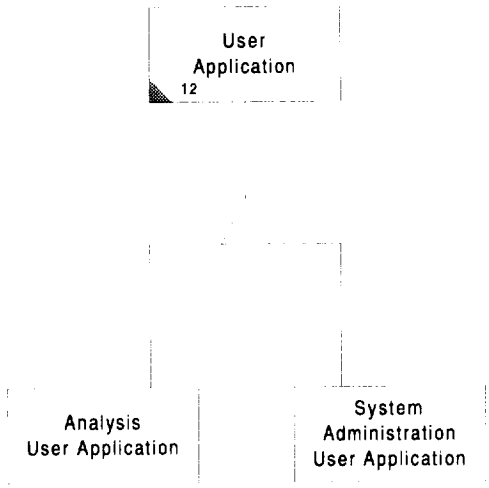


Figure 14. Repository Object Diagram

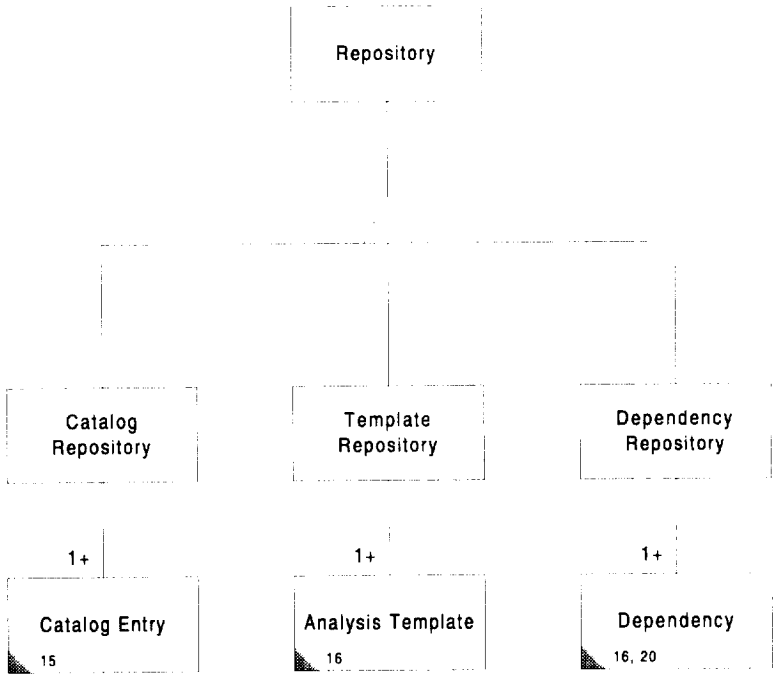


Figure 15. Catalog Entry Object Diagram

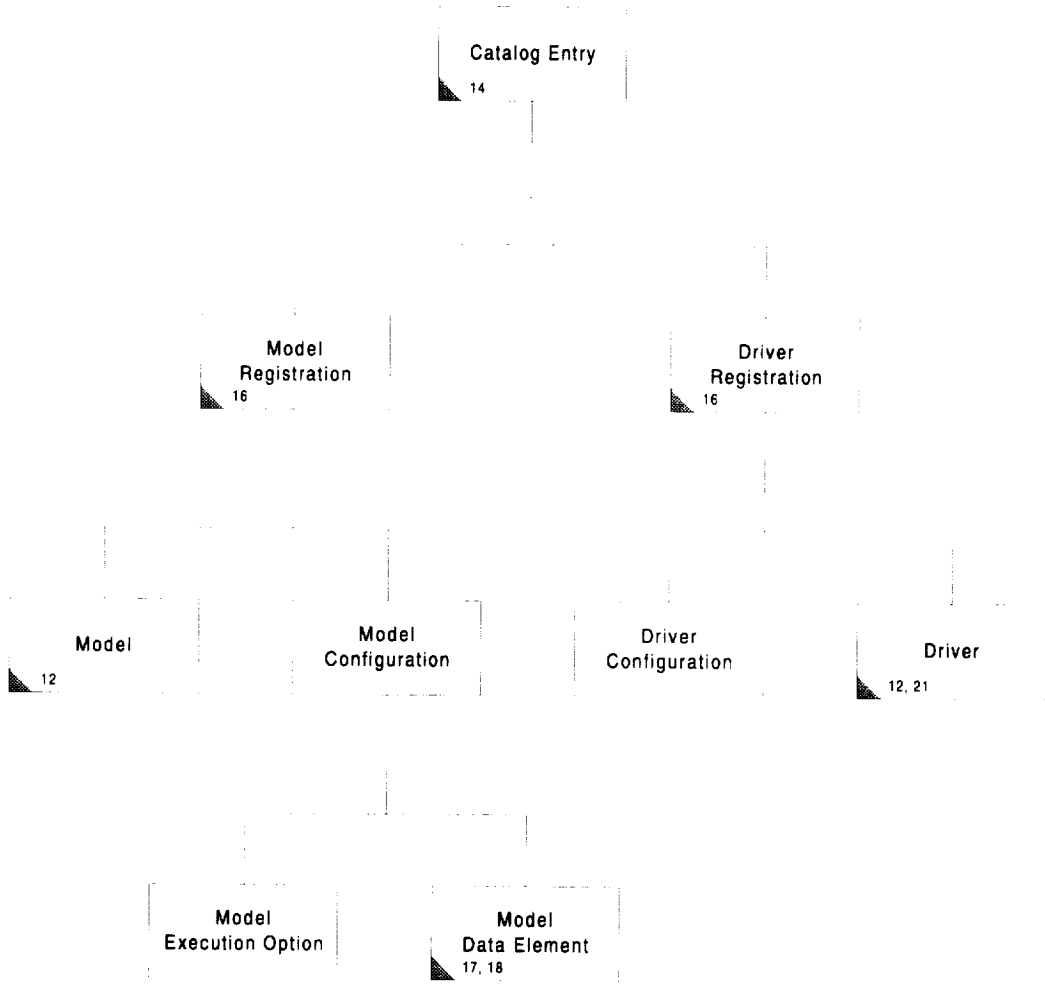


Figure 16. Analysis Template/Document Object Diagram

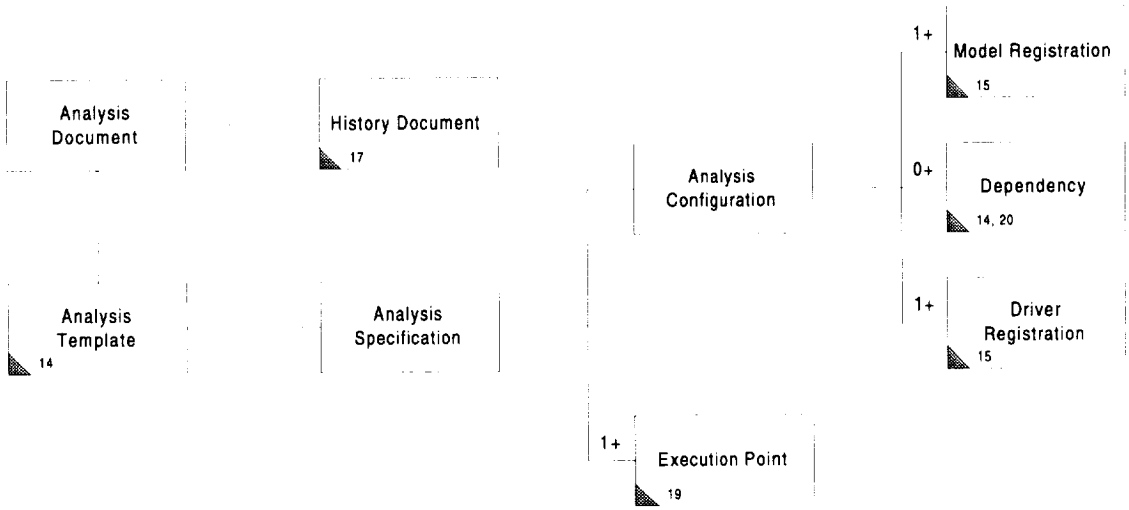


Figure 17. History Document Object Diagram

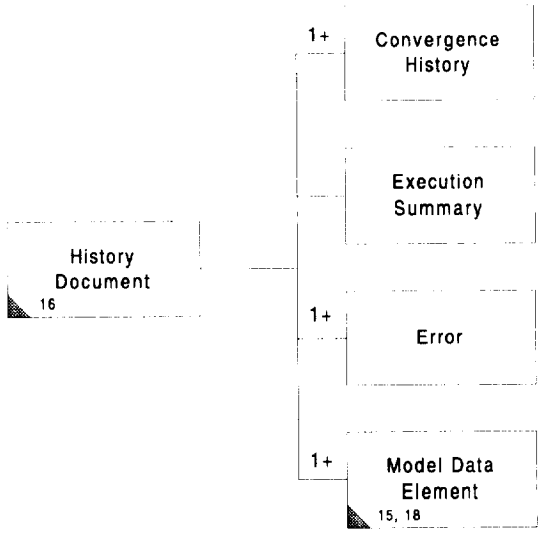


Figure 18. Model Data Element Object Diagram

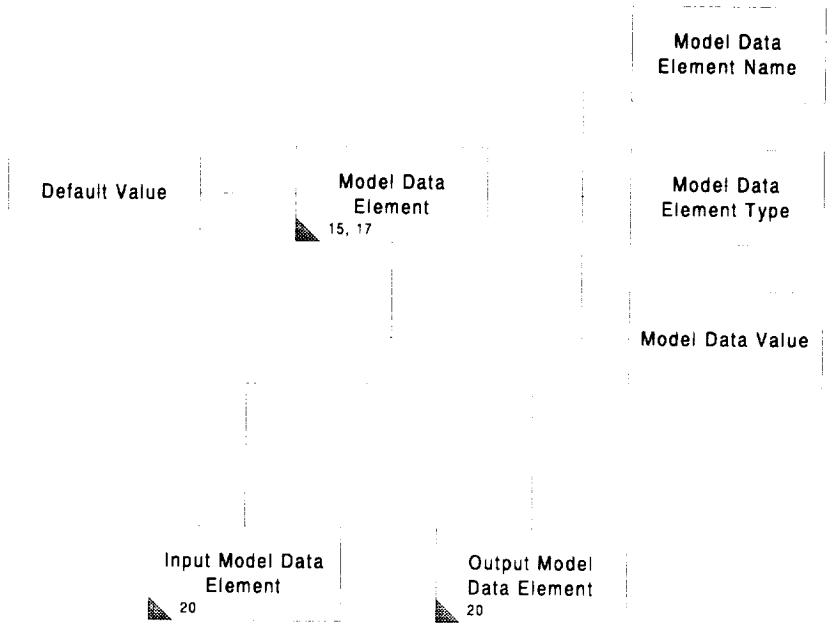


Figure 19. Execution Point Object Diagram

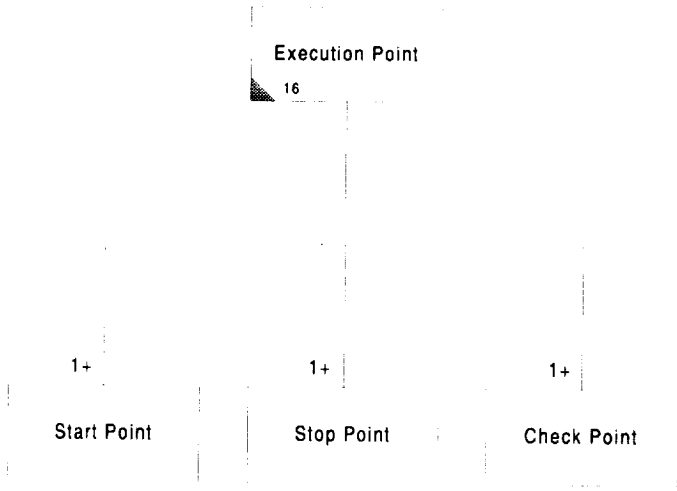


Figure 20. Dependency Object Diagram

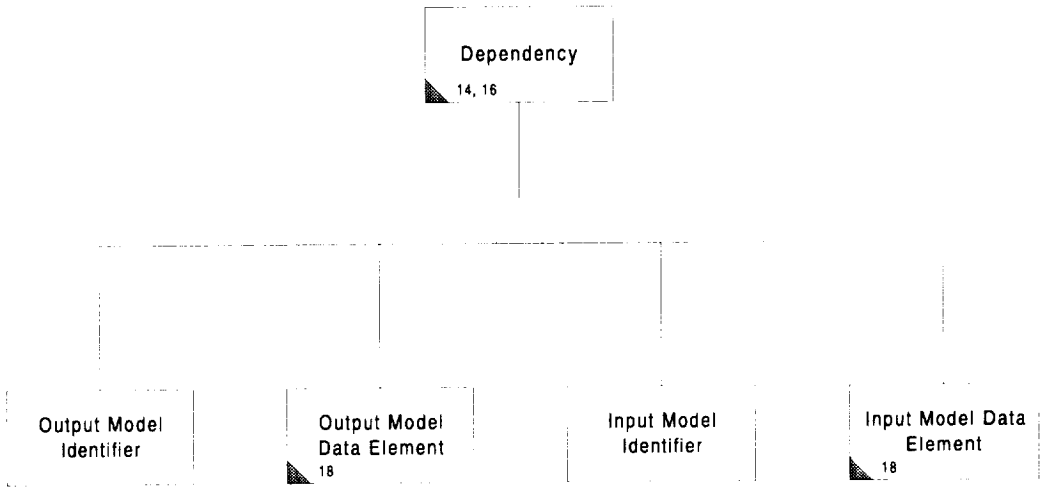


Figure 21. Driver Object Diagram

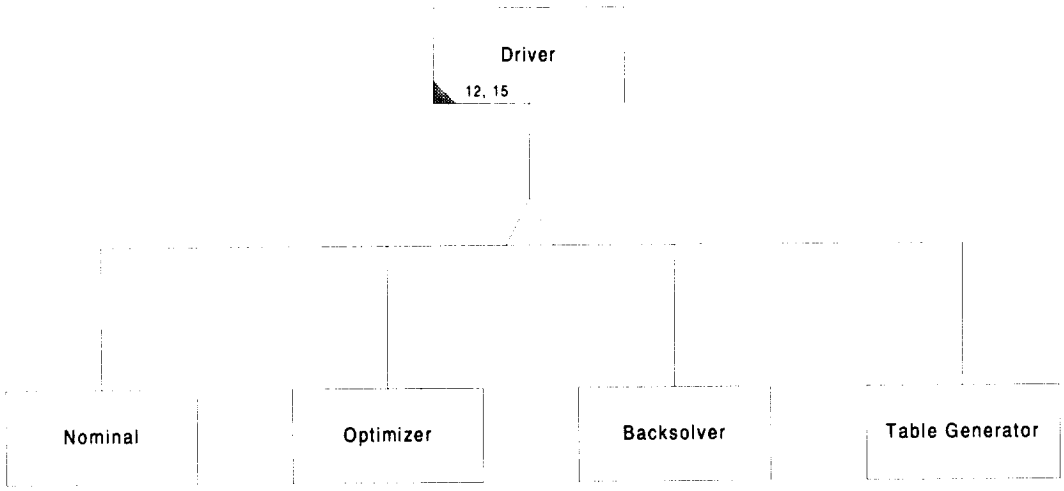


Figure 22. Application Service Object Diagram

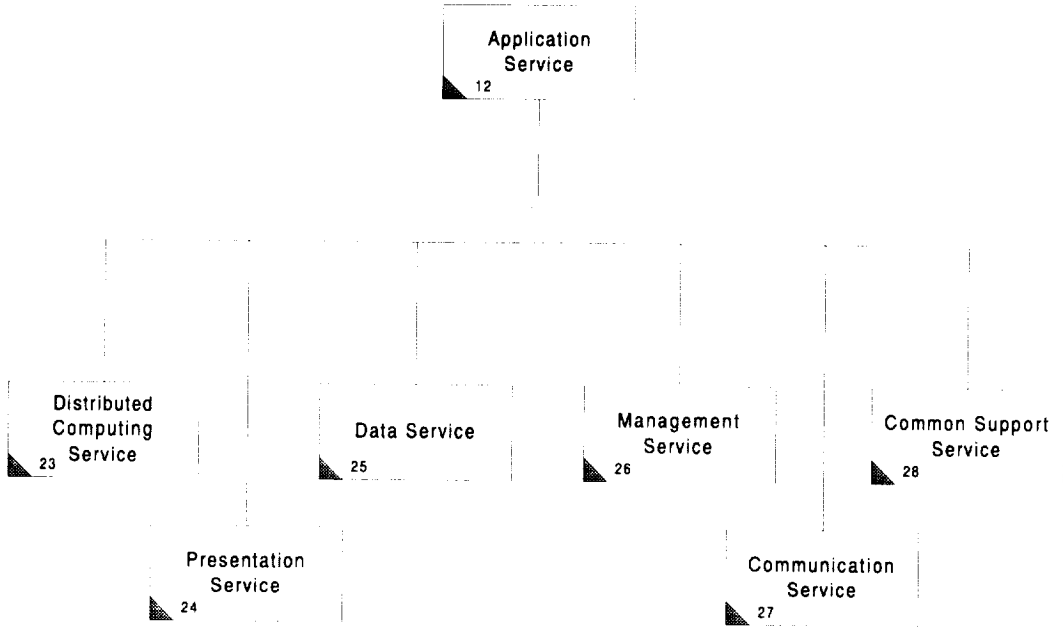


Figure 23. Distributed Computing Service Object Diagram

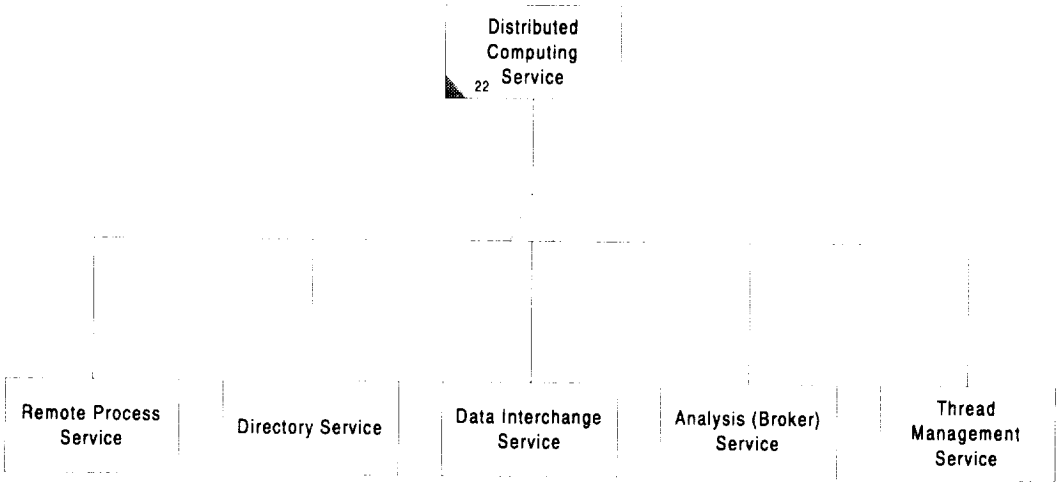


Figure 24. Presentation Service Object Diagram

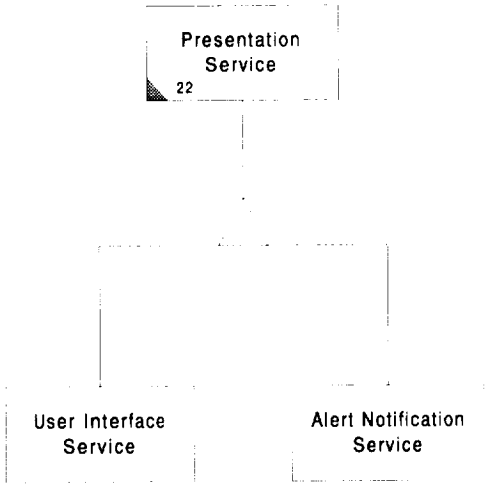


Figure 25. Data Service Object Diagram

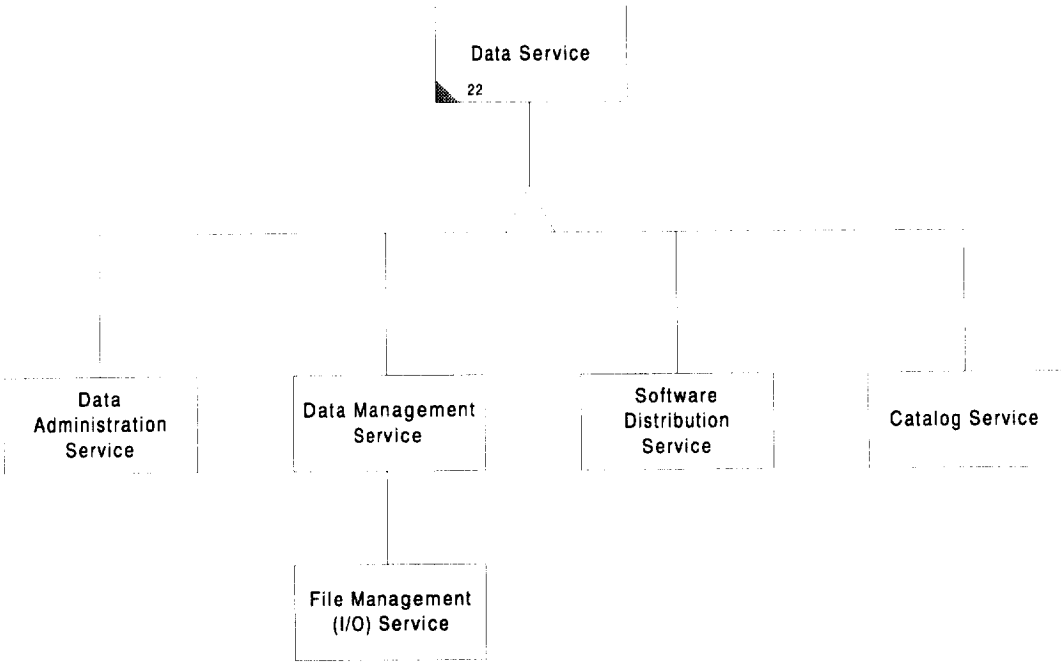


Figure 26. Management Service Object Diagram

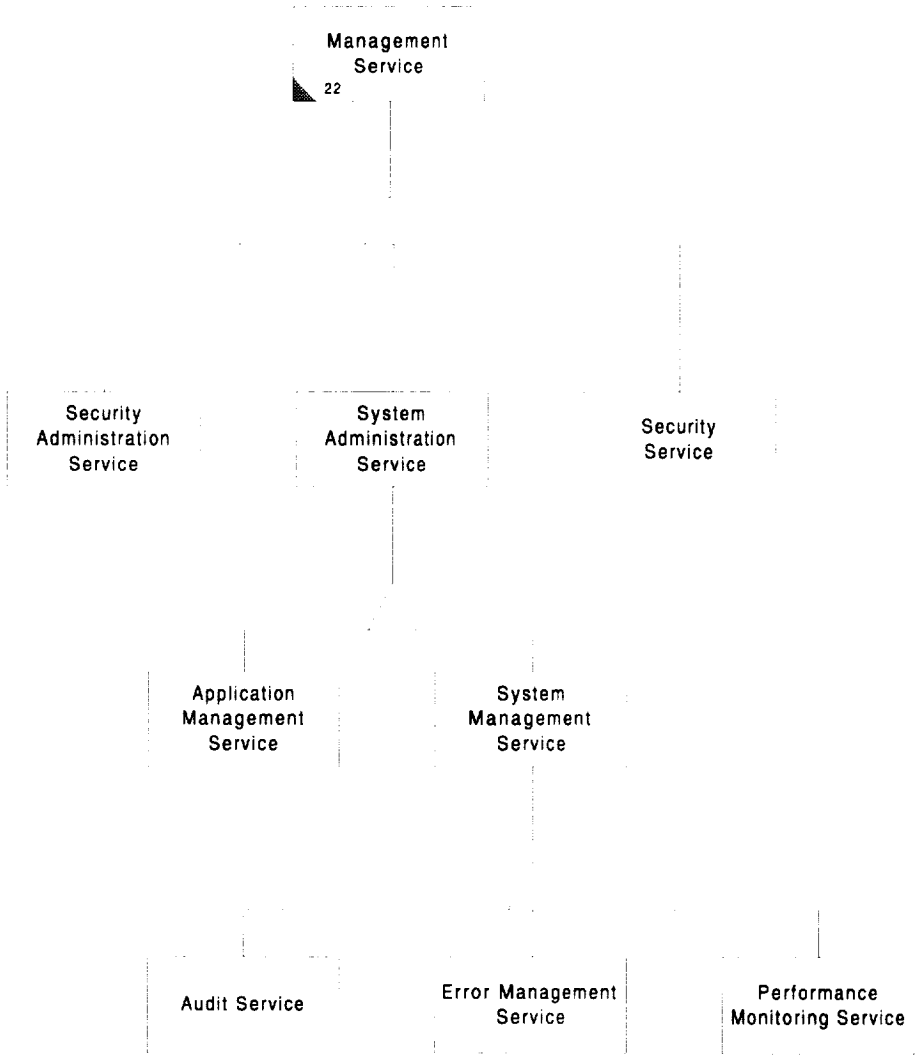


Figure 27. Communication Service Object Diagram

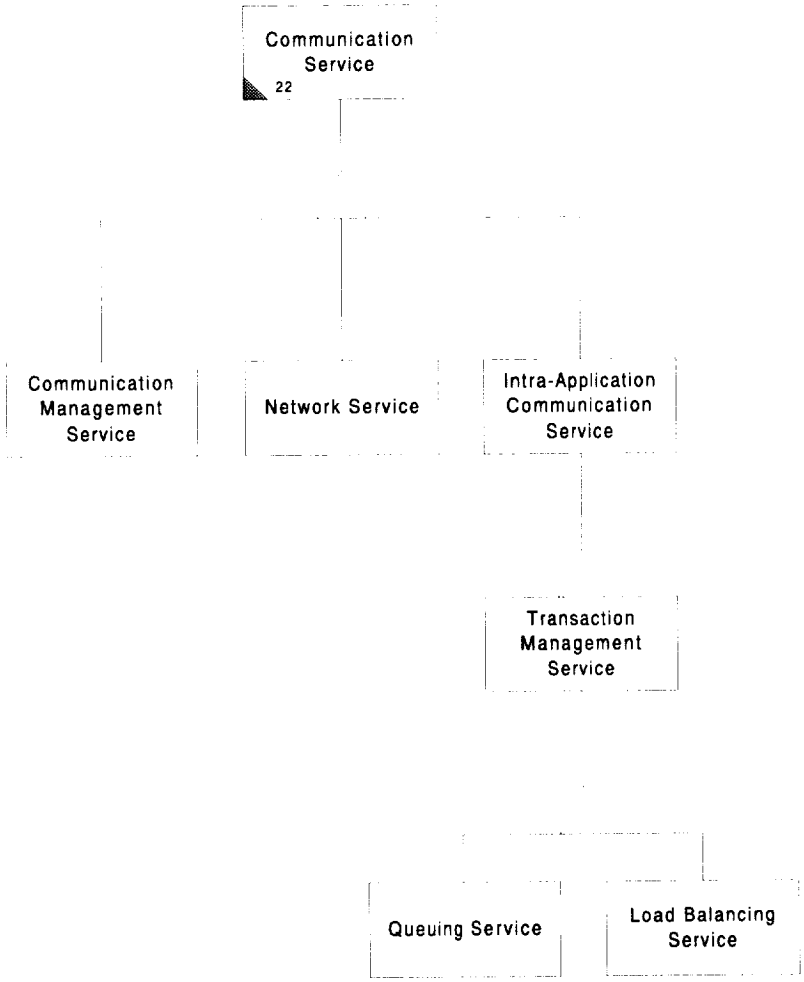


Figure 28. Common Support Service Object Diagram

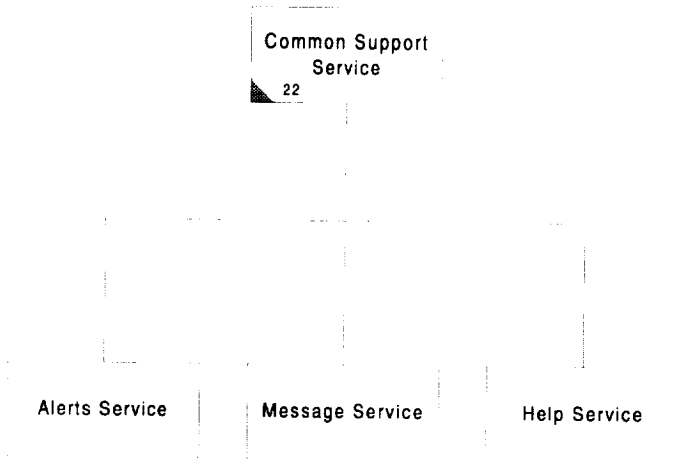
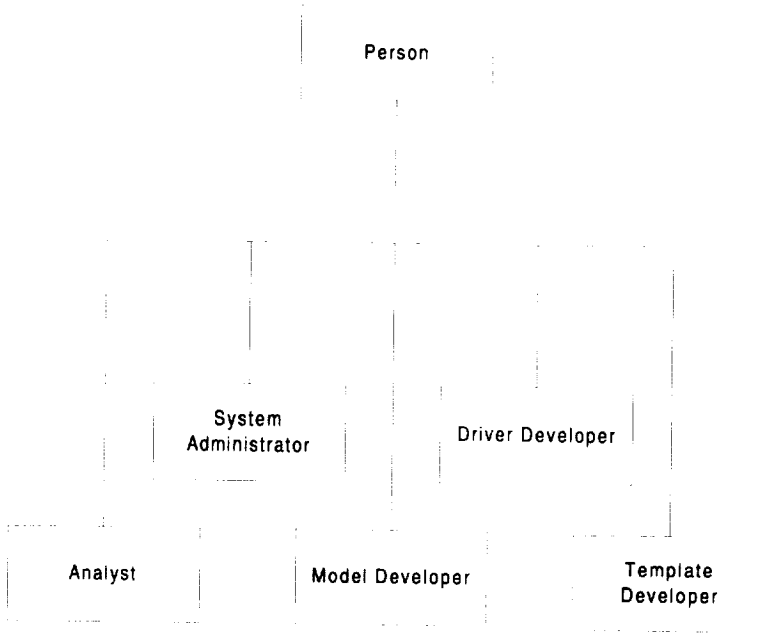


Figure 29. People Object Diagram



DSSA Sub-stage 2-5: ASAC EA Domain Dictionary

We created an ASAC EA Domain Dictionary (Appendix B) to define ASAC EA-specific terminology. The Domain Dictionary serves as a reference for building the ASAC EA and future models and templates.

DSSA Sub-stage 2-6: ASAC EA High-Level Requirements Specification

High-level requirements define the shalls and wills of the ASAC EA system. The following requirements have been added to the requirements defined at the March ASAC architecture meeting:

FUNCTIONAL REQUIREMENTS

1. The EA will integrate results of models run outside the EA into an analysis document within the system.

2. The EA will include a mechanism for evaluating and choosing from future shadow model sites.

3. When the Analysis System fails, the User Interface will detect the failure or time out.

4. The EA will provide valid default data values to models upon initialization.

5. Given a new input to a model, the system will minimize the number of models that need to be rerun. For example, given five models A, B, C, D, and E, where A = f(B + C) and B = f(D + E), when a value of E is changed, only A and B will be rerun (C and D will not be rerun).

NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements also play an important role in defining a system. They do not have an impact on system operation, but they do have an impact on architecture, design, and implementation. They fall into the following categories:

1. Performance

2. Security

3. Network stability
4. Extendibility
5. Scalability
6. Fault tolerance
7. Testability
8. Portability
9. Reusability
10. Usability
11. Interoperability.

Performance

1. Barring system failure, models run through the ASAC system will execute to completion in less than 24 hours.
2. User interface response time will be in accordance with best commercial practices.
3. EA performance time will be in accordance with best commercial practices.

Security

1. User authentication will control access to EA system.
2. User authentication will control access to stored analyses.
3. User authentication will control access to models.

Network stability

None

Extendibility

1. The EA will support the addition of models and drivers.

Scalability

1. The EA will support vertical and horizontal growth.
2. The EA will support replication of model applications.
3. The EA will support load balancing among replicated models.

Fault Tolerance

1. The EA will continue to provide standard services if a model application fails or crashes.
2. The EA will continue to provide standard services if a user application fails or crashes.

Testability

1. The EA will provide a log of system use.
2. The EA will have a built-in debug mode.

Portability

1. User applications will be portable across Personal Computer, Macintosh, and UNIX platforms.
2. Database connectivity will be open, i.e., not restricted to a particular database vendor.

Reusability

1. The interface between the analysis application and models will have reusable components.
2. User application components will be reusable.
3. EA interfaces to models will be modular so that they can be reused.

Usability

1. The user application will have an intuitive graphical user interface that adheres to the IBM CUA standards and, therefore, minimizes the need for user training.

2. The user application user interface will tolerate user errors.

3. The user application user interface will help the user follow the proper analysis process.

4. The help service will help with the user application.

Interoperability

1. The user application will be interoperable with WWW browsers supporting Java 2.0 and HTML 3.2.

DSSA Sub-stage 2-7: Define Roles

Persons involved with the ASAC EA system will have one or more of the roles listed in Table 8:

Table 8. ASAC Roles

Person	Role
Analyst	Login, logoff, input data, create template, create analysis document, start analysis, stop analysis, specify start point, specify stop point, specify checkpoint, choose models, choose drivers, view results, monitor progress of analysis, develop model, develop driver.
System Administrator	Assign login names and passwords, remove login names and passwords, monitor system usage, register new models, register new drivers, remove models, remove drivers, kill runaway processes, administer system processes, perform load balancing
Model Developer	Create model, register model with System Administrator
Template Developer	Create template, register template with System Administrator

DSSA Sub-stage 2-8: Define Assumptions

While creating the ASAC EA system architecture, we made the following assumptions:

1. Users have some expertise with models.
2. The system will have 30 to 40 models.
3. The system will have 30 to 40 end users.
4. The system will not use data encryption for security purposes.
5. Model inputs can be overridden manually.
6. All inputs have default values (templates are runable with options and configurations set).
7. Model progress is indicated to users through feedback.
8. The execution environment supports a multi-tasking or task-switching system.
9. The system has a password-protected security system.
10. No checkpoints are within a driver-wrapped model.
11. Inputs, outputs, configurations, and options for models and drivers are specified in a standard way.
12. The EA does not provide additional output formatting or graphing.
13. If an analysis is halted in-progress, the user will only get data up to completion of previous model (e.g., start of an optimization).
14. The system treats multiple models within a driver as one model (atomically).
15. Catalog updates are done manually.
16. Model versions are handled by naming convention.

17. Models must be registered with the EA to be used in the EA system. Models that are not registered must be run separately and their results must be manually entered into the EA system.

18. The analysis document does not have version tracking.

19. At a checkpoint, the system returns current results to the user.

20. The system will be designed so new technology can be injected in the future.

21. Users cannot substitute their own drivers. They can use ASAC system drivers only.

22. ASAC EA will use a standard suite of optimizers.

23. Models that take longer than 24 hours to run will be run by special request through the ASAC system administrator. They are deemed to be outside the ASAC system.

24. Users will be running on a 32-bit, multi-tasking or task-switching, windowed operating system, e.g., Windows 9X, Macintosh System 7, Windows NT, and UNIX.

25. ASAC will not have shadow sites; however, on the basis of ASAC system usage and performance, we will reevaluate the need for shadow sites in the future.

26. ASAC may have database replication; on the basis of ASAC system usage and performance, we will reevaluate the need for database replication in the future.

27. In the event of a model death, the analysis application has no responsibility to maintain the integrity of a model run. The system will abort the current analysis, notify the user of the model death, and return any analysis results that were calculated before the start of the dead model. In a worst case scenario, the entire in-process analysis will be lost.

28. In the event of analysis application death, only minimum system shutdown and crash recovery service will be provided. In a worst case scenario, the entire in-process analysis will be lost.

29. In the event of analysis application death, the ASAC System Administrator must manually cleanup and restart the application.

30. The ASAC System Administrator must register every model in the system. The system is not responsible for automatically detecting models or registration.

31. Multiple instances of a model from separate analyses may run simultaneously.

32. Within an analysis, only one model will run at a time (concurrent model configuration does not require concurrency).

33. The EA system will use open system standards when practical. No extraordinary attempt will be made to build a completely open system.

34. All EA system access will be through the EA system tools.

35. The EA system will be standards compliant. Deviation from standards will be justified and documented in the design phase.

36. The analysis application will run on Hewlett-Packard UNIX platforms only.

37. The models will be portable across UNIX platforms.

38. The model application will be portable across UNIX platforms.

39. Models are their own application.

40. If a model requires data from the QRS database, the model platform must support Sybase client libraries (ct-lib).

41. ASAC EA models will be under configuration management and control.

42. Models will be developed in accordance with established EA model development standards.

DSSA Sub-stage 2.9: Define Issues

Items that remain unresolvable at this point are deemed issues. The following issues must be resolved in the follow-on phase of ASAC EA system development:

1. How does the EA system handle or detect non-termination of models?
2. How is data passed among components? Pass data or data file name?
3. Should multiple processes be spawned for the analysis application, or should there be separate invocations of the program?
4. What are the space constraints on user systems (maximum size for the user application)?
5. What is the target size of the analysis applications?

DSSA Sub-stage 2-10: Review and Iterate

During this stage, we will review and iterate the items developed in Stage 2.

DSSA Stage 3—Define and Refine Domain-Specific Design and Implementation Constraints

The goal for this phase of the domain-engineering process is to characterize the discriminating features in the solution space.

DSSA Stage 3 has the following sub-stages:

- 3.1 Define constraints on the architecture
- 3.2 Review and iterate

DSSA Sub-stage 3-1: Define Constraints on the Architecture

To define constraints on the architecture, we focus on technologies that are of interest to us in developing the ASAC EA system. This focus narrows the solution space that is available for our use. We defined the following constraints:

- Implementation constraints
- Design constraints
- Software constraints

- Hardware and physical constraints
- Performance constraints

IMPLEMENTATION CONSTRAINTS

- The ASAC EA will support users running on the following platforms:
  - Personal Computer
  - Macintosh
  - HP
  - Sun
  - SGI
  - RS6000 (future)
- Models must be able to run on a socket-based multi-tasking environment (initially UNIX)
- The analysis application must be able to run on HP-UX v9.0 or above

DESIGN CONSTRAINTS

In addition to the non-functional requirements that will affect overall ASAC EA design, we have defined the following design contraint:

- The ASAC EA will be designed using object-oriented methodology

SOFTWARE CONSTRAINTS

- We will use the following languages to develop the ASAC EA:
  - C
  - C++
  - Perl
  - UNIX shell scripts

- Java
- HyperText Markup Language (HTML)
- Structured Query Language (SQL)

- The ASAC EA will support users running the following operating systems:
  - Microsoft Windows 3.1
  - Microsoft Windows 95
  - Macintosh System 7
  - UNIX
    - HP-UX v9.0 or greater
    - SunOS v5.4 or above
    - SGI IRIX v5.3 or above
    - AIX (future)
- The ASAC EA will support, at a minimum, the Sybase System 11 or above.
- The ASAC EA will support Transmission Control Protocol/Internet Protocol (TCP/IP)
- The ASAC EA will support no external software interfaces or development standards
- The ASAC EA will support the following browser software standards:
  - HTML v3.2 and above
  - Java 2.0 and above

HARDWARE AND PHYSICAL CONSTRAINTS

- ASAC EA applications will run on the following user platforms:

- Personal Computer
- Macintosh
- HP
- Sun
- SGI
- RS6000 (future)

- The user interface appearance will conform with a CUA compliant windowed application. It may be browser dependent.
- Space constraints for the user application are TBD.
- Space constraints for other ASAC EA applications are TBD.

PERFORMANCE CONSTRAINTS

There are no critical timing issues in the ASAC EA.

DSSA Sub-stage 3-2: Review and Iterate

During this sub stage, we will review and iterate the items developed in DSSA Stage 3.

DSSA STAGE 4—DEVELOP DOMAIN ARCHITECTURE(S)

The goal for this phase of the domain-engineering is to identify generic architecture(s). The emphasis is on defining interfaces and semantics.

The following sub-stages of DSSA Stage 4 will be completed during the ASAC architecture effort:

- 4-1 Develop the object model
- 4-2 Develop user operational concept
- 4-3 Define a sample user scenario

- 4-4 Develop a dynamic model
- 4-5 Develop a functional model
- 4-6 Perform requirements traceability
- 4-7 Review and iterate

The remainder of DSSA Stage 4 and all of DSSA Stage 5 will be completed as part of the follow-on ASAC design effort.

DSSA Sub-stage 4-1: Object Model

The object model describes the static structure of the objects in a system and their relationships. The object model contains the object diagrams that were developed in DSSA Stage 2. An object diagram is a graph in which nodes are object classes and arcs are relationships among classes. We used two types of relationships to develop our object diagrams—Aggregation, *a part of* relationship, represented by a diamond symbol and generalization, an *is a* relationship, represented by a triangle. The object model will contain a third type of relationship, called *Association*, represented by an annotated straight line.

Figure 30. ASAC Object Model

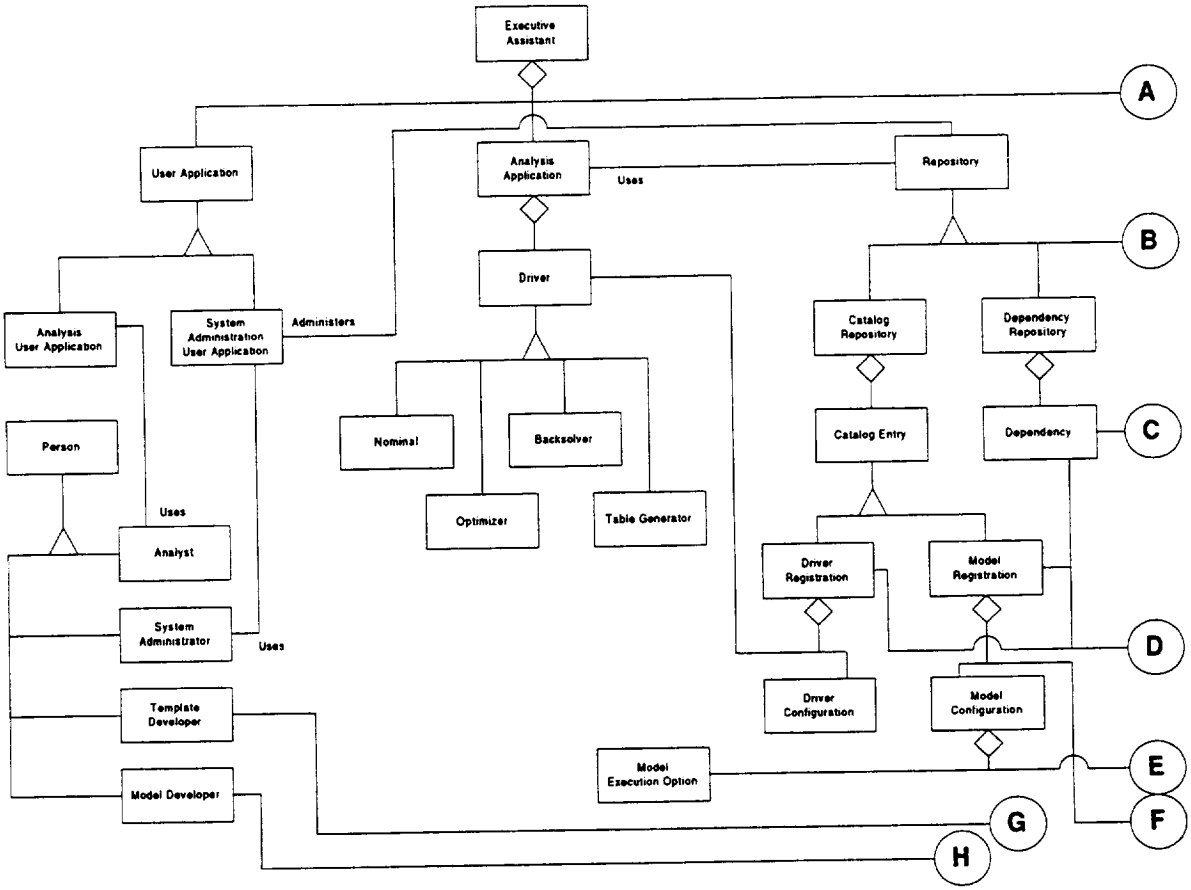


Figure 30. ASAC Object Model (Con't.)

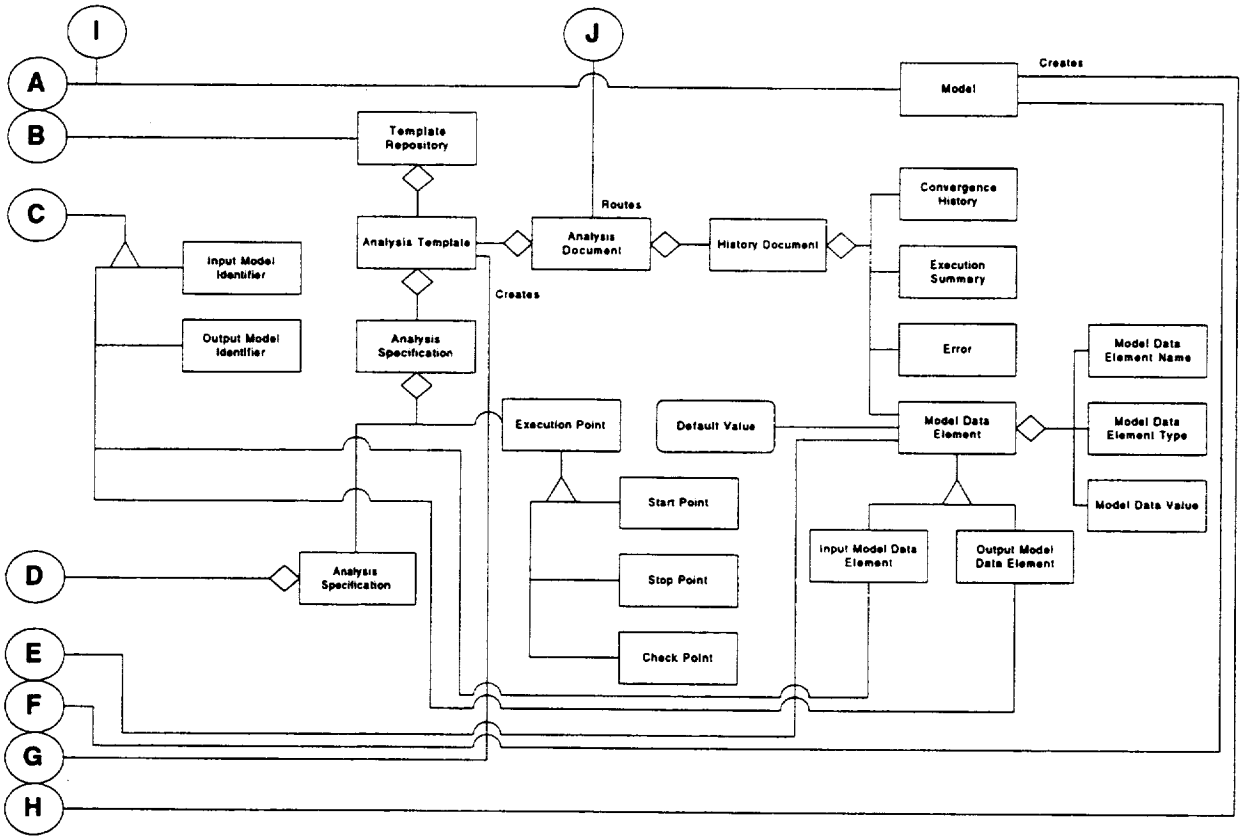


Figure 30. ASAC Object Model (Con't.)

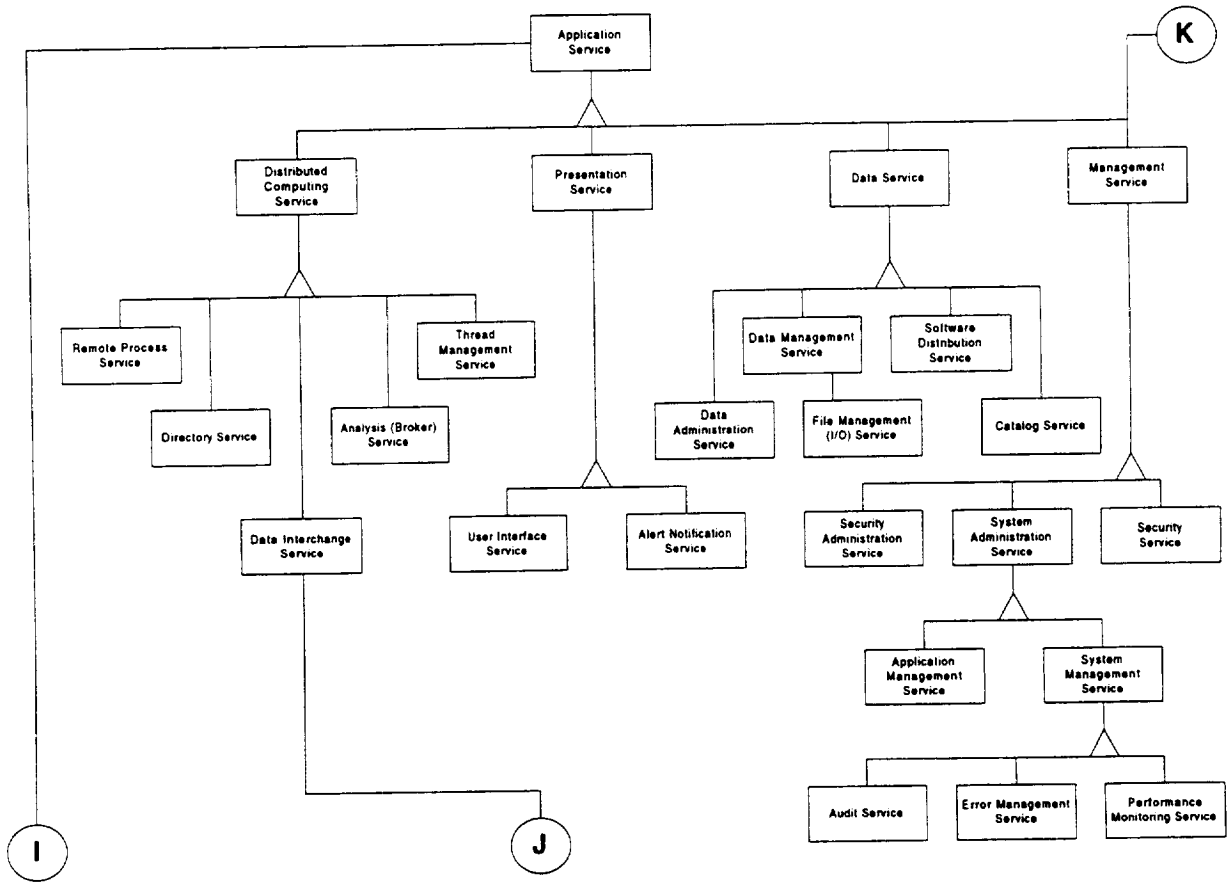
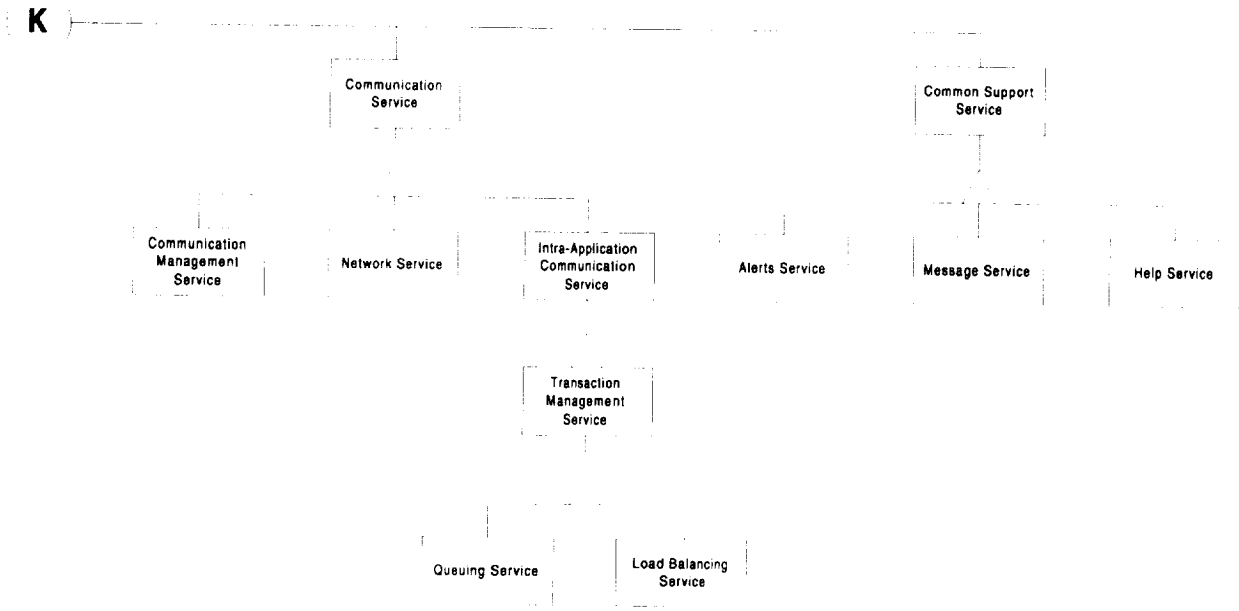


Figure 30. ASAC Object Model (Con't.)



DSSA Sub-stage 4-2: User Operational Concept

This section presents a sequence of options that are available to a user at any point. This sequence of options establishes an operational concept for users of the ASAC EA system. From the user operational concept, we can define a user scenario and associated models and diagrams. In this section, we discuss the following operational concepts:

- User login to EA
- Main EA session selection
- Analysis session initialization
- Model selection
- Model dependencies identification
- Driver application
- Analysis setup
- Analysis submission
- Run analysis
- Presentation of results
- File management

1. **User Login to EA**

    1.1. User Application displays Login Screen.

    1.2. User enters user name and password in User Application

    1.3. System validates user name with Security Services

    1.4. Security Services authorizes or denies access to the system according to authentication scheme and returns authorization or denial message to User Application (Security Services).

        1.4.1. Security Services authorizes access.

1.4.2. Security Services denies access.

1.5. Security Services determines security privileges of user.

1.6. Security Services authorizes or denies entry into EA (User Application) or System Management Utility (System Administration Application).

1.6.1. Security Services authorizes access.

1.6.2. Security Services denies access.

2. **Main EA Session Selection**

2.1. User Application displays Analysis Interface.

2.2. User enters Analysis User Application.

2.2.1. User requests list of analysis templates.

2.2.1.1. User Application requests template list from Catalog Services (template repository).

2.2.1.2. Catalog Services retrieves template list, returns it to User Application.

2.2.1.3. User Application formats list of templates.

2.2.1.4. User Application displays list of templates.

2.2.2. User checks status of running analyses.

2.2.3. User downloads results of completed analysis that was left running in the background.

2.2.4. User checks status of system load.

2.2.5. User works with an analysis.

2.2.5.1. User opens an existing analysis (3.0)

2.2.5.2. User creates an analysis from an analysis template. (3.2)

2.3. User enters System Management Utility

2.3.1. Manage system performance.

2.3.1.1.Start system (external).

2.3.1.2.Stop system.

2.3.1.3.Add resources to the system.

2.3.2. Manage system users.

2.3.2.1.Add users.

2.3.2.2.Delete users.

2.3.3. Manage models.

2.3.3.1.Add model registrations.

2.3.3.2.Delete model registrations.

2.3.4. Manage drivers.

2.3.4.1.Add driver registrations.

2.3.4.2.Delete driver registrations.

2.3.5. Manage system usage.

2.3.5.1.Manage system logs.

2.3.5.2.Kill runaway analyses.

2.4. User logs out of EA

2.4.1. User Application disconnects from Analysis Application.

2.4.2. Analysis Application deactivates user privileges.

2.4.3. User Application displays Login Window.

3. **Analysis Session Initialization**

3.1. User chooses existing analysis.

3.1.1. User Application loads existing analysis document from local storage.

3.2. User selects analysis template for new analysis.

3.2.1. User Application graphically highlights selected analysis template.

3.2.2. User requests new analysis from an analysis template.

3.2.2.1. User Application requests default (blank) analysis template from Catalog Services (template repository).

3.2.2.2. User Application requests predefined analysis template from Catalog Services (template repository).

3.2.3. Catalog Services returns analysis template to User Application.

3.2.4. User Application initializes new analysis document from analysis template.

3.3. User Application displays analysis document in analysis workspace.

4. **Model Selection**

4.1. User requests list of model configurations.

4.1.1. User Application requests model configuration list from Catalog Services (catalog repository).

4.1.2. Catalog Services retrieves model configuration list and returns it to User Application.

4.1.3. User Application organizes list of model configurations by chosen format.

4.1.3.1.User Application organizes list of model configurations by Category (default).

4.1.3.2.User Application organizes list of model configurations by Keyword.

4.1.3.3.User Application organizes list of model configurations by Name.

4.1.4. User Application displays list of model configurations.

4.2. User selects a model configuration for action.

4.2.1. User Application graphically highlights selected model configuration.

4.2.2. User requests basic information for model configuration.

4.2.2.1.User Application displays basic information for model configuration.

4.2.2.1.1.User Application displays model configuration name.

4.2.2.1.2.User Application displays model configuration description.

4.2.3. User requests detailed information for model configuration.

4.2.3.1.User Application displays detailed information for model configuration.

4.2.3.1.1.User Application displays inputs.

4.2.3.1.2.User Application displays outputs.

4.2.3.1.3.User Application displays model configuration options.

4.2.3.1.4.User Application displays model function.

4.2.4. User adds model configuration to analysis document.

4.2.4.1. User Application creates default dependencies for new model configuration.

4.2.4.2. User Application graphically displays new model configuration as part of analysis.

5. **Model Dependencies Identification**

5.1. System links predefined dependencies among model configurations automatically based on data dictionary names.

5.2. User links output data from one model configuration to dependent input data in another.

5.2.1. User creates dependency links.

5.2.1.1. User Application graphically displays created dependency.

5.2.2. User breaks dependency links.

5.2.2.1. User Application graphically displays broken dependency.

5.2.2.2. User edits input value for broken dependency in User Application.

5.2.3. User Application provides data type conversion between dependent data.

5.2.4. User Application provides units conversion between dependent data.

5.2.5. User links two or more outputs from one model configuration by some formula, function, or filter to a single input in another model configuration.

5.2.5.1. Many outputs to one input through a function.

5.2.5.2. Many inputs to one output through a function.

6. **Driver Application**

6.1. User selects class of driver.

6.1.1. User selects Nominal driver (default).

6.1.2. User selects Optimizer driver.

6.1.3. User selects Backsolver driver.

6.1.4. User selects Table generator driver.

6.2. User associates driver with applicable model configurations.

6.3. User identifies parameters of applicable driver.

6.3.1. User identifies parameters of Nominal driver.

6.3.2. User identifies parameters of Optimizer driver.

6.3.2.1.User identifies merit function.

6.3.2.2.User identifies bounds of optimization.

6.3.3. User identifies parameters of Backsolver driver.

6.3.3.1.User identifies merit function.

6.3.3.2.User identifies bounds of backsolver.

6.3.4. User identifies parameters of Table generator driver.

6.3.4.1.User identifies merit function (x and y axes).

6.3.4.2.User identifies bounds of table generation.

7. **Analysis Setup**

7.1. User edits model configuration parameters.

7.1.1. User edits values of non-dependent input data elements to model configuration.

7.1.2. User edits visibility of output data from model configurations.

7.1.3. User edits model configuration options.

7.2. User sets analysis execution points.

7.2.1. User modifies default start point.

7.2.2. User modifies default stop point.

7.2.3. User sets breakpoints.

7.2.4. User removes breakpoints.

8. **Analysis Submission**

8.1. User submits analysis.

8.1.1. User Application updates and saves analysis document information.

8.1.2. User Application requests that Analysis Application run analysis using current analysis document.

8.1.3. Analysis Application interprets analysis specification.

8.1.4. Analysis Application chooses appropriate Driver Application on the basis of analysis specification.

8.1.5. Analysis Application configures Driver Application for analysis.

8.1.6. Driver Application requests data update from Analysis Application.

8.1.7. Analysis Application requests data value update from appropriate Model Application.

8.1.8. Model Application calculates new data value.

8.1.9. Model Application returns updated data to Analysis Application.

8.1.10. Analysis Application returns updated data to Driver Application.

8.1.11. Driver Application sends message when analysis is complete.

8.1.12. Driver Application sends analysis results to the Analysis Application.

8.1.13. Analysis Application stores results in analysis document.

8.1.14. Analysis Application returns results to User Application.

8.1.15. User Application displays results (10).

8.2. User cancels analysis.

9. **Run Analysis**

9.1. User halts analysis.

9.1.1. Analysis Application interrupts analysis when it can be done safely.

9.1.2. Analysis Application stores last valid intermediate results in analysis document.

9.1.3. Analysis Application returns last valid intermediate results to User Application.

9.1.4. User Application displays last valid intermediate results (10.6).

9.2. User runs analysis in background. While jobs are in the background:

9.2.1. User queries Analysis Application for information on running jobs.

9.2.2. User queries Analysis Application for job status.

9.2.3. Analysis Application sends E-mail to user when analysis is complete.

9.3. Analysis Application pauses at checkpoint.

9.3.1. Analysis Application stores intermediate results in analysis document.

9.3.2. Analysis Application returns intermediate results to User Application.

9.3.3. User Application displays intermediate results (10.6).

9.3.4. User edits input values for upcoming models.

9.3.5. User continues analysis.

9.4. User waits for analysis to complete.

9.4.1. Analysis Application sends periodic messages detailing analysis progress to User Application.

9.4.2. User Application displays periodic messages detailing analysis progress.

10. **Presentation of Results**

10.1. User Application displays raw output files from model configurations in format native to model configuration.

10.2. User Application displays results as a list of outputs from all model configurations in a format similar to the input order created in analysis setup.

10.3. User Application displays history data from analysis.

10.4. User Application displays convergence history (from optimization drivers).

10.5. User Application displays output tables (from table generator drivers).

10.6. User Application displays intermediate results.

10.7. User Application displays results as a graph.

10.8. User compares results from separate analysis documents.

11. **File Management**

11.1. User saves analysis template to local storage.

11.2. User saves analysis document to local storage.

11.3. User saves analysis results to local storage.

11.4. User saves sections of output as text to local storage.

DSSA Sub-stage 4-3: Specific User Scenario

This section describes the process a user undertakes in performing some analysis. We selected steps from the overall user scenario that would accomplish the chosen analysis.

SCENARIO DESCRIPTION AND ASSUMPTIONS

We made the following assumptions for this scenario:

1. The user application is loaded on the user's system.
2. The user is an authorized user.
3. The user is an analyst.
4. The analyst is creating a new analysis based on an existing template.
5. The analysis uses the nominal driver.
6. The analyst chooses to view model configurations by name.
7. The analyst adds one model configuration that is not in the template to the analysis.
8. No errors occur.

Table 9. Selected Steps for the Specific User Scenario

Step Number	Description
1.1	User Application displays Login Window.
1.2	User enters user name and password in User Application.
1.3	System validates user name with Security Services.
1.4	Security Services authorizes or denies access to the system according to authentication scheme and returns authorization or denial message to User Application (Security Services).
1.4.1	Security Services authorizes access.
1.5	Security Services determines security privileges of user.
1.6	Security Services authorizes or denies entry into EA (User Application) or System Management Utility (System Administration Application).
1.6.1	Security Services authorizes access.
2.1	User Application displays Analysis Interface.
2.2	User enters Analysis User Application.
2.2.1	User requests list of analysis templates.
2.2.1.1	User Application requests template list from Catalog Services (template repository).
2.2.1.2	Catalog Services retrieves template list, returns it to User Application.
2.2.1.3	User Application formats list of templates.
2.2.1.4	User Application displays list of templates.
2.2.5.2	User creates an analysis from an analysis template. (3.2)
3.2	User selects analysis template for new analysis.
3.2.1	User Application graphically highlights selected analysis template.
3.2.2	User requests new analysis from an analysis template.
3.2.2.2	User Application requests predefined analysis template from Catalog Services (template repository).
3.2.3	Catalog Services returns analysis template to User Application.
3.2.4	User Application initializes new analysis document from analysis template.
3.3	User Application displays analysis document in analysis workspace.
4.1	User requests list of model configurations.
4.1.1	User Application requests model configuration list from Catalog Services (catalog repository).
4.1.2	Catalog Services retrieves model configuration list and returns it to User Application.
4.1.3	User Application organizes list of model configurations by chosen format.
4.1.3.3	User Application organizes list of model configurations by Name.
4.1.4	User Application displays list of model configurations.
4.2	User selects a model configuration for action.
4.2.1	User Application graphically highlights selected model configuration.
4.2.2	User requests basic information for model configuration.
4.2.2.1	User Application displays basic information for model configuration.

Table 9. Selected Steps for the Specific User Scenario

Step Number	Description
4.2.2.1.1	User Application displays model configuration name.
4.2.2.1.2	User Application displays model configuration description.
4.2.3	User requests detailed information for model configuration.
4.2.3.1	User Application displays detailed information for model configuration.
4.2.3.1.1	User Application displays inputs.
4.2.3.1.2	User Application displays outputs.
4.2.3.1.3	User Application displays model configuration options.
4.2.3.1.4	User Application displays model function.
4.2.4	User adds model configuration to analysis document.
4.2.4.1	User Application creates default dependencies for new model configuration.
4.2.4.2	User Application graphically displays new model configuration as part of analysis.
5.2.2	User breaks dependency links.
5.2.2.1	User Application graphically displays broken dependency.
5.2.2.2	User edits input value for broken dependency in User Application.
6.1	User selects class of driver.
6.1.1	User selects Nominal driver (default).
6.2	User associates driver with applicable model configurations.
6.3.1	User identifies parameters of Nominal driver.
7.1.1	User edits values of non-dependent input data elements to model configuration.
8.1	User submits analysis.
9.4	User waits for analysis to complete.
8.1.1	User Application updates and saves analysis document information.
8.1.2	User Application requests that Analysis Application run analysis using current analysis document.
8.1.3	Analysis Application interprets analysis specification.
8.1.4	Analysis Application chooses appropriate Driver Application on the basis of analysis specification.
8.1.5	Analysis Application configures Driver Application for analysis.
9.4.1	Analysis Application sends periodic messages detailing analysis progress to User Application.
9.4.2	User Application displays periodic messages detailing analysis progress.
8.1.6	Driver Application requests data update from Analysis Application.
8.1.7	Analysis Application requests data value update from appropriate Model Application.
8.1.8	Model Application calculates new data value.
8.1.9	Model Application returns updated data to Analysis Application.
8.1.10	Analysis Application returns updated data to Driver Application.
8.1.11	Driver Application sends message when analysis is complete.

Table 9. Selected Steps for the Specific User Scenario

Step Number	Description
8.1.12	Driver Application sends analysis results to the Analysis Application.
8.1.13	Analysis Application stores results in analysis document.
8.1.14	Analysis Application returns results to User Application.
8.1.15	User Application displays results (10).
10.2	User Application displays results as a list of outputs from all model configurations in a format similar to the input order created in analysis setup.
11.3	User saves analysis results to local storage.
2.4	User logs out of EA.
2.4.1	User Application disconnects from Analysis Application.
2.4.2	Analysis Application deactivates user privileges.
2.4.3	User Application displays Login Window.

DSSA Sub-stage 4-4: Dynamic Model

The dynamic model describes the aspects of a system that change. We use the dynamic model to specify the control aspects of a system. The dynamic model includes an event trace diagram and state diagrams.

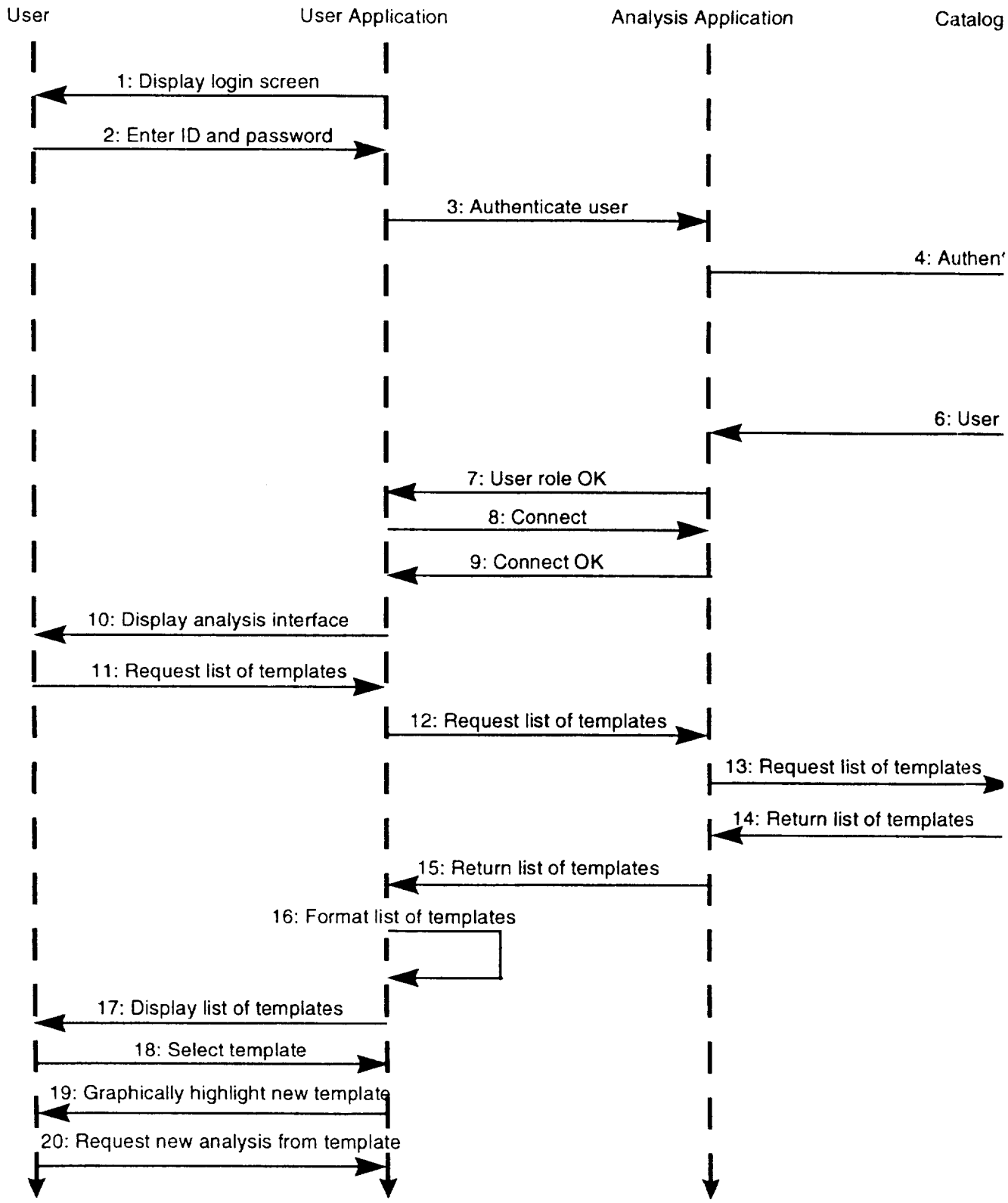
EVENT TRACE DIAGRAM

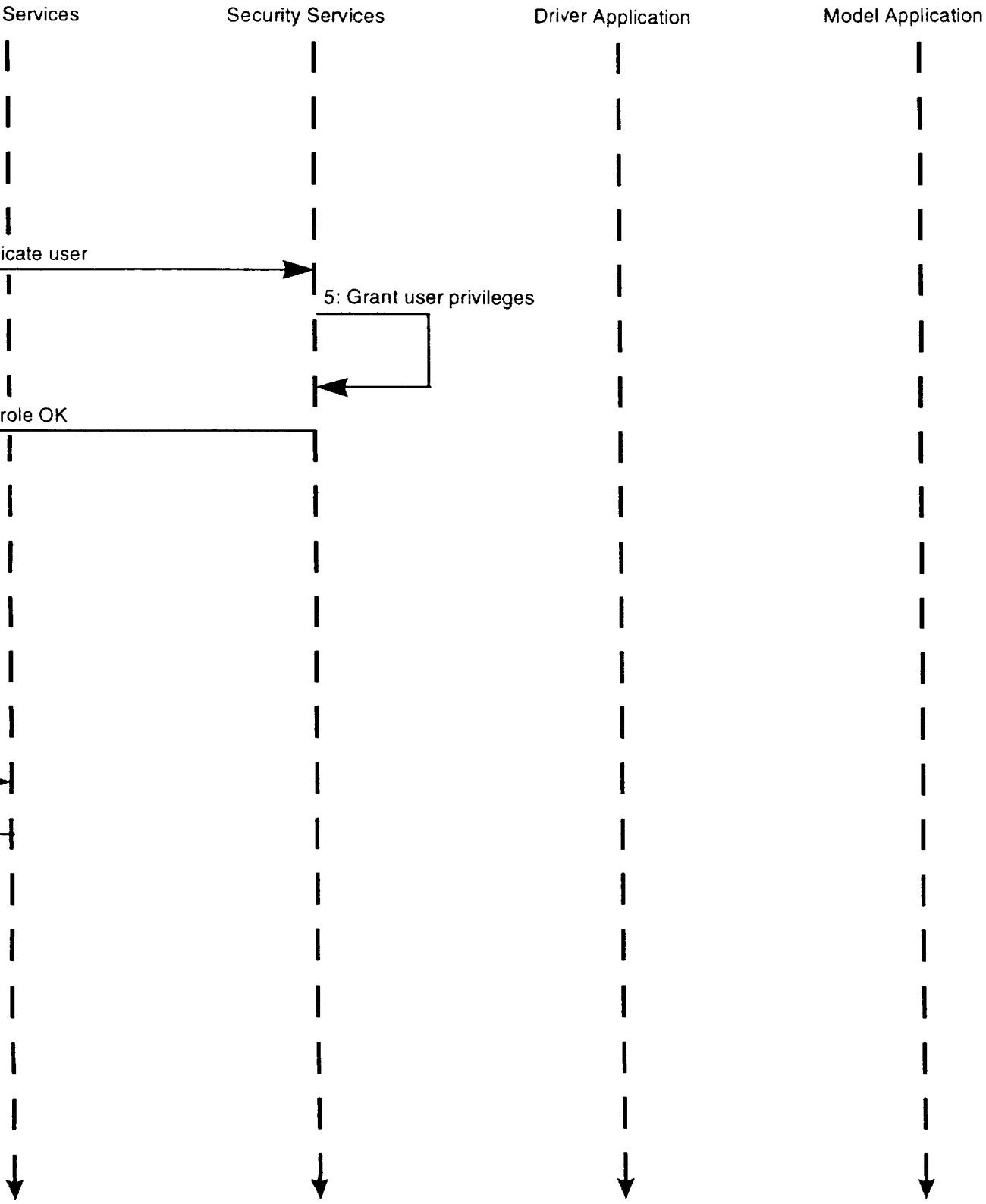
We have created an event trace diagram for our sample user scenario. The event trace diagram depicts a sequence of events and the objects exchanging the events. We chose seven objects for representation in this diagram:

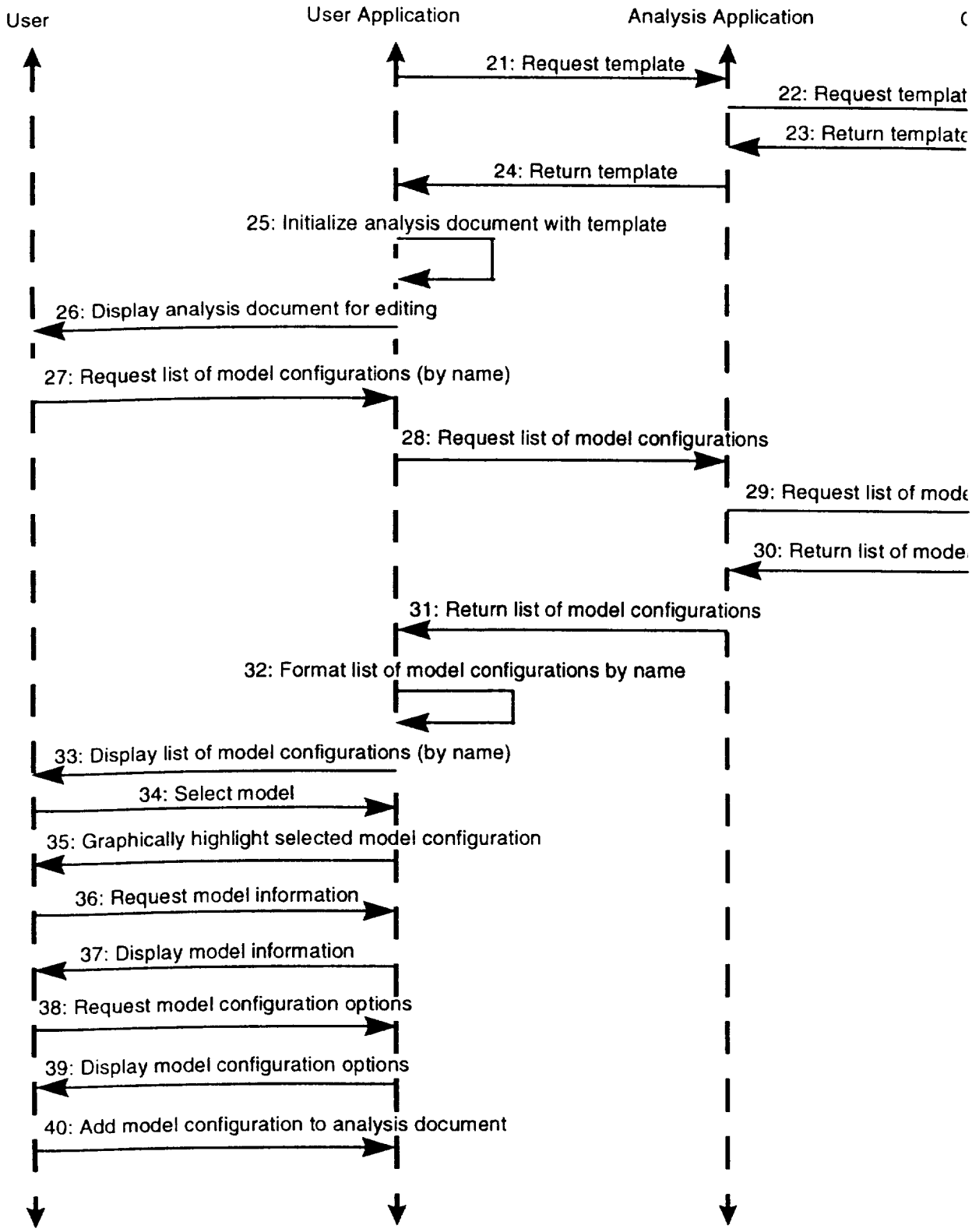
- User
- User application
- Analysis application
- Catalog services
- Security services
- Driver application
- Model application.

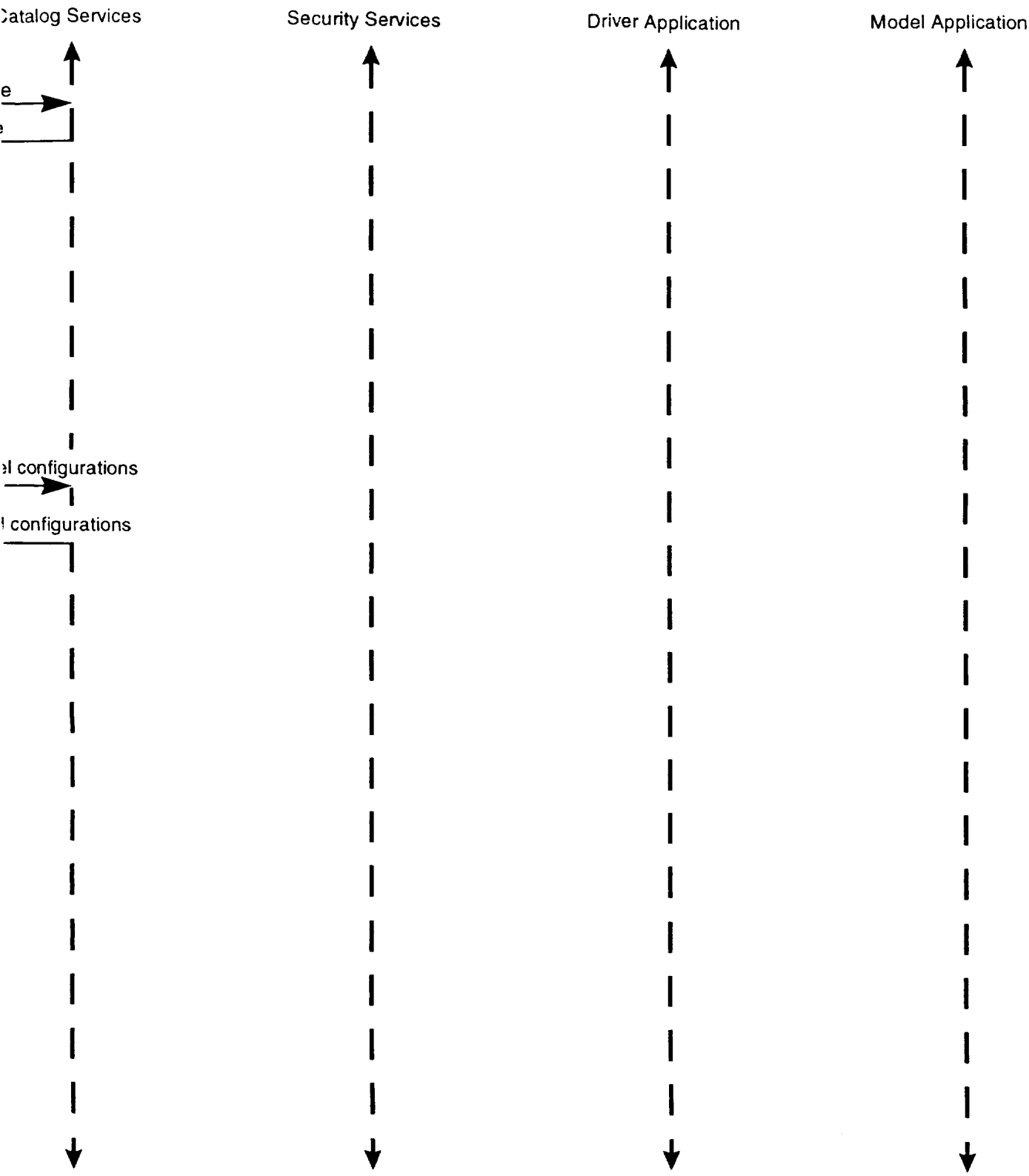
The event trace diagram shows each object as a vertical line and each event as a horizontal arrow from the sender to the receiver object.

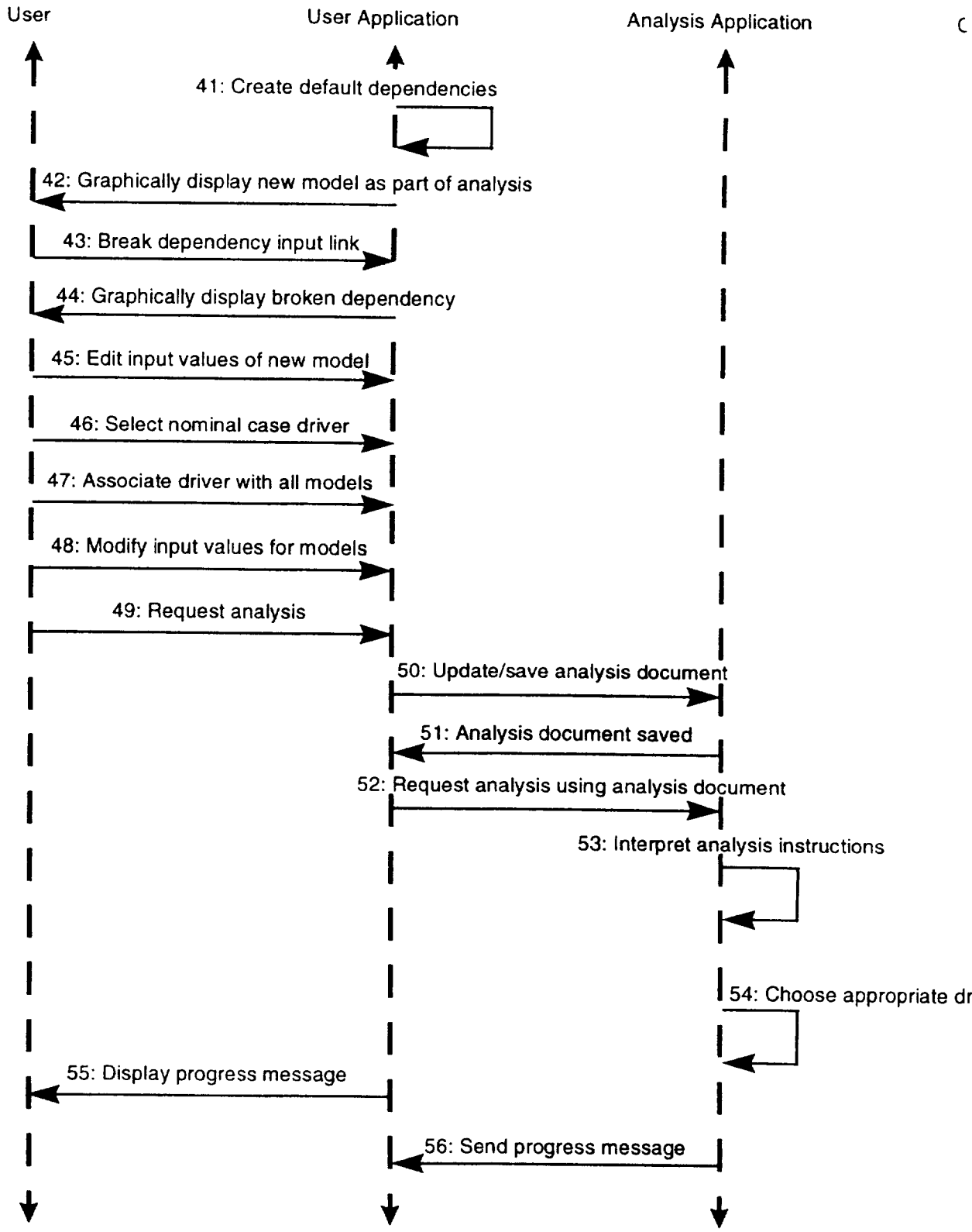
Figure 31.

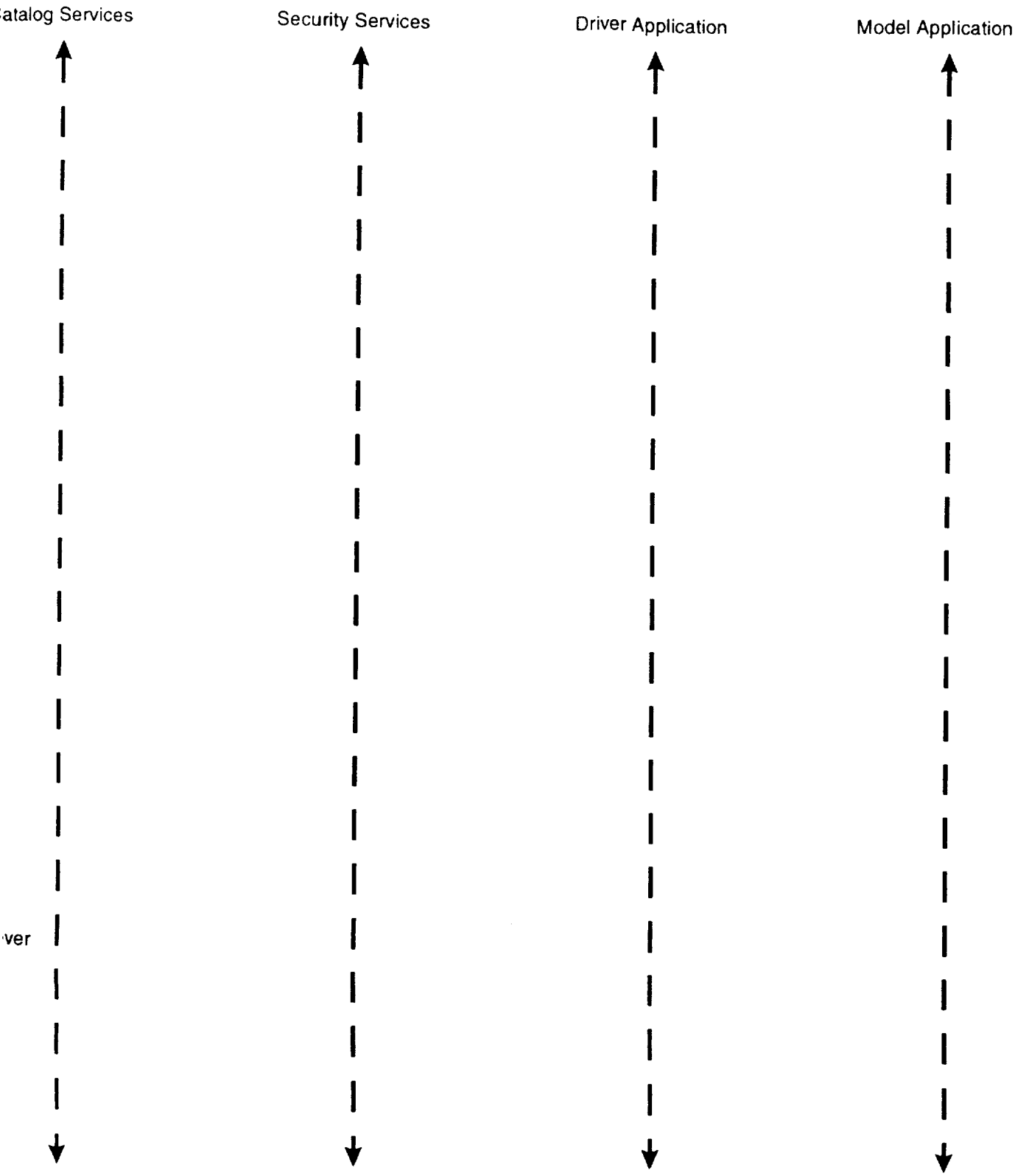

User
User Application
Analysis Application
Catalog
1: Display login screen
2: Enter ID and password
3: Authenticate user
4: Authen'
6: User
7: User role OK
8: Connect
9: Connect OK
10: Display analysis interface
11: Request list of templates
12: Request list of templates
13: Request list of templates
14: Return list of templates
15: Return list of templates
16: Format list of templates
17: Display list of templates
18: Select template
19: Graphically highlight new template
20: Request new analysis from template

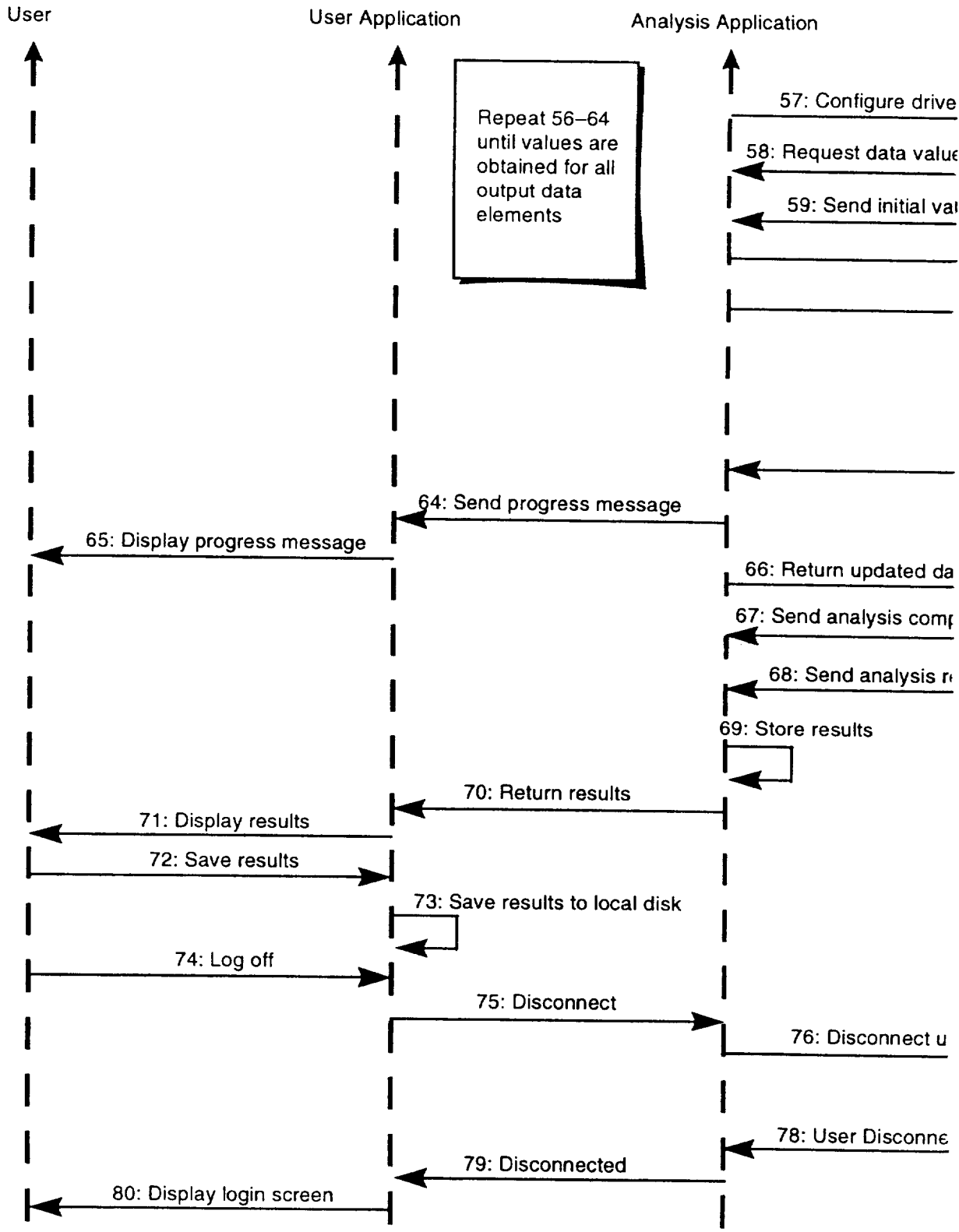

Services
Security Services
Driver Application
Model Application
icate user
5: Grant user privileges
role OK

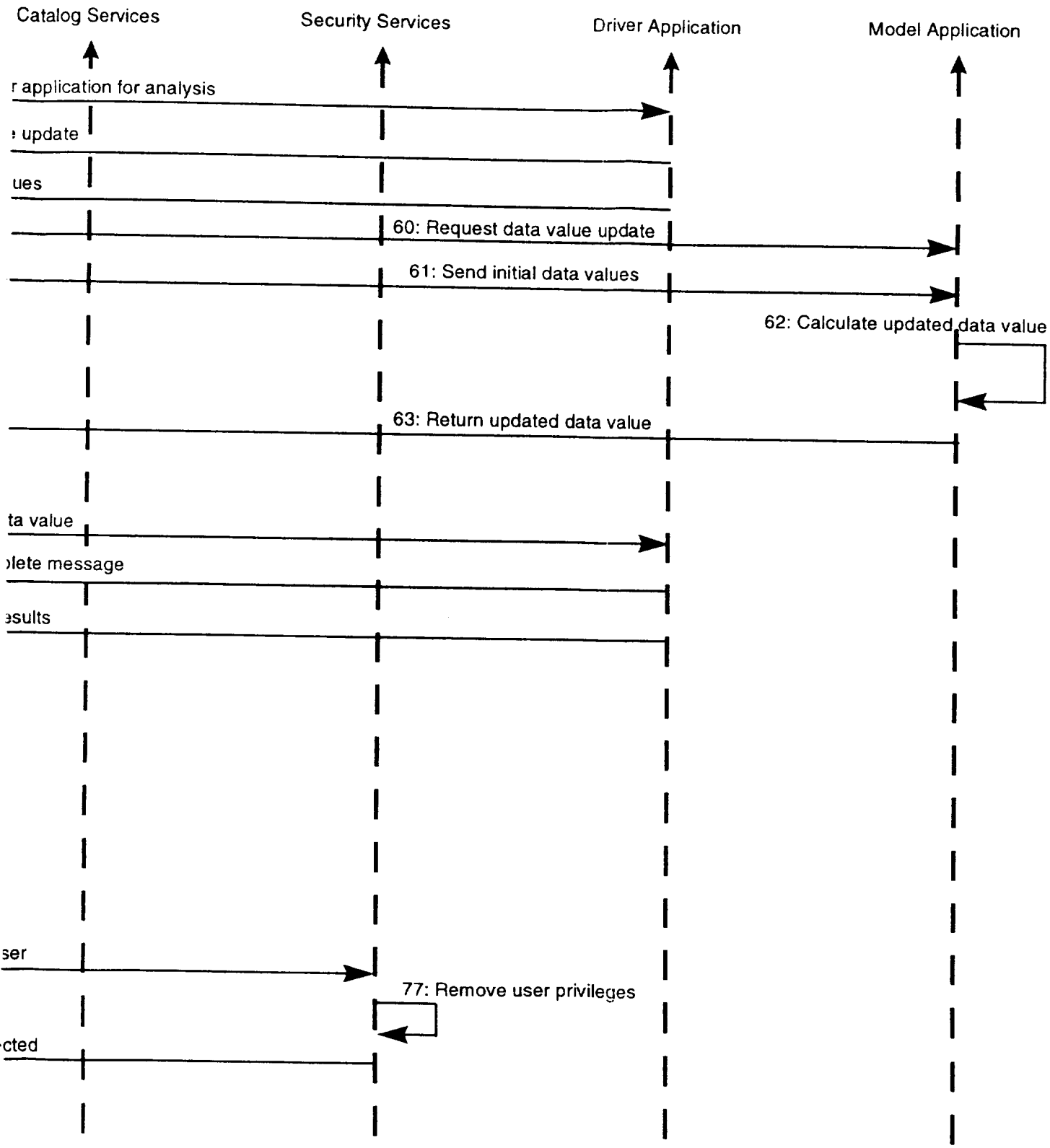

User
User Application
Analysis Application
21: Request template
22: Request templat
23: Return template
24: Return template
25: Initialize analysis document with template
26: Display analysis document for editing
27: Request list of model configurations (by name)
28: Request list of model configurations
29: Request list of mode
30: Return list of mode
31: Return list of model configurations
32: Format list of model configurations by name
33: Display list of model configurations (by name)
34: Select model
35: Graphically highlight selected model configuration
36: Request model information
37: Display model information
38: Request model configuration options
39: Display model configuration options
40: Add model configuration to analysis document


Catalog Services
Security Services
Driver Application
Model Application
e
e
el configurations
l configurations


User
User Application
Analysis Application
41: Create default dependencies
42: Graphically display new model as part of analysis
43: Break dependency input link
44: Graphically display broken dependency
45: Edit input values of new model
46: Select nominal case driver
47: Associate driver with all models
48: Modify input values for models
49: Request analysis
50: Update/save analysis document
51: Analysis document saved
52: Request analysis using analysis document
53: Interpret analysis instructions
54: Choose appropriate dr
55: Display progress message
56: Send progress message


atalog Services
Security Services
Driver Application
Model Application
ver


User
User Application
Analysis Application
Repeat 56–64 until values are obtained for all output data elements
57: Configure drive
58: Request data value
59: Send initial va
64: Send progress message
65: Display progress message
66: Return updated da
67: Send analysis comp
68: Send analysis r
69: Store results
70: Return results
71: Display results
72: Save results
73: Save results to local disk
74: Log off
75: Disconnect
76: Disconnect u
78: User Disconne
79: Disconnected
80: Display login screen


Catalog Services
Security Services
Driver Application
Model Application
r application for analysis
update
ues
60: Request data value update
61: Send initial data values
62: Calculate updated data value
63: Return updated data value
ta value
lete message
esults
ser
77: Remove user privileges
cted

STATE DIAGRAM

From the above event trace diagram, we can create a state diagram. A state diagram is a graph in which nodes are states and arcs are transitions among states caused by events. Six state diagrams, one for each of the objects in the event trace diagram, except user, are represented. It does not make sense to create a state diagram for a user.

In a state diagram, a state is drawn as a rounded box with the state name contained in the box. A transition is drawn as an arrow from the receiving state to the target state; the label on the arrow is the name of the event causing the transition. A solid circle represents the initial state; a bull's-eye the final state.

Figure 32. User Application State Diagram

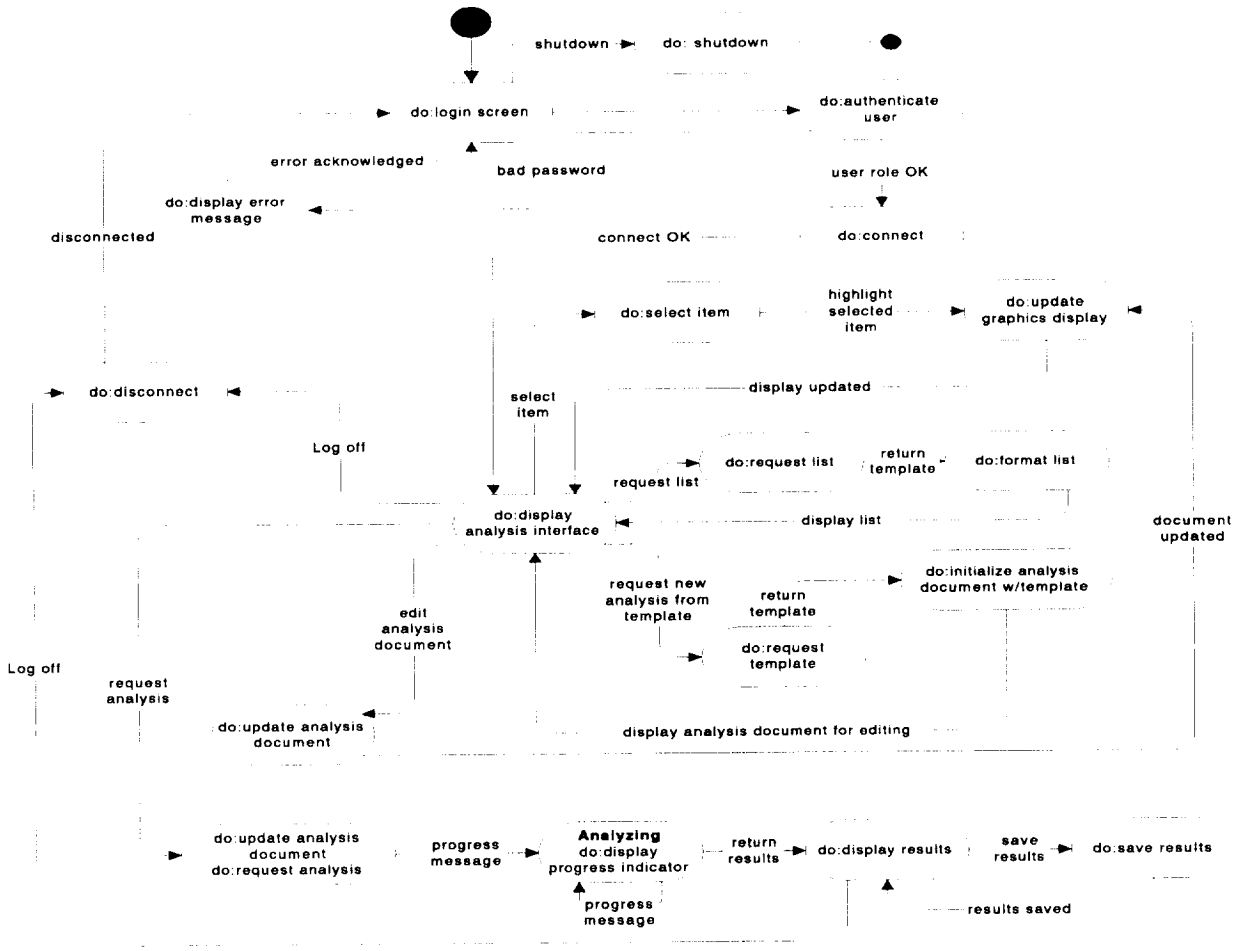


Figure 33. Analysis Application State Diagram

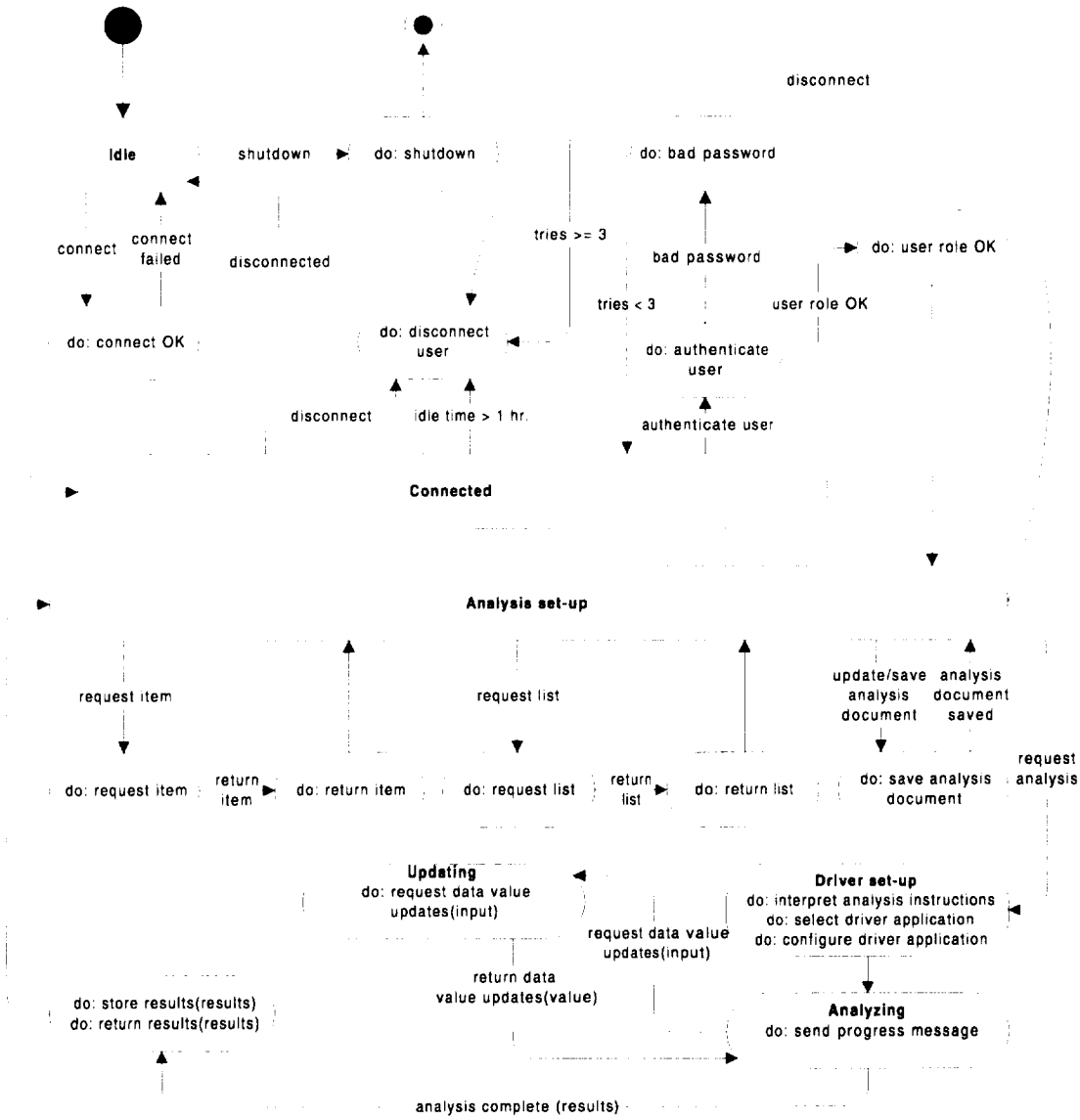


Figure 34. Catalog Services State Diagram

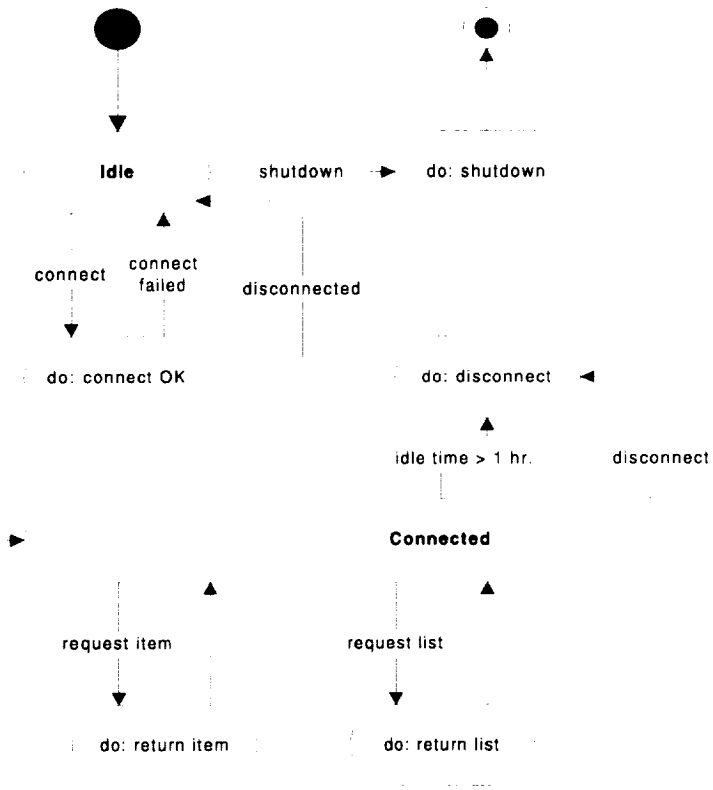


Figure 35. Security Services State Diagram

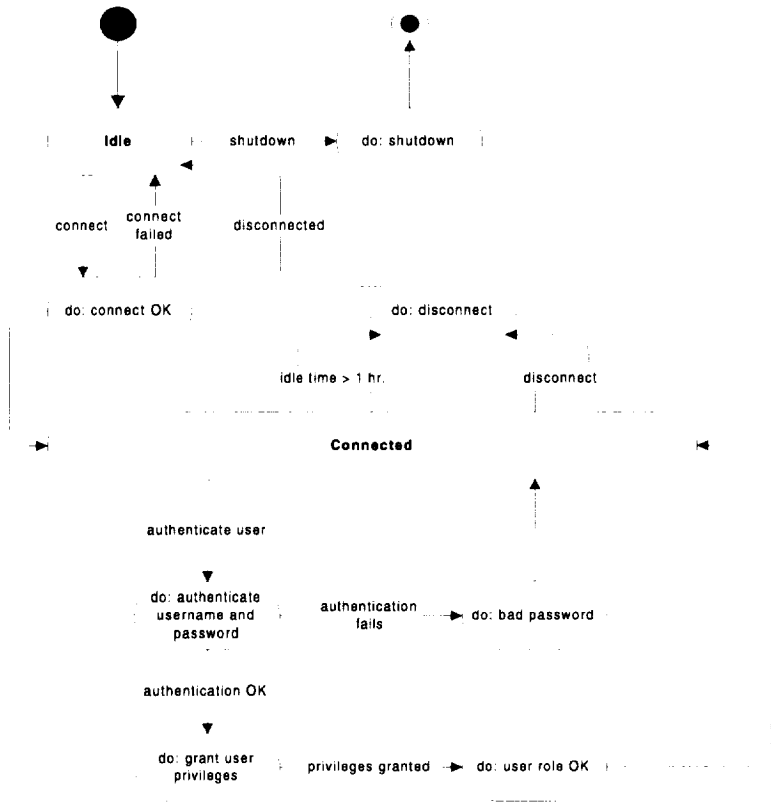


Figure 36. Driver Application State Diagram

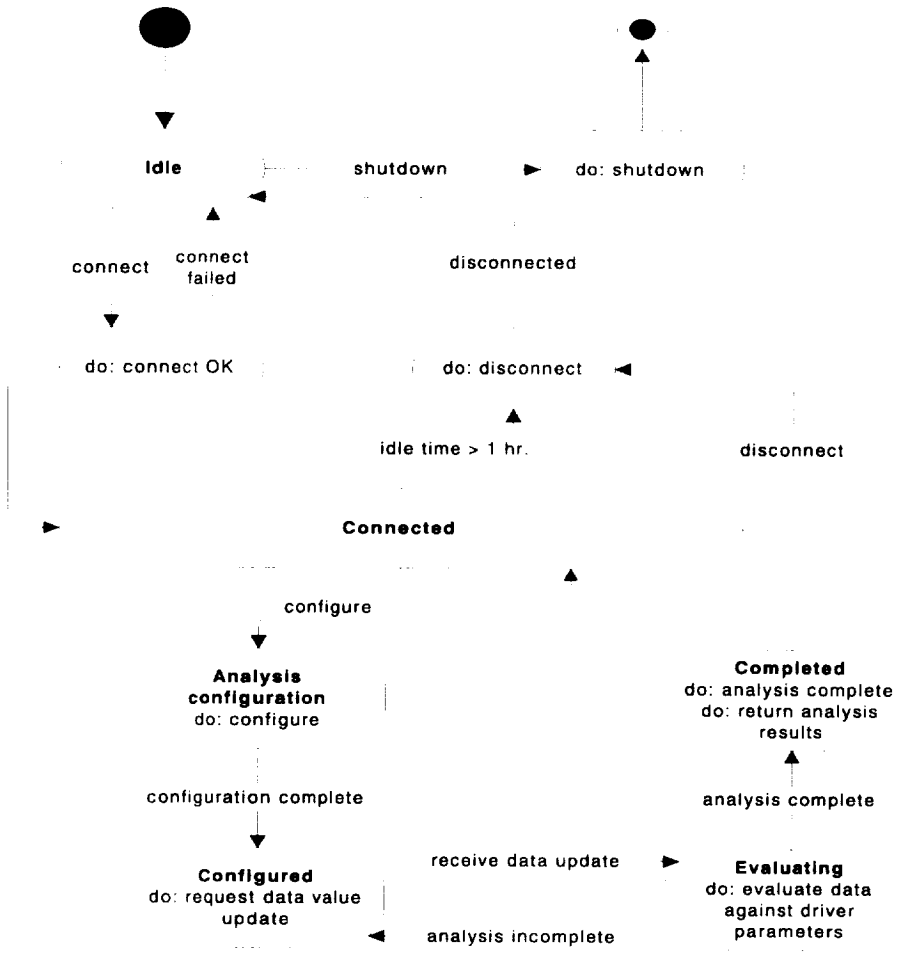
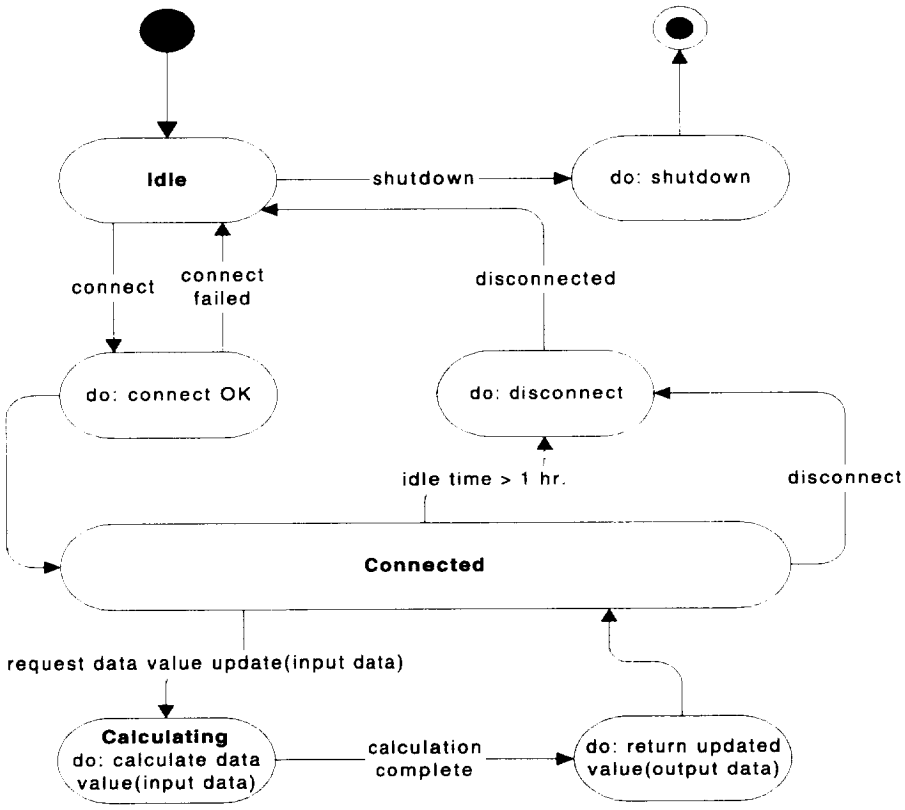


Figure 37. Model Application State Diagram



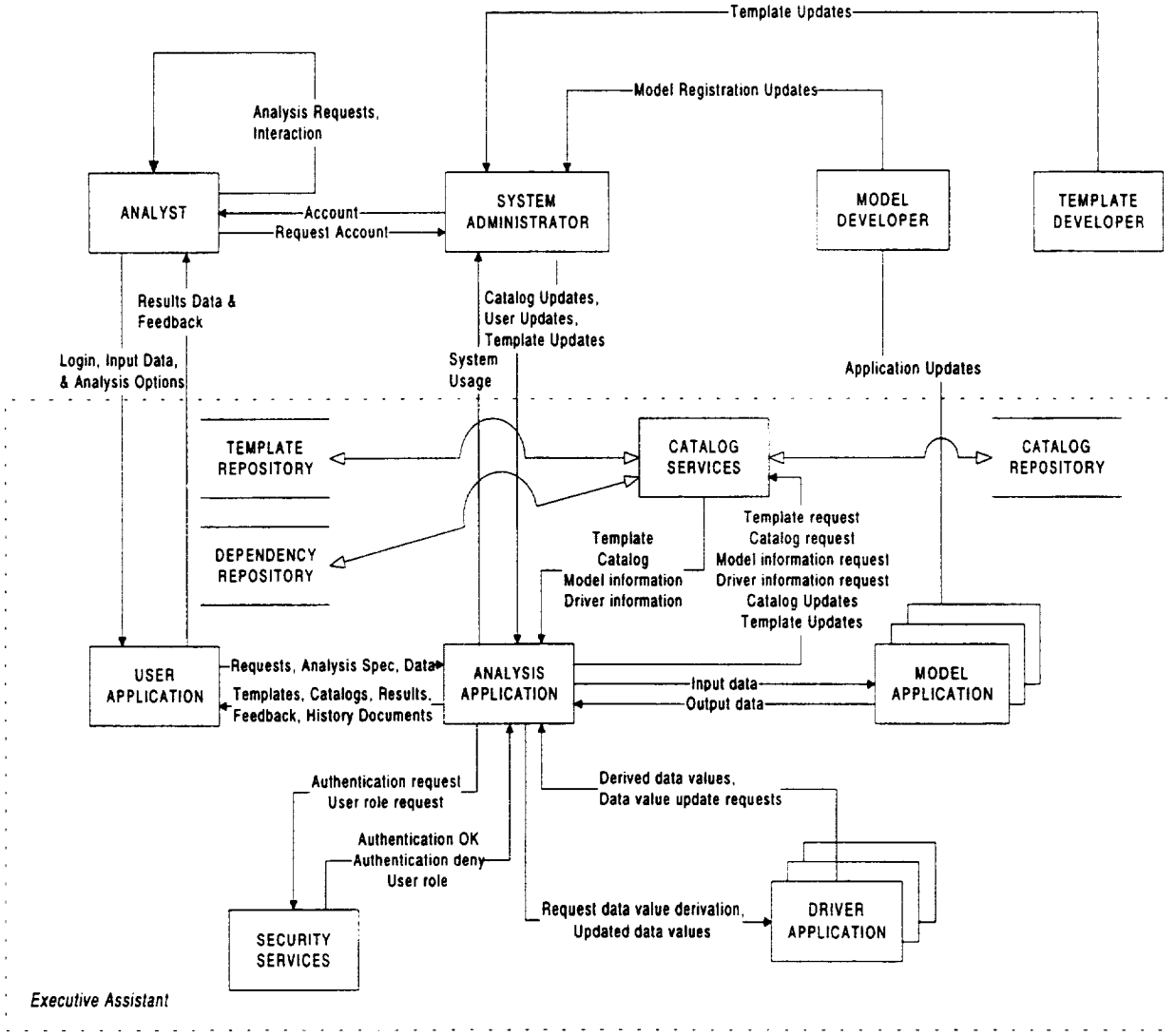
DSSA Sub-stage 4-5: Functional Model

The functional model describes the data value transformations within a system. The functional model contains data flow diagrams.

DATA FLOW DIAGRAM

Next, we created a top level data flow diagram for our sample user scenario. A data flow diagram is a graph in which nodes are processes and arcs are data flows.

Figure 38. Top Level Data Flow Diagram



DSSA Sub-stage 4-6: Requirements Traceability

As a final step, we reviewed requirements to ensure that this architecture meets them.

Table 10. Requirements Traceability

Requirement	How Met
General and EA	
1. Implementation of ASAC will remain flexible as analyses are performed in response to the changing need of its users (users can substitute models and data).	Analysis User Application
2. Implementation of ASAC will allow the addition of new models and tools to the system.	System Administration Service Catalog Service Directory Service Analysis (Broker) Service
3. ASAC will use industry standards such as HTML 2.0.	Addressed in Define Constraints on the Architecture (DSSA Sub-stage 3.1)
4. ASAC development will be under configuration control.	Will be met in development phase
5. ASAC will support users using Windows, Macintosh, HP-UX, SunOS, and SGI IRIX.	Analysis User Application User Interface Service Addressed in Define Constraints on the Architecture (DSSA Sub-stage 3.1)
6. ASAC results will be consistent and reliable.	Use of DSSA methodology
EA	
1. With the EA, users will have guided access (with uniform look and feel) to information now widely scattered and in widely varying formats.	Analysis User Application User Interface Service Directory Service Analysis (Broker) Service Catalog Service
2. EA will accommodate operation of models and databases at remote sites.	Directory Service Analysis (Broker) Service Communication Service
3. With the EA, the user will be able to evaluate a class of predefined representative problems.	Analysis Template
4. With EA, the user will be able to view and modify data at intermediate steps (visually inspect and change data being transferred among models).	Checkpoints Analysis User Application User Interface Service
5. EA will provide an audit trail showing models used, and input and output files.	Audit Service
6. EA will run analyses comprised of one or more models.	Directory Service Analysis (Broker) Service Driver
7. EA model inputs will be predefined.	Catalog Service
8. Users will be able to change EA model input values.	Analysis User Application User Interface Service

Table 10. Requirements Traceability (Con't.)

Requirement	How Met
9. EA models will allow for multiple inputs.	Catalog Service
10. The EA will determine the number of times the model needs to be run.	Driver
11. Invalid EA model inputs for a specific analysis will not be available.	Catalog Service
12. EA model inputs may be a file.	Analysis User Application
13. Users will be able to view EA model inputs in raw and converted format.	Analysis User Application User Interface Service Analysis Document
14. EA analysis will stop at selected intermediate steps.	Checkpoints
15. With the EA, users will be able to select the model inputs and outputs to view.	Analysis User Application User Interface Service
16. With the EA, users will be able to save viewed data.	Analysis User Application File Management Service
17. The EA will mail a notification of completion to the user.	Communication Management Service
18. In an information message, the EA will indicate an estimate of the run time for selected models.	Alert Notification Service Catalog Service
19. With the EA, users will be able to cancel analysis at all intermediate steps.	Checkpoints Analysis User Application User Interface Service
20. The EA will provide a general optimizing tool.	Driver
21. For each analysis, the EA will create an analysis document that includes elements needed to recreate the analysis.	Analysis Document
Functional Requirements	
1. The EA will integrate results of models run outside the EA into an analysis document with the system	Analysis User Application User Interface Service Catalog Service Analysis Document
2. The EA will include a mechanism for evaluating and choosing from future shadow model sites.	Directory Service
3. When the Analysis System fails, the User Interface will detect the failure or time out.	Communication Service Alert Notification Service Directory Service Analysis (Broker) Service
4. The EA will provide valid default data values to models upon initialization.	Catalog Service
5. Given a new input to a model, the system will minimize the number of models that need to be rerun. For example, given five models A, B, C, D, and E, where A = f(B+C) and B = f(D+E), when a value of E is changed, only A and B will be rerun (C and D will not be rerun).	Analysis Application (Inference Engine)
Non-Functional Requirements	
Performance	
1. Barring system failure, models run through the ASAC system will execute to completion in less than 24 hours.	System Administrator Function Catalog Service
2. User interface response time will be in accordance with best commercial practices.	User Interface Service

Table 10. Requirements Traceability (Con't.)

Requirement	How Met
3. EA performance time will be in accordance with best commercial practices.	Analysis (Broker) Service Directory Service Load Balancing Service Addressed in Define Constraints on the Architecture (DSSA Sub-stage 3.1)
Security	
1. User authentication will control access to the EA system.	Security Service
2. User authentication will control access to stored analyses.	Security Service
3. User authentication will control access to models.	Security Service
Extendability	
1. The EA system will support the addition of models and drivers.	Catalog Service Directory Service Analysis (Broker) Service
Scalability	
1. The EA will support vertical and horizontal growth.	Directory Service Analysis (Broker) Service
2. The EA will support replication of model applications.	Directory Service
3. The EA will support load balancing among replicated models.	Load Balancing Service
Fault Tolerance	
1. The EA will continue to provide standard services if a model application fails or crashes.	Distributed Computing Service Application Management Service
2. The EA will continue to provide standard services if a client application fails or crashes.	Distributed Computing Service Application Management Service
Testability	
1. The EA will provide a log of system use.	Audit Service
2. The EA will have a built-in debug mode.	Will be met in design phase
Portability	
1. User applications will be portable across Personal Computer, Macintosh, and UNIX platforms.	Addressed in Define Constraints on the Architecture (DSSA Sub-stage 3.1)
2. Database connectivity will be open, i.e., not restricted to a particular database vendor.	Addressed in Assumptions (DSSA Sub-stage 2.8)
Reusability	
1. The interface between the analysis application and models will have reusable components.	Will be met in design phase
2. EA interfaces to models will be modular so that they can be reused.	Will be met in design phase
Usability	
1. The user application will have an intuitive graphical user interface that adheres to the IBM CUA standards and, therefore, minimizes the need for user training.	User Interface Service Analysis User Application
2. The user application user interface will tolerate user errors.	User Interface Service Analysis User Application
3. The user application user interface will help the user follow the proper analysis process.	User Interface Service Analysis User Application
4. The help service will help with the user application.	Help Service

Table 10. Requirements Traceability (Con't.)

Requirement	How Met
Interoperability	
1. The user application will be interoperable with WWW browsers supporting Java 2.0 and HTML 3.2.	Use Application Service Analysis User Application

DSSA Sub-stage 4-7: Review and Iterate

Review and iterate the items developed in DSSA Stage 4.

CONCLUSION

We used published and respected methodologies to analyze the requirements and define an architecture for the ASAC EA system. The architecture definition includes an object model composed of object diagrams, a dynamic model composed of an event trace diagram and state diagrams, and a functional model composed of a data flow diagram.

We received endorsement of our architecture from one of the principal investigators on the DARPA DSSA program, Dr. Will Tracz of Lockheed-Martin Corporation, Owego, New York.

Work will begin on the next phase, design and development of the ASAC EA, in fiscal year 1997.

Bibliography

Amako, Katsuya. "Object Modeling Techniques—Summary Note", (summary of J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen "Object-Oriented Modeling and Design", Prentice Hall International Editions, ISBN 0-13-630054-5), http://arkhp.kek.jp/managers/computing/activities/OO_CollectInfor/Methodologies/OMT/OMTBook/OMTBook.html.

ARPA Tech '96, "The Big Picture," http://www.arpa.mil/ARPATech-96/overview.html, March 1996.

Assaf, A., M. Brewer, and S. Maheshwari. "CS837 Project, Client/Server Architecture, Implementation using BlackWidow, CORBA, Java," Spring 1996. http://www.cs.uga.edu/~brewer/BW/handout.html

Assaf, A., M. Brewer, and S. Maheshwari. "Multi-Tiered Architectures, CS837," Spring 1996. http://www.cs.uga.edu/~brewer/BW/paper.html

Assaf, A., M. Brewer, and S. Maheshwari. "Presentation Issues," Spring 1996. http://www.cs.uga.edu/~brewer/BW/presentissues.html

Assaf, A., M. Brewer, and S. Maheshwari. "Questions to be posed to a client/server solution," Spring 1996. http://www.cs.uga.edu/~brewer/BW/questions.html

Avellino, Ken. "Introduction to Client/Server Computing," Learning Tree International, 369/CN(2)/D.2/601/D.1, 1996.

Bakker, Gerbrand. "OMT Object Model," University of Twente, Department of Computer Science Educational Network, February 1996, http://wwwedu.cs.utwente.nl/~misop007

Cheury, Derek. "ATM Software Analysis and Design." *Dr. Dobb's Journal*, October 1996.

Defense Information Systems Agency. "Defense Information Infrastructure (DII) Common Operating Environment (COE) I&RTS: Rev 2.0, October 1995."

Defense Information Systems Agency. "Defense Information Infrastructure (DII) Common Operating Environment (COE), Version 2.0 (Series), Baseline Specifications, June 1996."

Domain Specific Software Architectures (DSSA), http://www.sei.cmu.edu/arpa/evo/dssa-sum.html

Evolutionary Software Development, http://www.sei.cmu.edu/arpa/evo.html

Hurwitz, J. "The Movable Enterprise, A new architecture to meet the changing needs of the virtual corporation." *DBMS*, August 1996.

JSIMS Domain Definition, http://www.jsims.mil/dom_eng/domain.html

Kaneshige, T. "Migrating to Multitiering." *Client/Server Computing*, June 1996.

Lockheed Martin Advanced Concepts Center and Rational Software Corporation. "Succeeding with the Booch and OMT Methods, A Practical Approach,", Addison Wesley, 1996.

Malone, B., and S. Woyak. "An Object-Oriented Analysis and Optimization Control Environment for the Conceptual Design of Aircraft." 1st AIAA Aircraft Engineering, Technology, and Operations Congress, September, 1995.

Meta Group, Software AG, "Internet, Networking, and Middleware Glossary," 1996.

Might, R. J. "Domain Models, What are They? How are They Used." George Mason University, The Institute of Public Policy. December 1994.

National Institute of Standards and Technology. "Application Portability Profile (APP) The U.S. Government's Open System Environment Profile Version 3.0 (Supersedes NIST SP 500-210." June 1996.

Open Enterprise Group of IBM Europe. "A Guide to Open Client/Server." May 1994.

Rumbaugh, James, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen "Object-Oriented Modeling and Design," Prentice Hall, 1991

Schussel, G. "Client/Server: Past, Present and Future." June 1996. http://www.dciexpo.com/geos/dbsejava.htm.

Schussel, G. "Client/Server, from dBASE to JAVA: The Next Step." June 1996. http://www.dciexpo.com/geos/javastep.htm.

Section 3, GCCS COE Architecture, http://164.117.208.50/coe/coe3.html, June 1995.

Section 4, GCCS Architecture Guidelines, http://164.117.208.50/coe/coe4.html, June 1995.

Tibbets, and Bernstein. "Sharpen Those Definitions – Confusion results whenever people misuse terms such as 'client-server' and 'three-tier'." Information Week, January 9, 1997.

Tracz, W. "Domain-Specific Software Architecture (DSSA) Frequently Asked Questions (FAQ), Version 1.2 - ADAGE-IBM-93-12B." Loral Federal Systems, Owego, March 1995. http://www.sei.cmu.edu/arpa/dssa/dssa-adage/faq/faq.html

Tracz, W. "What is DSSA?" Loral Federal Systems, Owego. http://www.owego.com/dssa/what-is-dssa.html

Tracz, W. and L. Coglianese. "Domain-Specific Software Architecture Engineering Process Guidelines, ADAGE-IBM-92-02B." Loral Federal Systems, Owego, 1992.

Wingrove, E. R. III, P. F. Kostiuk, R.C. Sickles, and D. H. Good. "The ASAC Air Carrier Investment Model (Revised), NS301RD2." Logistics Management Institute, June 1996.

Woyak, S. A., B. Malone, and A. Myklebust. "An Architecture for Creating Engineering Applications: The Dynamic Integration System." ASME Computers in Engineering, September 1995.

Appendix A
Acronyms

ACSYNT	Aircraft Synthesis Model
AIA	Aerospace Industries Association
AND	Approximate Network Delay Model
APP	Application Portability Profile
ASAC	Aviation System Analysis Capability
AST	Advanced Subsonic Technology Program
ATA	Air Transportation Association
ATC	Air Traffic Control
ATM	Automated Teller Machine
CAASD	Center for Advanced Aviation System Development
CGI	Common Gateway Interface
COE	Common Operating Environment
COSTMOD	Delay Cost Model
CUA	Common User Access
DARPA	Defense Advanced Research Projects Agency
DBMS	Database Management System
DOT	Department of Transportation
DSSA	Domain-Specific Software Architecture
EA	Executive Assistant
FAA	Federal Aviation Administration
FLOPS	Flight Optimization System Model
GUI	Graphical User Interface
HP	Hewlett-Packard
HTML	HyperText Markup Language
I/O	Input/Output

JSIMS	U.S. Army's Joint Simulation System
LAN	Local Area Network
NARIM	National Airspace Research and Investment Model
NASA	National Aeronautics and Space Administration
NIST	National Institute of Standards and Technology
OMT	Object Modeling Technique
OSE	Open System Environment
OSF	Open Software Foundation
QRS	Quick Response System
RPC	Remote Procedure Call
SGI	Silicon Graphics, Incorporated
SQL	Structured Query Language
STAT	Small Transportation Aircraft Technology
TBD	To Be Determined
TCP/IP	Transmission Control Protocol/Internet Protocol
WAN	Wide Area Network

Appendix B
Domain Dictionary

Alert Notification Service—a component of the Presentation Service that provides presentation for user- and application-generated alerts and notifications.

Alerts Service—a component of the Common Support Service that provides functionality for user- and application-generated alerts and notifications. It routes and manages alert messages throughout the domain.

Analysis Application—A software component that interacts with user input, models, and drivers to create analytical results.

Analysis Data—the part of an analysis document that contains the input and output data from each stage of an analysis.

Analysis Document—a document that contains the analysis specification and analysis data for a specific analysis; it is an instantiation of an analysis template.

Analysis (Broker) Service—a component of the Distributed Computing Service that allows applications to use methods or objects that are remote. It allows objects to dynamically discover each other and interact across machines, operating systems, and networks.

Analysis Specification—the part of the analysis template or document that provides detailed instructions to the application performing an analysis.

Analysis Template—a generic document that provides an outline for performing specific types of analysis; the user customizes the template for a specific task and saves it as an analysis document.

Analyst—a person who interacts with the Executive Assistant to perform an analysis, define an analysis, save data, halt an analysis, inquire about the state of a running analysis, choose an analysis template, or choose an existing analysis to run.

Application Architecture—the architecture for a single system (the result of instantiating and refining a reference architecture).

Application Engineering—the process of instantiating/refining and/or extending a reference architecture.

Application Management Service—a component of the Management Service that maintains a dynamic configuration of application services. It starts the appropriate application services on appropriate machines. It also monitors application services to ensure they are available and performing correctly, restarts lost services, and starts and stops services.

Audit Service—a component of the Management Service that provides a permanent record of system usage. It includes user access, model usage, error logging, and time stamping.

Authentication—verification of the user's validity.

Authorization—control of user access.

Aviation System Analysis Capability (ASAC)—a decision support system consisting of models, databases, and tools, used to support analysis of the effects of advanced technologies on the integrated aviation system.

Backsolver—a type of driver that finds a value for one or more variables, given a set of constraints.

Catalog Repository—database containing the registrations for all models and drivers within the Executive Assistant.

Catalog Service—a component of the Data Service that registers and assembles model and driver descriptions, configurations, I/O, type, and average execution time requirements.

Checkpoint—a point along an analysis string at which the user chooses to suspend the analysis to examine its current state. Setting checkpoints lets the user stop an analysis run in between running models. At such a point, the user may view data both after it exits a model and before it enters the next (there may be differences because of units translation).

Client—a software program used to contact and obtain data from a server software program.

Common Support Service—a service that provides common system-level support across all components of the EA system.

Communication Management Service—a component of the Communication Service that provides overarching communication services between system applications and other services.

Communication Service—a service that provides communication among all components within the domain. It provides translation if more than one protocol is used. It also provides facilities for receiving data external to the domain and sending data out of the domain.

Data Access and Connectivity Service— a service that provides DBMS connectivity; performs query transactions.

Data Administration Service—a component of the Data Service that allows a System Administrator to enter, maintain, change, and remove data in repositories.

Data Flow Diagram—a graph on which nodes are processes and arcs are data flows; part of a functional model.

Data Interchange Service—a component of the Distributed Computing Service that provides specialized support for applications that may be dispersed among computer systems in the network but must maintain a cooperative processing environment.

Data Management Service—a component of the Data Service that provides DBMS connectivity, and performs query transactions.

Data Service—a service that provides data administration, management, input/output, and distribution services.

Default Data—predefined input values for models.

Default Template—a template that contains no model or data.

Dependency Repository—database containing the dependencies for all models and drivers within the Executive Assistant.

Directory Service—a component of the Distributed Computing Service that maintains a dynamic list of all application services, models, and drivers supported by EA.

Distributed Computing Service—a service that provides specialized support for applications that may be dispersed among computer systems in the network but must maintain a cooperative processing environment.

Domain Engineering—the process of creating a DSSA (Domain Analysis and Domain Modeling followed by creating a software architecture and populating it with components).

Domain Expert—individual whose expertise and experience in the domain can lend insight into various aspects of the domain.

Domain Model—any representation of elements in a domain that shows some relationship among them. In DSSA this model usually consists of a lexicon, ontology, and taxonomy of the terms that characterize the domain, including objects, relationships, products, and perhaps behavioral terms such as actions and events.

Domain-Specific Software Architecture—a software architecture with reference requirements and domain model, infrastructure to support it, and process to instantiate and refine it.

Driver—an interactive problem-solving software application that interfaces with the analysis application; examples include an optimizer, backsolver, and table generator.

Driver Application—an application that handles different types of drivers.

Driver Developer—a person who develops a driver.

Driver-wrapped Model—a model or set of models that a driver uses to perform an analysis.

Dynamic Model—a model that describes the aspects of a system that change over time; used to specify and implement the control aspects of a system.

Error Management Service—a component of the Management Service that provides error management for system level errors. It preserves the integrity of the system.

Event Trace Diagram—a diagram that depicts a sequence of events and the objects exchanging the events; part of a dynamic model.

Executive Assistant—subdomain of ASAC that encompasses applications with which an analyst can use a series of one or more integrated models and drivers to perform an analysis.

Execution Point—a point along an analysis string at which the user chooses to perform an action; examples are a checkpoint, start point, and stop point.

Extendability—the ability to add additional (new) functionality to a system.

File I/O Service—a component of the Data Service that provides file management capability.

Functional Model—a model that describes the data value transformations within a system; contains data flow diagrams.

Global—components that are available to all users.

Help Service—a component of the Common Support Service that provides help to users.

History Document—a document that contains the results of running an analysis using an analysis document.

Horizontal Growth—horizontal expansion of a system, e.g., 1 to 2 to 3 servers.

Intra-application Communication Service—a component of the Communication Service that provides communication capability among components within the EA system.

Load Balancing Service—a component of the Communication Service that provides load balancing among replicated models to ensure no model is overloaded.

Local—components that are local to a user's system; not available to all users.

Login—the account name used to gain access to a computer system.

Log in—to enter a computer system.

Logon Transparency—the use of a single password to gain access to all servers and their services within a system (multiple system elements recognize one log in).

Management Service—a service that provides system, application, error, performance, and security management.

Message Service—a component of the Common Support Service that handles message traffic (parsing and distribution) among applications. It provides message queuing capability.

Model—a standalone software application that provides non-interactive data transformation.

Model Application—an application that handles different types of models.

Model Configuration—an execution path through a model that has unique inputs and outputs; every model has at least one configuration.

Model Developer—a person who develops a model.

Model Option—a mechanism that specifies the configuration of a model.

Network Service—a component of the Communication Service that provides a connection between the external communications network and the EA system.

Nominal Driver—the default driver that does not perform a function, e.g., not an optimizer, backsolver, or table generator.

Object Diagram—a graph on which nodes are object classes and arcs are relationships among classes; part of an object model.

Object Model—a model that describes the static structure of the objects in a system and their relationships; contains object diagrams.

Optimizer—a type of driver that finds a minimum or maximum value for a variable.

Performance Monitoring Service—a component of the Management Service that provides routine diagnostics and performance monitoring.

Presentation Service—a service that provides the user interface layer. Here, the user formulates a request and receives a reply. This service also displays alert information to the user.

Queuing Service—a component of the Communication Service that provides process queuing to ensure fair access to system resources.

Reference Architecture—a software architecture for a family of application systems.

Registration—a description and specification of a model or driver; it is stored in the Catalog Repository and used by the Catalog Service.

Remote Process Service—a component of the Distributed Computing Service that provides remote procedure call (RPC) capability to the system.

Replication—hosting of multiple copies of an application or service on multiple machines to ease network contention and processor load.

Repository—a database that contains the catalog, template, and dependency repositories.

Scalability—the ability to add more of an existing function to a system.

Scheduling Service—provides thread management, queuing, and load balancing.

Security Administration Service—a component of the Management Service that maintains a registry of all authorized users throughout the domain and tracks which functions each user or group is allowed to perform (authorization). It assigns password protection, model security, groups, and permissions.

Security Service—a component of the Management Service that provides a single login service for all systems throughout the domain. It authenticates a user and provides logon transparency.

Server—a software program that provides a specific service to client software. A single server machine can run several different server software programs at the same time.

Software Distribution Service—a component of the Data Service that provides electronic software distribution capability.

Start Point—the point along an analysis string at which a user chooses to begin an analysis.

State Diagram—a graph whose nodes are states and whose arcs are transitions among states caused by events; part of a dynamic model.

Stop Point—the point along an analysis string at which a user chooses to stop an analysis.

System Administration Service—a component of the Management Service that provides the capability to configure, operate, maintain, and manage the local configuration.

System Administrator—person who administers configuration of Executive Assistant applications, adds users, administers passwords, administers security levels, creates and administers model and driver registrations.

System Management Service—a component of the Management Service that provides system management and query governing and handles runaway queries.

Table Generator—a type of driver that populates a table of variables with values.

Template Developer—a person who develops a template.

Template Repository—database containing globally available templates.

Thread—a unit of concurrency provided in a program. They are used to create a programs execution environment that weaves together instructions from multiple independent execution paths or "threads."

Thread Management Service—a component of the Distributed Computing Service that provides thread management capabilities for applicable application services.

Transaction Management Service—a component of the Communication Service that supports creation and management of transaction logs and transaction processing.

User Application—software component with which the user interacts with the Executive Assistant.

User Interface Service—a component of the Presentation Service that provides direct interaction with a user through windows, icons, menus, keyboard, mouse, or other means.

Vertical Growth—vertical expansion, e.g., add additional models or clients to a system.

Appendix C
Proceedings of the ASAC Architecture Meetings, 11–20 March 1996

This section contains the proceeding of the ASAC architecture meetings, held 11-20 March 1996, along with the model information flow diagrams. These meeting were held to define the purposes, goals, requirements, and components of ASAC.

Note: Two model names have changed since the March ASAC architecture proceedings were documented. They are identified in Table C-1.

Table C-1. ASAC Model Name Changes

Old Model Name	New Model Name
2.4 ASAC Preferential Runway Use Model	2.4 ASAC Runway Use Noise Impact Model
3.2.1 ASAC Air Carrier Flight Track Efficiency Model	3.2.1 ASAC Flight Track Noise Impact Model

Introduction:

Meetings were held 11 - 20 March 1996 to set the groundwork for ASAC Architecture development. Topics covered during these meetings included:

- What is ASAC?
- Users
- Goals and Attributes
- Terms
- Tasks
- Requirements
- Models
- Data
- Information Flow – Models
- Analytical Functions
- Analyses
- User Scenario
- First Generation ASAC
- Detailed Information – First Generation Models

Attendees:

Attendees of the 11 - 20 March 1996 ASAC meetings:

Name	Organization	Phone	E-mail
Curia, Marjorie	LMI	703-917-7136	mcuria@lmi.org
Johnson, Jesse	LMI	703-917-7424	jjohnson@lmi.org
Kaplan, Bruce	LMI	703-917-7284	bkaplan@lmi.org
Kostiuk, Pete	LMI	703-917-7427	pkostiuk@lmi.org
Lee, Dave	LMI	703-917-7557	dlee@lmi.org
Malone, Brett	Phoenix Integration	540-231-7215	malone@phoenix-int.com
Roberts, Eileen	LMI	703-917-7263	eroberts@lmi.org
Shapiro, Gerald	LMI	703-917-7401	gshapiro@lmi.org
Villani, Jim	LMI	703-917-7302	jvillani@lmi.org
Wingrove, Earl	LMI	703-917-7387	ewingrov@lmi.org
Woyak, Scott	Phoenix Integration	540-231-7215	woyak@phoenix-int.com

Agenda:

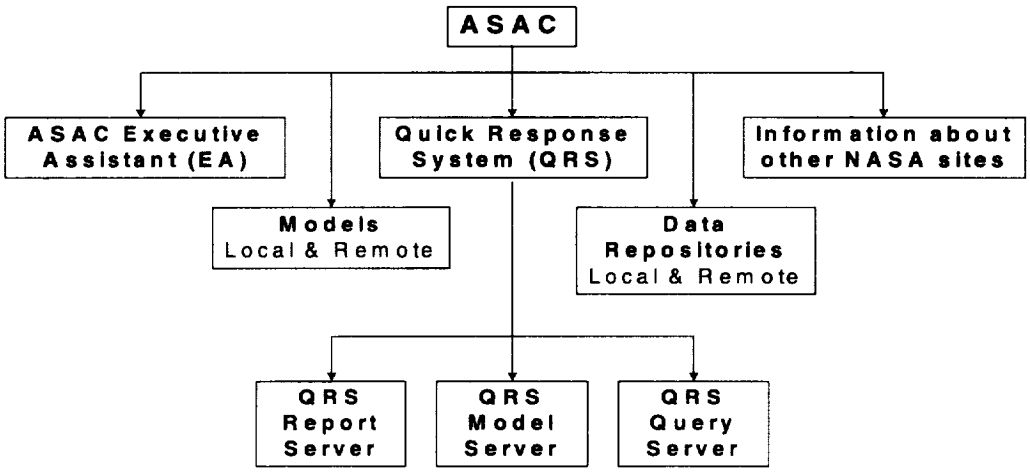
Initial agenda for the ASAC meetings:

Monday March 11	**Tuesday March 12**	**Wednesday March 13**	**Thursday March 14**	**Friday March 15**
Terms	Tasks	Models	Data Repositories	Information Flow – Models
Goals	Requirements		New Data	NARIM discussion
Users				
Monday March 18	**Tuesday March 19**	**Wednesday March 20**	**Thursday March 21**	**Friday March 22**
Information Flow – Models	First Generation ASAC	Define sample problems	Detailed Information – First Generation Models	Detailed Information – Remaining Models
Analyses		Define user scenarios		
		Detailed Information – First Generation Models		

The meetings were completed on 20 March 96. The Detailed Information – Remaining Models session was not held; it is too early to address non-First Generation ASAC models in detail.

What is ASAC?:

Create a high-level ASAC diagram.



Users:

Who are ASAC users?
How will they use ASAC?

ASAC work started at NASA Headquarters. Responsibility moved to NASA Ames and will probably move to NASA Langley.

Two general types of users:

- Macro analysis
  Used for high level decision-making and resource allocation
- Bottom up
  Used for detailed decision-making, e.g., determine if a technology is worth pursuing.

Users are listed below:

- NASA Headquarters
  - Performs macro analysis
- NASA Research Centers (Ames, Langley, and Lewis)
  - Conduct technology evaluations
  - Evaluate the NASA research program
  - Manage and conduct NASA aeronautical research program
- FAA
  - ASD-400
    - Involved with substantial parts of NASA research
- Academic researchers
  - Contractors to NASA and FAA for model development and studies
    - Wyle Labs
    - Draper Lab
    - MIT
- Aircraft manufacturing and air transportation industries
  - Contractors to NASA and FAA for studies. Validate results of our models.
    - Pratt & Whitney
    - Lockheed
    - Lockheed Martin
    - Air carriers
- LMI
  - Performs analysis and studies for all the above users. Defines and develops ASAC.

Goals and Attributes:

Are the existing goals and attributes of ASAC clear? Are they valid?
Re-write, add, and delete goals and attributes as required.

Goals:

1. The primary NASA goal for ASAC is to evaluate its aeronautics research program. NASA will use ASAC both internally, for planning at several organizational levels, and externally, to justify the resulting program to stakeholders, e.g., to the Congress and to contractors and grantees or would-be grantees. Therefore, ASAC must be capable of performing analyses of fundamental importance to NASA, other government agencies, and industry, at both a macro and detailed level. This goal requires impeccable objectivity, credibility, and the complete absence of conflicts of interest.
2. The principal objective of ASAC is to develop credible evaluations of the economic and technological impacts of advanced aviation technologies on the integrated aviation system.
3. ASAC should provide access to a collection of data and analytical tools that allow researchers at NASA and elsewhere to quickly evaluate the economic potential of alternative technologies and systems.
4. ASAC must serve as a commonly accepted, credible vehicle for interaction and cooperation both within NASA and among NASA, other government agencies, and industry.
5. ASAC should protect proprietary data, commercial data, and intellectual property.
6. ASAC's functional responsibilities are to allow independent and incremental development of the capability.

Attributes:

1. Model the decision-making of the air carriers who actually buy and operate aircraft and systems.
2. Maximize the enhanced analytical capability that can be achieved within the first two years of development.
3. Incorporate modular operation of individual models wherever feasible, using available models as appropriate.
4. Minimize NASA's risk during the development period by limiting the amount of model integration required, and by designing model improvements and development to reduce the number of tasks on critical paths.
5. Allow for a hierarchy of models at varying levels of detail that are appropriate for the analytical task at hand. However, when such detailed information is not needed, the computational burden can be greatly reduced by selecting only those critical items of data needed to pass the analysis on to the next step.

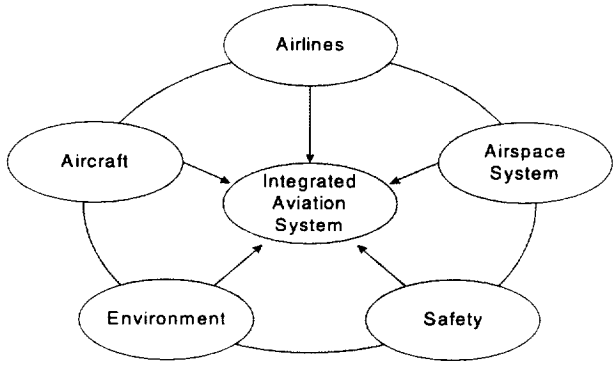
Terms:

Define ASAC related terms.

- Air carrier is the default name when referring to an airline or other air transportation firm.
- Air Transportation System – see Integrated Aviation System.
- Airline – see Air carrier.
- ASAC Data Repositories are all data used in ASAC.
- ASAC Executive Assistant (EA) is a tool used to help a user navigate ASAC.
- ASAC Information System – see Aviation System Analysis Capability.
- ASAC Server hosts the ASAC local data repositories and local models.
- ASAC Steering Committee is a group composed of members from NASA Headquarters, each NASA Research Center, and LMI. This group met during the first year of ASAC. The group no longer meets.
- Aviation System – see Integrated Aviation System.

- Aviation System Analysis Capability (ASAC) is a decision support system consisting of models, databases, and tools. It is used to support analysis of the effects of advanced technologies on the integrated aviation system.
- "Bootstrap" Development Method – see Spiral Development Method.
- Decision Support System – see Aviation System Analysis Capability.
- "Dispersed" Operating Mode – see Distributed Operating Mode.
- Distributed Operating Mode is a geographically dispersed ASAC system.
- FAR – Federal Aviation Regulations
- First Generation ASAC – A standalone, demonstration prototype composed of a selection of ASAC models and data repositories.
- Instrument Flight Rules (IFR) are rules that are used when a pilot is flying using instruments. All commercial aircraft use instrument flight rules.
- Instrument Landing System (ILS) is an airport landing system for periods of inclement weather.

- Integrated Aviation System – also known as the "Whitehead Wheel".



- QRS Model Server is a tool that provides user access to selected local and remote ASAC models.
- QRS Query Server is a tool that provides the user with predefined queries that allow real-time access to the ASAQ data repositories.
- QRS Report Server is a tool that delivers formatted reports based on data located in the ASAC data repositories.
- Quick Response Capability (QRC) – see Quick Response System.
- Quick Response System (QRS) is an automated on-line capability to access a subset of ASAC models and databases to support analysis.
- Spiral Development Method is an interdependent development process in which ASAC integrates and develops models and databases and validates the process by applying them to real analytical problems.
- TAP airports are the initial 10 airports being researched by LMI for TAP. They are: ATL, BOS, DFW, DTW, EWR, JFK, LAX, LGA, ORD, and SFO.
- TAP data is the data used by LMI to support TAP research and evaluations.
- Terminal Area Forecast (TAF) gives historical data and predictions for several measures of aviation activity, for about 4000 United States public use airports. It is maintained by the FAA.
- Terminal Area Productivity (TAP) is a NASA AST program element, consisting of a collection of research efforts aimed at achieving VFR capacity during IFR weather.
- Top-down approach is the operational concept developed for ASAC. It concentrates on the data required to perform the economic analysis of aircraft operations and investments.
- Visual Flight Rules (VFR) are rules that are used when a pilot is flying by sight (without instruments).

Tasks:

Review the tasks listed in the Strategic Development Plan. Are they valid? Re-write as necessary.

Task	Title	Description	NS601	NS602	NS603	New
1.0	ASAC Integration Functions	Build the ASAC. Provide for model access; develop databases and tools.	X			
Development of ASAC -- Six areas of analysis						
2.0	Aircraft Technology & Aviation Industry System Analyses	Incorporate ACSYNT and FLOPS into ASAC. Build a detailed cost-benefit model to analyze the viability of advanced aircraft technologies.	X	X		
3.0	Airspace Information and Modeling	Construct databases describing air traffic and airport operations and capacity. Build tools to analyze the potential impact of airspace technologies and new aircraft.			X	
4.0 *(was 4.0 and 5.0)*	Airline Operations Analysis and Investment Modeling	Build a model to generate forecasts of air travel demand, airline costs, and required aircraft inventories. Build a second model that estimates the cost to operate a representative (or actual) airline route structure with different aircraft.	X	X		
5.0 *(was 6.0)*	Model Impact on the U.S. Economy, Employment, and Trade	Quantify the impact of various technologies on the U.S. economy, employment, and trade.		X		
6.0 *(was 7.0 and 8.0)*	Methods to Assess Environmental Impacts	Estimate the noise characteristics of new aircraft and evaluate the noise impacts of aircraft operations on airport areas. Design methods to analyze the impact of aircraft operations on the atmosphere, in both the terminal area and enroute air spaces.		X		X
7.0 *(was 9.0)*	Develop System Safety Information and Measures	Provide access to searchable databases of aviation incidents and accidents. Develop a model to estimate system safety tolerance.				X
Applications of ASAC -- Studies						
8.0 *(new)*	Technology Integration Studies	Perform analytical studies in parallel with ASAC development.				X

Requirements:

Are the existing ASAC requirements clear? Are they valid?
Re-write, add, and delete requirements as necessary.

1. ASAC must be able to analyze aircraft, airspace, air carrier operations and investments, environmental issues, and safety.
2. Build the ASAC. Provide for model access; develop databases and tools.
3. Incorporate ACSYNT and FLOPS into ASAC.
4. Build a detailed cost-benefit model to analyze the viability of advanced aircraft technologies.
5. Construct databases describing air traffic and airport operations and capacity.
6. Build tools to analyze the potential impact of airspace technologies and new aircraft.
7. Build a model to generate forecasts of air travel demand, airline costs, and required aircraft inventories.
8. Build a second model that estimates the cost to operate a representative (or actual) airline route structure with different aircraft.
9. Quantify the impact of various technologies on the U.S. economy, employment, and trade.
10. Estimate the noise characteristics of new aircraft and evaluate the noise impacts of aircraft operations on airport areas.
11. Design methods to analyze the impact of aircraft operations on the atmosphere, in both the terminal area and enroute air spaces.
12. Provide access to searchable databases of aviation incidents and accidents.
13. Develop a model to estimate system safety tolerance.
14. Perform analytical studies in parallel with ASAC development.
15. A common query tool will be developed so that the interface to the databases appears seamless to the user.
16. The Executive Assistant (EA) gives users guided access with uniform "look and feel," to information now widely scattered and in widely varying formats.
17. ASAC analyses methods will be openly represented (documented and described) to the extent allowed (protect proprietary data and intellectual property rights). (ASAC documentation will provide descriptions of the model input and output files.)
18. Implementation of ASAC should remain flexible as analyses are performed in response to the changing need of its users (users can substitute models and data).
19. Implementation of ASAC should allow new models and tools to be added to the system.
20. ASAC shall provide on-line responses, and shall provide immediate feedback on potential economic benefits when analytically appropriate.
21. ASAC will accommodate operation of its models and databases at remote sites.
22. ASAC will provide the ability to evaluate a class of pre-defined representative problems.
23. Selected data repositories will be updated.
24. Models may use multiple data repositories.
25. External data will be made available to models as input.
26. ASAC shall provide raw and processed data on air carrier operations, aircraft utilization, and aircraft operating costs.
27. ASAC should provide on-line access to the NASA and FAA sites.
28. ASAC should provide on-line access to the NASA Technical Report server.
29. During development, ASAC should restrict Internet access to specific remote machines.
30. ASAC will provide system security in accordance with best commercial practices.
31. ASAC will provide the capability to exercise individual models.
32. ASAC should allow the user to view and modify data at intermediate steps (visually inspect and change data being transferred between models).
33. ASAC will provide an audit trail showing models used, and input and output files.

34. ASAC will establish standards to be used by model developers.
35. ASAC will use industry standards such as HTML 2.0 – ***need to research and list***
36. ASAC development will be under configuration control.
37. ASAC will use existing models to the maximum extent possible (reusability).
38. ASAC will provide a message to the user indicating the time required to deliver a response.
39. ASAC will support Windows, Macintosh, HP-UX, SunOS, SGI IRIX, and IBM AIX.
40. ASAC results should be consistent and reliable.
41. ASAC should request that legacy model developers provide advance notice of changes, detailed information about changes, and allow LMI to be a Beta tester for changes.
42. ASAC EA shall run analysis comprised of multiple models.
43. ASAC EA shall run one model (segment piece), including all input and outputs used for ASAC
44. EA model inputs shall be predefined.
45. EA model inputs shall be user changeable.
46. EA models shall allow for multiple inputs.
47. The EA shall determine the number of times the model needs to be run.
48. Invalid EA model inputs for a specific analysis shall be grayed out.
49. EA model inputs may be a file.
50. EA model outputs shall be viewable in raw and converted format.
51. EA analysis shall stop at selected intermediate steps.
52. The EA shall allow users the option of viewing intermediate results.
53. The EA shall allow users to select the model inputs and outputs to view.
54. The EA shall allow user to save viewed data.
55. The EA shall mail a notification of completion to the user.
56. The EA shall provide information message with estimate of how long selected items will take to run.
57. The EA shall allow users to cancel analysis at all intermediate steps.
58. The EA shall provide a general optimizing tool.
59. For each analysis, the EA shall create an analysis document that contains elements needed to recreate the analysis.
60. The EA shall provide a link to all ASAC model log-in screens. User must have access to the model and must log-in as a user.

Models:

For every ASAC model, provide the following information:

Name
Description
A/k/a name
Is it new?
Is it self-contained?
Is it modular?
Related database
POC
When will model be available
Platform
OS
Development language.

- Aircraft/ATC Functional Analysis Model
  - Evaluates demands; pilot and controller task loading of flights through the national airspace system
  - New - under development
  - Is self-contained
  - Owned jointly by LMI and CSSI, POC is Mel Ethridge
  - Prototype will be complete 12/96 - in Extend (OO simulation application)
  - No related database
  - Long term will be in UNIX, C++
- Aircraft Synthesis (ACSYNT) Model
  - Conceptual-level aircraft design and sizing model
  - Is self-contained
  - Owned by NASA Ames and Phoenix Integration, POCs are Paul Gelhausen and Phoenix Integration
  - May merge with FLOPS
  - Potential for two versions of ACSYNT – Ames version (may be "Web-ACSYNT") and Phoenix Integration version
  - No related database
  - UNIX for HP, IBM, and SGI
  - C, FORTRAN, and C++
- Ames Research Center (ARC) Models
  - Future models
  - Owned by NASA Ames, POC is Tom Galloway
  - *Pete to provide information*
- Approximate Network Delays (AND) Model
  - A national network model of airports
  - Exists
  - Is self-contained
  - Jointly owned by MIT and MITRE, POC is Pete Kostiuk
  - No related database, will use ASAC repositories
  - UNIX - Sun
  - C
- ASAC Aggregate Economic Model
  - Future model
  - POC is Pete Kostiuk
- ASAC Air Cargo Demand Model
  - Future model
  - POC is Pete Kostiuk
- ASAC Air Carrier Flight Track Efficiency Model
  - Calculates extra cost to an airline of flying an inefficient flight track due to noise restrictions; a GIS.
  - New - to be developed
  - Self-contained
  - Owned by LMI, POC is Earl Wingrove
  - Developed by Wyle
  - Complete 1/97
  - Requires input from the Integrated Noise Model
  - Related database: Airport Flight Track Alternatives Database
  - PC based
  - C++ or Visual Basic

- ASAC Air Carrier Investment Model
  - Forecasts air travel demand and airline costs
  - Exists
  - Owned by LMI, POC is Jesse Johnson
  - Excel v4.0 spreadsheet
  - Will be modified, modifications complete 1/97
  - Related database: B-43 inventory data (will become part of ASAC data repositories)
  - Will migrate to C
- ASAC Air Carrier Network Cost Model
  - Calculates operating cost of air carrier route structure given aircraft technology and the ATM system.
  - New - to be developed
  - Is self-contained
  - Owned by LMI, POC is Bruce Kaplan
  - Developed by LMI
  - Requires input from the Flight Segment Cost Model
  - No related database, will use ASAC data repositories
  - UNIX
  - C
- ASAC Airport Capacity Model
  - Calculates the capacity of an airport given information such as technology parameters and weather.
  - New - under development
  - Is self-contained
  - Owned by LMI, POC is Gerald Shapiro
  - Complete 1/97
  - Related database: Weather data
  - UNIX
  - Pascal
- ASAC Airport Delay Model
  - Calculates the delay at an airport.
  - New - under development
  - Is self-contained
  - Owned by LMI, POC is Gerald Shapiro
  - Complete 1/97
  - No related database
  - UNIX
  - Pascal
- ASAC Flight Segment Cost Model
  - Calculates the cost of flying a particular aircraft from one city to another
  - New - to be developed
  - Is self-contained
  - Owned by LMI, POC is Pete Kostiuk
  - Will interface with ACSYNT
  - Will feed the Air Carrier Network Cost Model
  - Complete 1/97
  - No related database, will use ASAC data repositories
  - UNIX
  - C++

- ASAC General Aviation Economic Model
  - Future model
  - POC is Jesse Johnson
- ASAC Preferential Runway Use Model
  - Evaluates the impact of preferential runway use schemes and associated ground delays
  - New - to be developed
  - Is self-contained
  - Owned by LMI, POC is Earl Wingrove
  - Developed by Wyle
  - Complete 1/97
  - Related database: Runway Use database
  - PC based
  - C++ or Visual Basic
- ASAC Regional and Commuter Economic Model
  - Future model
  - POC is Pete Kostiuk
- ASAC Regional and Commuter Network Cost Model
  - Future model
  - POC is Pete Kostiuk
- ASAC System Safety Tolerance Analysis Model
  - Future Model
  - POC is Pete Kostiuk
- BOS and DTW LMI Airport Capacity Models
  - Serve as the basis for the ASAC Airport Capacity Model
- COSTADE Cost Optimization Software for Transport Aircraft Design Evaluation
  - Part of FLOPS
  - Is not standalone
- COSTMOD Cost Optimization Software for ???
  - Description???
  - Exists
  - Owned by ?, POC is ?
  - Related database?
  - UNIX
- Emission and Dispersion Modeling System (EDMS)
  - Calculates emissions within a airport from ground and air
  - Exists
  - Is self-contained
  - Owned by FAA, POC is Bob Hemm
  - No related database
  - PC-DOS?
  - Spreadsheet?
- Flight Optimization System (FLOPS) Model
  - Conceptual-level aircraft design and sizing model
  - Exists
  - Is self-contained
  - Owned by NASA LaRC, POC is Phoenix Integration
  - No related database
  - UNIX
  - FORTRAN

- Global Aircraft Emissions Forecasting (GAEF) Model
    - Analyzes enroute emissions
    - Owned by FAA, POC is Pete Kostiuk
    - No related database, may use ASAC data repositories
    - Exists?
- Integrated Noise Model (INM)
    - Estimates noise at an airport
    - Exists
    - Is self-contained
    - Owned by FAA/ATAC, POC is Bob Hemm
    - No related database, will use ASAC data repositories (schedule information)
    - Windows
- LMI Airline Investment Model
    - Same as the ASAC Air Carrier Investment Model
- LMI Network Model
    - Potential substitute for the AND model
    - Exists
    - Is self-contained
    - Owned by LMI, POC is Dave Lee
    - No related database, will use ASAC data repositories (OAG)
    - UNIX
    - C
- National Airspace Research and Investment Model (NARIM)
    - Simulation model of the national airspace system, plus other tools
    - Parts exist
    - Owned by FAA, funded by NASA (part of ASAC), POCs are Steve Bradford (FAA) and Bill Colligan (CSSI).
    - Modular
    - NASPAC+ is a part of this model
    - No related database
    - UNIX
    - C, FORTRAN?
- National Airspace System Performance Analysis Capability (NASPAC)
    - Latest version is NASPAC+
- NASPAC+
    - Provides national airspace system performance analysis capability
    - Part of NARIM
    - Exists
    - Is self-contained
    - Owned by FAA, POC is Bill Culligan
    - No related database
    - UNIX
    - C, FORTRAN
- Navy/NASA Engine Program (NNEP)
    - Is part of ACSYNT
- Network Cost Model
    - Is the ASAC Air Carrier Network Cost Model

- Noise Impact Model
  - Is the ASAC Air Carrier Flight Track Efficiency Model plus the Preferential Runway Use Model
- Segment Cost Model
  - Is the ASAC Flight Segment Cost Model
- Small Transport Aircraft Technology (STAT) Model
  - Preliminary aircraft design model for small aircraft (regional and commuter aircraft)
  - Exists
  - Is self-contained
  - Owned by NASA Ames, POC is Tom Galloway
  - Related database???
  - Mainframe based?

Data:

Does the data exist or is it new? Is the data associated with a model? Is the data external or internal to a model? Is the data local or remote?

Data Repositories:

- Airport Flight Track Alternatives
  - Data covers 20 airports
  - Used by the ASAC Air Carrier Flight Track Efficiency Model
  - Under development
  - External
  - Local
- ASAC Data Repository at LMI (ASAC Server) – all the following data exists:
  - DOT Airline Service Quality Performance (ASQP)
  - DOT Form 41
    - DOT Form 41 Financial
    - DOT T-3/T-100 Airport Rank
    - DOT T-100 Flight Segment
    - DOT Origin and Destination
  - FAA Terminal Area Forecast (TAF)
  - Official Airline Guides (OAG) North American and Worldwide Merge
  - World Jet Inventory
- Runway Use
  - New
  - Data covers 20 airports
  - Used by the ASAC Preferential Runway Use Model
  - External
  - Local

New Data:

Data that will be added to the ASAC Server (near term):

- 1994 data for Form 41 Financial
    - May be model related
    - Standalone
    - Exists
- 1994 data for Origin and Destination Matrices
    - May be model related
    - Standalone
    - Exists
- 1994 data for T-3/T-100 Airport Rank
    - May be model related
    - Standalone
    - Exists
- 1994 data for T-100 Flight Segment
    - May be model related
    - Standalone
    - Exists
- 1995 data for World Jet Inventory
- 1995 data for ASQP
- Airport weather data
    - Used by the ASAC Airport Capacity Model
    - From the NCDC
    - Standalone
    - Exists
    - Flat file
- B-43 aircraft inventory plus data added by LMI, i.e., noise
    - Used by the ASAC Air Carrier Investment Model
    - Standalone
    - Exists
    - Spreadsheet
- FAA Noise Certification Database
    - May be model related
    - Standalone
    - Exists
- TAF data update

Other new data:

- ASRS database of aviation accidents and incidents
    - Used for safety (task 7)
    - No model relation
    - Exists
- Avmark Commercial Aircraft Fleets
    - Provides international aircraft inventory by age
    - model relation?
    - Exists

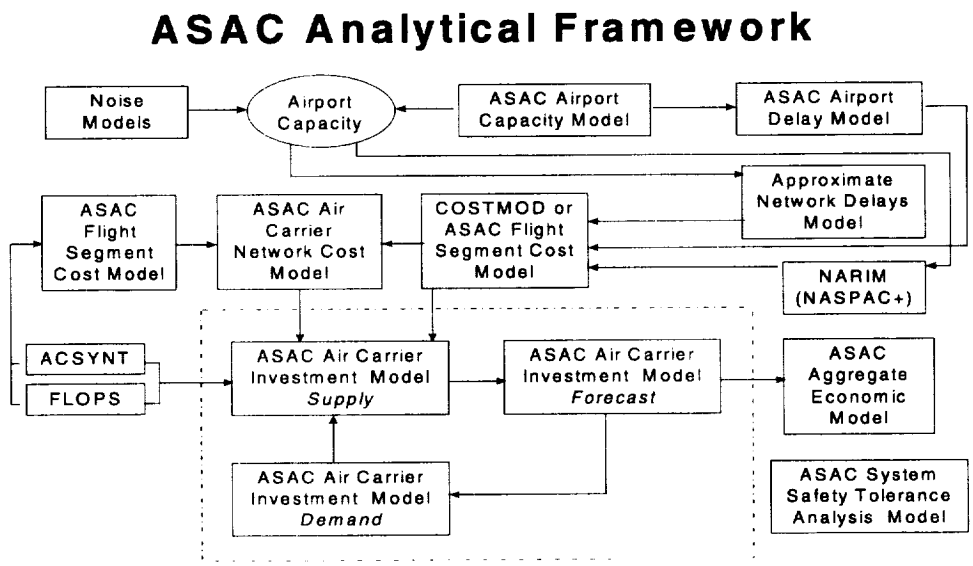
- Boeing Current Market Outlook
    - May add to the QRS and ASAC Server
    - No model relation
    - Exists
- FAA Aircraft Emissions Database
    - May add to the QRS and ASAC Server
    - No model relation
    - Exists
    - Spreadsheet
- High Altitude Wind Database
    - From Goddard
    - Exists
- International Civil Aviation Organization (ICAO) Airport Characteristics Data Bank
    - Provides international airport layouts
    - model relation?
    - Standalone
    - Exists
- International Civil Aviation Organization (ICAO) Airport Traffic
    - Provides historic airport operations data
    - model relation?
    - Standalone
    - Exists
- Leasing Data
    - New
    - Source to be determined
    - May add to the QRS and ASAC Server
- NTSB database of aviation accidents and incidents
    - Used for safety (task 7)
    - No model relation
    - Exists

Data External to ASAC:

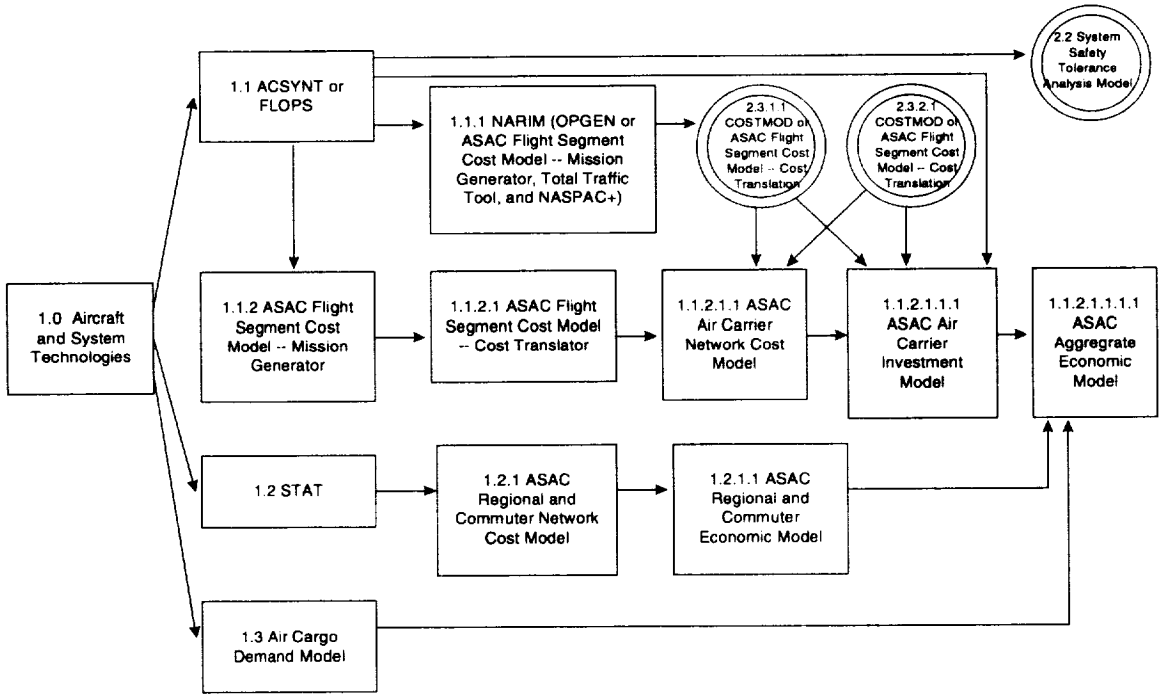
- Avitas BlueBook
- Avmark Transport Aircraft Values
- BLS Employment figures
- Comparative Economic Statistics (GDP, GNP, Productivity), source to be determined
- DOC Input-Output Tables
- Gentsch and Peterson Model
- ICAO Air Carriers' Fleet and Personnel
- ICAO Air Carriers' Financial
- ICAO Air Carriers' Fleet and Personnel
- ICAO Air Carriers' Traffic
- ICAO Onflight Origin and Destination
- ICAO Survey of International Transport Fares
- ICAO Traffic by Flight Stage
- Standard and Poor's
- Value Line

Information Flow – Models:

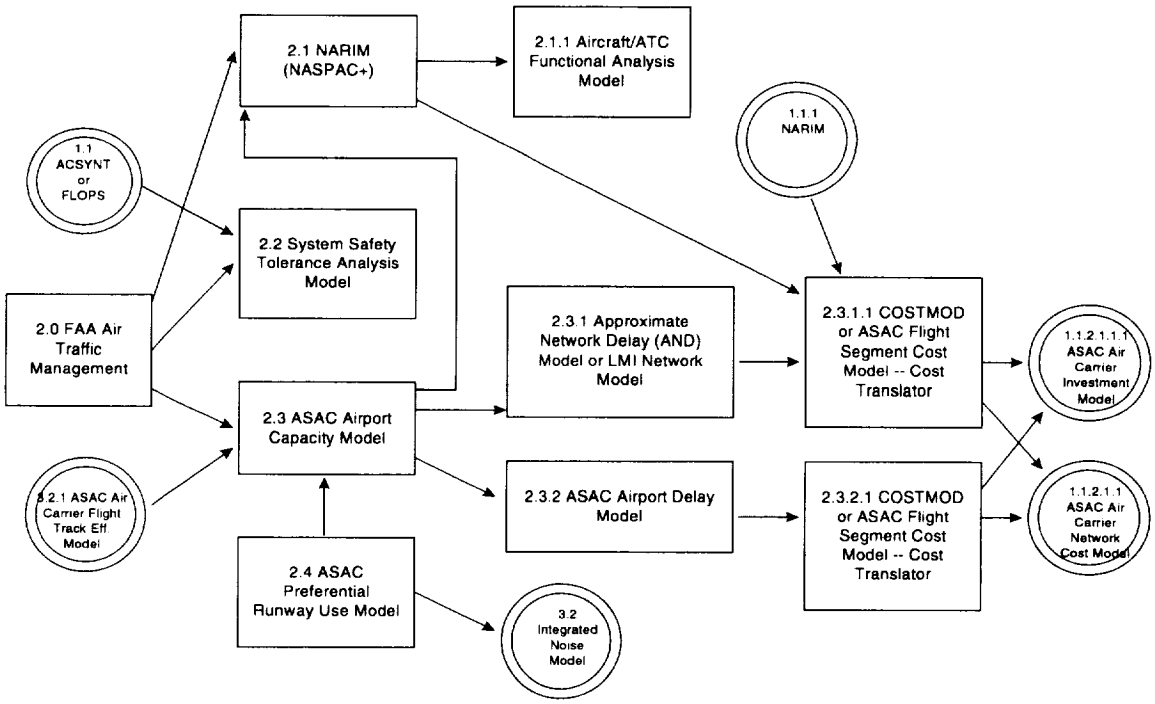
Discuss information elements and create data flows.



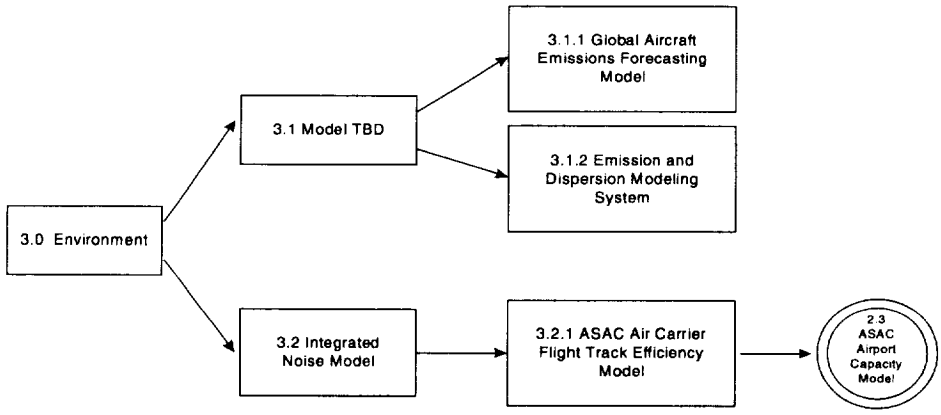
1.0 Aircraft and System Technologies



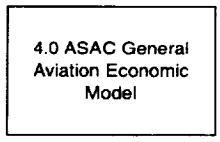
2.0 FAA Air Traffic Management



3.0 Environment



4.0 ASAC General Aviation Economic Model



Additional information on each of the above listed models can be found in Annex C-1.

Analytical Functions:

Are all analytical functions represented? Add items to data flows and data repositories as required.

Aircraft Technology and Aviation Industry

Analytical Function	Data Flow
Translate aircraft technology concepts into aircraft physical performance measures, such as speed, fuel burn, payload, range, and airfield accessibility.	ACSYNT, FLOPS
Estimate development and production costs for new aircraft (including airframes, propulsion, and avionics) and substantial modifications to existing aircraft and aircraft components, from preliminary aircraft design information.	ACSYNT, FLOPS
Estimate operating costs for new and existing aircraft as a function of utilization	ACSYNT, FLOPS, STAT
Estimate the economic viability of advanced aircraft designs, through quantitative technical and business performance comparisons of conceptual designs with existing aircraft or other possible alternatives	ACSYNT, FLOPS ⇒ ASAC Aggregate Economic Model, ASAC Air Carrier Network Cost Model
Evaluate design or operational constraints.	ACSYNT, FLOPS

Airspace

Analytical Function	Data Flow
Translate Air Traffic Control (ATC) technology concepts and investments into system performance parameters.	FAA ⇒ ASAC Airport Capacity Model ⇒ NARIM
Estimate air transportation system performance changes due to the introduction of advanced technologies, with performance measured by total travel time, delay time, and total system operating costs.	FAA ⇒ ASAC Airport Capacity Model ⇒ NARIM
Evaluate the impact of improved ATC performance on passenger and cargo services.	FAA ⇒ COSTMOD or ASAC Flight Segment Cost Model ⇒ ASAC Air Carrier Investment Model
Provide representative flight schedules for use in analyzing the impact of new aircraft and Air Traffic Management (ATM) technologies on air transportation system performance.	Data Repositories (OAG)

Air Carrier Operations

Analytical Function	Data Flow
Provide up-to-date estimates for air carrier flight segment and cost information to serve as a baseline comparison for evaluating new technologies, and changes in carrier operational strategies.	Data Repositories (T-100, Form 41)
Provide representative flight schedules for use in forecasting the use and impact of new aircraft and procedures by airlines.	Data Repositories (OAG)
Evaluate changes in procedures to maximize existing technology compared to stasis in procedures coupled with new technology.	FAA ⇒ NARIM

Air Carrier Investments

Analytical Function	Data Flow
Provide current airframe and engine inventory data, linked to utilization data on flights by aircraft type and city pair.	Data Repositories (Boeing World Jet Airplane Inventory, T-100, OAG)
Estimate air carrier industry cost, revenue, and profit functions.	ASAC Air Carrier Investment Model
Estimate air carrier investments as a function of aircraft characteristics, aircraft acquisition and operating costs, and air traffic performance.	ACSYNT, FLOPS ⇒ ASAC Aggregate Economic Model
Estimate long-run world-wide demand for passenger and cargo transportation services, as a function of demographics, economic variables.	ASAC Air Carrier Investment Model
Provide long-run forecasts of aircraft purchases	ASAC Air Carrier Investment Model
Estimate the long-run demand for general aviation aircraft, as a function of aircraft acquisition and operating costs, and aircraft performance, by category (e.g., business jet, single piston).	ASAC General Aviation Economic Model

Aggregate Economic Impacts

Analytical Function	Data Flow
Estimate impact on the U.S. economy of the introduction of new technology and other changes in the air transportation sector, based on the aviation industry analysis results.	Aggregate Economic Model
Calculate the feedback effect of general economic conditions on the aviation-related industries.	ASAC Air Carrier Investment Model, ASAC General Aviation Economic Model, ASAC Air Cargo Demand Model, and ASAC Regional and Commuter Economic Model ⇒ ASAC Aggregate Economic Model
Estimate the non-US impact of new technologies (international sales and market share).	ASAC Air Carrier Investment Model ⇒ ASAC Aggregate Economic Model
Evaluate the potential implications of regulatory changes in other countries on the sales and services of U.S. airlines and manufacturers.	ASAC Air Carrier Investment Model ⇒ ASAC Aggregate Economic Model

Environmental Impacts: Noise

Analytical Function	Data Flow
Integrate noise exposure models and demographic databases for use in estimating community noise impact.	Integrated Noise Model ⇒ ASAC Air Carrier Flight Track Efficiency Model
Analyze noise effects at selected airports resulting from advanced designs and aircraft modifications.	Integrated Noise Model ⇒ ASAC Air Carrier Flight Track Efficiency Model ASAC Preferential Runway Use Model ⇒ Integrated Noise Model
Develop an engine/airframe noise database for rapid comparison with advanced designs and potential market analysis.	Data Repository (FAA Noise Certification Database)
Quantify the impact of local noise restrictions on airline operations and the overall level and quality of air transportation services provided.	ASAC Air Carrier Flight Track Efficiency Model, ASAC Preferential Runway Use Model, ASAC Airport Capacity Model

Environmental Impacts: Emissions

Analytical Function	Data Flow
Quantify the emissions of critical pollutants from aircraft operations.	EDMS Model, Data Repository
Provide a database of emission profiles of existing engines.	Data Repository (FAA Aircraft Emissions Database)
Link engine emission data to aircraft flight information to build a distributed emissions database using actual and projected aircraft designs.	GAEF Model

Safety

Analytical Function	Data Flow
Predict safety impacts of advanced aircraft designs based on current and proposed regulations.	ACSYNT, FLOPS, ASAC System Safety Tolerance Analysis Model
Evaluate the impact of safety regulations on the cost, quantity, and quality of air transportation services.	ACSYNT, FLOPS
Provide access to current aviation accident/incident databases.	Data Repositories (NTSB, ASRS)
Develop methodologies for evaluating the safety of the air transportation system from the overall system viewpoint.	ASAC System Safety Tolerance Analysis Model

Analyses:

What type of analyses will be done? Use the analyses to validate the ASAC information flow and models. What did we miss? (what other analytical requirements are necessary?)

ASAC will answer the following questions and will require the key data elements shown:

- **Aircraft Fleets**

Category	Required data elements	Data Flow – Data Source
What is the composition of the current transport aircraft fleet?	Quantity by type of equipment, engine type and model installed on aircraft	World Jet Inventory
Which countries and firms own and operate these aircraft?	Country, company	World Jet Inventory, Leasing data
How old are the aircraft?	Dates of order, delivery, sale, lease, or loss by accident	B-43 Inventory of airframe and aircraft engines, Avmark Commercial Air Fleets
What noise stage are they?	Typical noise stage by equipment and engine combination	FAA Noise Certification Database
What level of emissions do they create?	Emissions per idle, takeoff, climbout, and approach by type of engine	FAA Aircraft Emissions Database
How is a specific type of equipment typically used?	Annual usage in terms of block hours operated, number of trips, and aircraft miles flown; average duration of flights in terms of block time and stage length	Form 41, OAG
What are the operating costs?	Operating cost per: year, typical flight, block hour, aircraft mile, and available seat mile (ASM)	Form 41

- **Underlying Demand for Air Travel**

Category	Required data elements	Data Flow – Data Source
What is the underlying world-wide demand for air travel?	Origin and destination data for: passengers, revenues, itinerary miles, and average coupons	DOT Origin and Destination, ASAC Air Carrier Investment Model
What are the airlines' revenues generated by this air travel?	Average fare per revenue passenger mile, and revenue passenger miles flown	Form 41, ASAC Air Carrier Investment Model
What are the highest density city pairs?	Passengers per day, revenue passenger miles, and aggregate revenues for the top 100 domestic and international city pairs	OAG, T-100
What are the forecasted levels of passenger and freight traffic on global, regional, and U.S. national bases?	Revenue passenger miles and revenue ton miles	ASAC Air Cargo Demand Model, ASAC Air Carrier Investment Model

- **Air Traffic**

Category	Required data elements	Data Flow – Data Source
What are the current patterns of air traffic (i.e., how do the airlines satisfy the underlying world-wide demand for air travel through their decisions about routing)? (domestic only)	Point-to-point flights versus hubbing operations	T-100, O&D
What are the current patterns of air traffic (i.e., how do the airlines satisfy the underlying world-wide demand for air travel through their decisions about equipment usage)?	Type of equipment used on each flight segment	T-100, OAG
What are the current patterns of air traffic (i.e., how do the airlines satisfy the underlying world-wide demand for air travel through their decisions about frequency of flights)?	Number of flights per day, week, and year	T-100, OAG
What are the block times for these flights?	Average block time per flight segment	T-100, OAG
Which airports are the busiest?	Ranked by number of aircraft departures and enplaned passengers	T-3, OAG
What are the capacities of U.S. and foreign airports? How do airport usage rates compare with airport capacities (which airports are the most critically constrained by their current capacities)?	Number of runways, instrument landing systems, practical annual capacity, and visual flight rule (VFR) days per year	TAF, ICAO Airport Characteristics Data Bank
What are the current levels of aircraft and passenger delay by airport and cause of delay?	Delay by cause (weather, terminal volume, center volume, closed runway or taxiway, or NAS equipment interruptions) and by phase of flight (gate-hold, taxi-out, airborne, or taxi-in)	CODAS

- **Economic Value of Airline Industry**

Category	Required data elements	Data Flow – Data Source
How important is the air transportation industry to the U.S. economy in terms of economic activity generated?	Industry revenues, gross domestic product, employment	ASAC Aggregate Economic Model
What are the largest firms in terms of output, revenues, and employment? How healthy are they in terms of operating incomes, balance sheets, and debt ratings?	Ranked by sales, profits, assets, and financial health	Form 41
Which industries are the primary suppliers to the air transportation industry?	Ranked by industry sales and value added	ASAC Aggregate Economic Model

- **Economic Value of Aircraft and Aircraft Systems Manufacturing**

Category	Required data elements	Data Flow – Data Source
How important are the aircraft and aircraft systems manufacturing industries to the U.S. economy in terms of economic activity generated?	industry revenues, gross domestic product, employment	ASAC Aggregate Economic Model
What is the current U.S. balance of trade in aircraft and aircraft systems and what is the trend?	U.S. exports minus U.S. imports	ASAC Aggregate Economic Model
Which industries are the primary suppliers to the aircraft and aircraft systems manufacturing industries?	ranked by industry sales and value added	ASAC Aggregate Economic Model

User Scenario:

Walk through how a person would use ASAC.

ASAC Executive Assistant

Two options:

1. Perform or Conduct Analyses
2. Run Standalone Model.

Perform or Conduct Analyses

Select an analysis:

1. User created analysis 1 - one or multiple models
2. User created analysis 2 - one or multiple models
3. ...
4. Model 1 (segment piece), including all input and outputs used for ASAC
5. Model 2 (segment piece), including all input and outputs used for ASAC
6. ...
7. Selected analysis 1 - one or multiple models
8. Selected analysis 2 - one or multiple models

<u>Analysis example</u>

Title, Description, models used

Model inputs:

Predefined
User changeable
Allow for multiple inputs, let the server determine the number of times the model needs to be run
Gray out invalid inputs
Input can be a file

Model outputs:

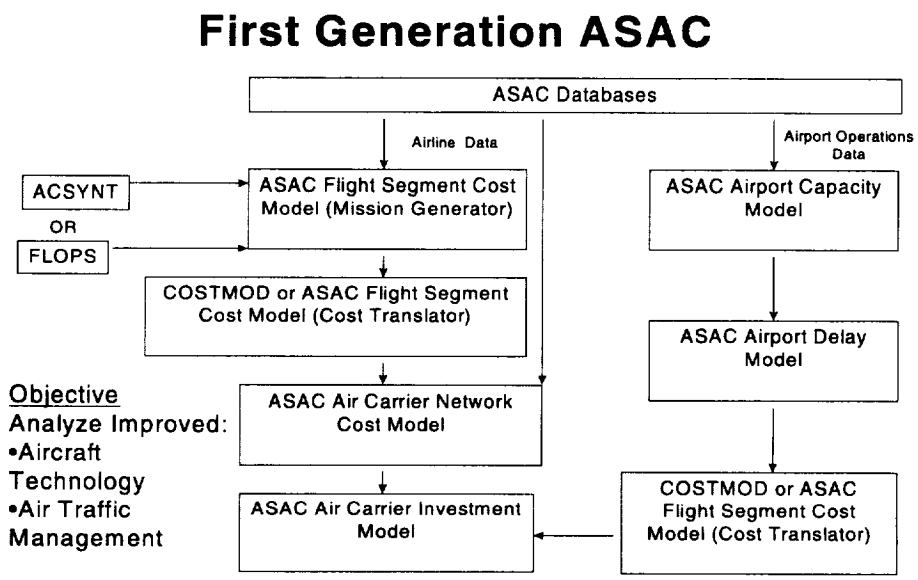
- Viewable in raw and converted format
- Stop at selected intermediate steps
- Allow user the option of viewing intermediate results
- Allow user to select the model inputs and outputs to view
- Allow user to save viewed data
- Mail notification of completion to the user
- Provide information message with estimate of how long selected items will take to run.
- Allow user to cancel analysis at all intermediate steps
- Provide a general optimizing tool
- For each analysis, create an analysis document that contains elements needed to recreate the analysis

Run Standalone Model

For all ASAC models, provide a link to model log-in screen. User must have access to the model and must log-in as a user.
Provide a disclaimer: purpose and equipment required.

First Generation ASAC:

Define the First Generation ASAC. Specify sample problems to solve using the First Generation ASAC.



Analyses to be performed using the First Generation ASAC:

Aircraft Technology

Uses ACSYNT or FLOPS, Flight Segment Cost Model, Air Carrier Network Cost Model, and ASAC Air Carrier Investment Model.

Analyze change in aircraft technology for one or two airlines.

Inputs:
- Change in propulsion technology, e.g. engine deck, to effect an:
    - 8% change in specific fuel consumption (SFC) for large engines
    - 10% change in SFC for small engines
    - Can resize airframe, or
- Change in aerodynamic inputs, or
- Change in structure weight relations.

Intermediate Outputs:
- Reduction in airline flight segment costs
- Reduction in airline network costs.

Final Outputs:
- Change in RPM flown
- Change in number of aircraft in fleet
- Change in employment levels.

Air Traffic Management Improvement

Uses ASAC Airport Capacity Model, COSTMOD, and ASAC Air Carrier Investment Model.

Analyze change in air traffic management for one or two airports. Note: display all 10 TAP airports, but gray out the ones that are not available.

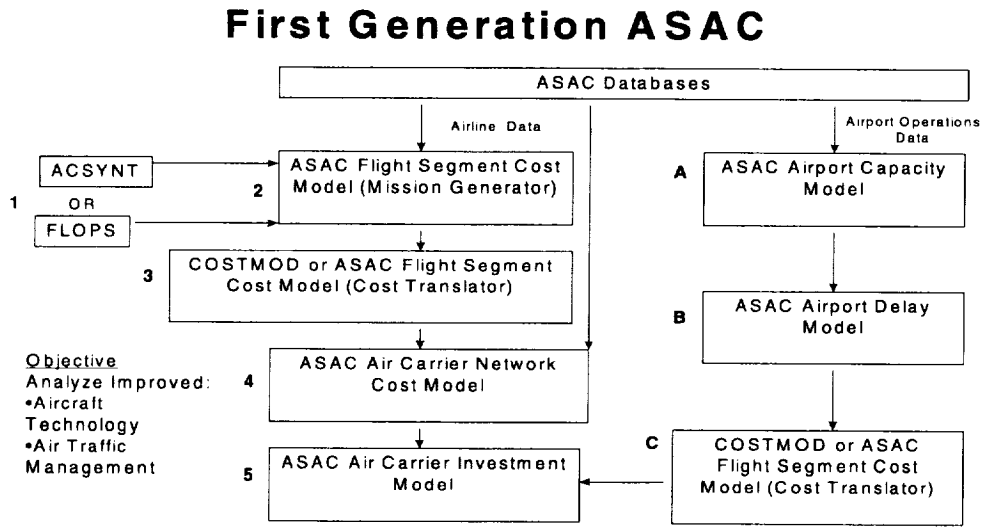
Inputs:
- Change in runway occupancy time, and/or
- Change in position uncertainty, and/or
- Change in miles in trail, and/or
- Change in approach speed uncertainty.

Intermediate Outputs:
- Reduction in arrival and/or departure delay
- Reduction in airline flight segment costs.

Final Outputs:
- Change in RPM flown
- Change in number of aircraft in fleet
- Change in employment levels.

Detailed Information – First Generation Models:

Provide detailed information for First Generation ASAC models.



Need to define where data conversion takes place.

ACSYNT (1)

Input:

Text file, 300 - 500 lines, tabular data and switch settings
Need a database of lookup information (input files) to reflect user input, e.g., if a user chooses a B-737, need data that describes a B-737

Four input categories (files):

- Engine deck
- Aerodynamic inputs
- Structure weights
- Design mission (payload, range, weight), may be fixed input

The first three categories may have pre-defined sets of data to give a specified result, e.g., to provide 8% SFC improvement in engine performance.

Each of the above four categories will have data for four different types of aircraft.

Output:

- Designed aircraft (text file)

ASAC Flight Segment Cost Model – Mission Generator (2)

Input:

- Performance model (file or lookup):
  - specific fuel as a function of altitude and mach number
  - drag as a function of lift (drag polar)

  - extracted from designed aircraft (ACSYNT output), or
  - a predefined aircraft, or
  - a user defined aircraft
- City pairs (user input)
- Latitude and longitude (lookup)
- Wind data (lookup or routine)
- Network description (from database – also used as input to ASAC Air Carrier Network Cost Model and ASAC Air Carrier Investment Model):
  - Flight schedules
  - Origin and destination airport pairs
  - Aircraft type
  - Number of flights
  - Schedule
    - Day
    - Airline
    - Scheduled arrival time
    - Scheduled departure time
    - Gate to gate time
  - Load factor
  - Number of seats
- Cost index – what to optimize (user input):
  - Fuel
  - Time
  - Fuel and Time
- Cost factors (user or table input – also used as input to ASAC Flight Segment Cost Model – Cost Translator or COSTMOD and ASAC Air Carrier Investment Model):
  - Fuel (price per pound)
  - Wages (per unit time)
  - Ownership costs
  - Insurance costs
  - Indirect operating costs
  - Passenger time factor
- Trajectory constraints:
  - Free flight
  - Change in altitude
  - Preferred routes
  - Restricted airspace

Need multiple inputs to get a network output table

Output:

- Block Fuel (table) – one per flight
- Block Time (table) – one per flight
- Trajectory (table) – three dimensional position as a function of time, one per flight

ASAC Flight Segment Cost Model – Cost Translator, or COSTMOD (3)

Input:

- Block Fuel (table) – one per flight (ASAC Flight Segment Cost Model – Mission Generator output)
- Block Time (table) – one per flight (ASAC Flight Segment Cost Model – Mission Generator output)
- Cost factors (user or table input – also used as input to ASAC Flight Segment Cost Model – Mission Generator and ASAC Air Carrier Investment Model):
    - Fuel (price per pound)
    - Wages (per unit time)
    - Ownership costs
    - Insurance costs
    - Indirect operating costs
    - Passenger time factor

Output:

- Flight segment cost by cost category (table)
- Block Fuel (table) – one per flight
- Block Time (table) – one per flight
- Flight identifier (table):
    - City pair
    - Aircraft type
    - Schedule

Table	Cost	Fuel	Time
Flight Segment			

ASAC Air Carrier Network Cost Model (4)

Input:

- ASAC Flight Segment Cost Model – Cost Translator or COSTMOD output:
  - Flight segment cost by cost category (table)
  - Block Fuel (table) – one per flight
  - Block Time (table) – one per flight
  - Flight identifier (table):
    - City pair
    - Aircraft type
    - Schedule
- Network description (from database – also used as input to ASAC Flight Segment Cost Model – Mission Generator and ASAC Air Carrier Investment Model):
  - Flight schedules
  - Origin and destination airport pairs
  - Aircraft type
  - Number of flights
  - Schedule
    - Day
    - Airline
    - Scheduled arrival time
    - Scheduled departure time
    - Gate to gate time
  - Load factor
  - Number of seats
- [Delay cost estimates (from COSTMOD) – not part of First Generation ASAC]

Output:

- Airline Operating Cost
  - Flight segment cost by cost category (table – same categories used in ASAC Flight Segment Cost Model – Cost Translator or COSTMOD)
  - Airline overhead costs (table)
- Aggregate block fuel
- Aggregate block time
- Summary statistics

ASAC Air Carrier Investment Model (5)

Input:

- Output from the ASAC Air Carrier Network Cost Model (table):
  - Airline Operating Cost
    - Flight segment cost by cost category (table – same categories used in Flight Segment Cost Model – Cost Translator or COSTMOD)
    - Airline overhead costs (table)
  - Aggregate block fuel
  - Aggregate block time

    - Summary statistics
- Network description (from database – also used as input to ASAC Flight Segment Cost Model – Mission Generator and ASAC Air Carrier Network Cost Model):
    - Flight schedules
    - Origin and destination airport pairs
    - Aircraft type
    - Number of flights
    - Schedule
        - Day
        - Airline
        - Scheduled arrival time
        - Scheduled departure time
        - Gate to gate time
    - Load factor
    - Number of seats
- Cost factors (user or table input – also used as input to ASAC Flight Segment Cost Model – Cost Translator or COSTMOD and ASAC Flight Segment Cost Model – Mission Generator):
    - Fuel (price per pound)
    - Wages (per unit time)
    - Ownership costs
    - Insurance costs
    - Indirect operating costs
    - Passenger time factor
- Economic data (user selected)
- Demographic data (user selected)
- Airport delay costs (from ASAC Flight Segment Cost Model – Cost Translator or COSTMOD)
- B-43 data (database)
- T-100 data (database)
- Target profit (user selected)
- [Development and production costs for aircraft (from ACSYNT or FLOPS) – not part of First Generation ASAC]
- [Network delay costs (from ASAC Flight Segment Cost Model – Cost Translator or COSTMOD) – not part of First Generation ASAC]

Output:

- Absolute and percentage changes in:
    - Revenue passenger miles
    - Airline revenues, operating costs, and operating profits
    - Airline employment
    - Aircraft inventory
    - Aircraft retirements
    - Aircraft purchases
    - Distribution of aircraft purchases by average seats
    - Aircraft manufacturing employment

ASAC Airport Capacity Model (A)

Input:

- Change in ATM Regulations (user input – defaults provided)
- Change in ATM Automation and Technology (user input – defaults provided)
- Airport operations:
  - Aircraft mix (user input)
  - Arrival demands (database – also used as input to ASAC Airport Delay Model)
  - Departure demands (database – also used as input to ASAC Airport Delay Model)
- Weather data (from database):
  - Ceiling
  - Visibility
  - Wind speed
  - Wind direction
  - Temperature

- Runway configurations (for each runway in a configuration, direction, ILS minima for CAT I, II, III) (database)
- [Runway available? (from ASAC Preferential Runway Use Model – not part of First Generation ASAC]

Output:

- Arrival capacity – arrivals/hour
- Departure capacity – departures/hour

(The airport capacity model returns arrival and departure capacity given the inputs described above. Usually, all inputs except weather data are fixed. Weather data generate time-varying capacity outputs).

ASAC Airport Delay Model (B)

Input:

- From ASAC Airport Capacity Model:
  - Arrival capacity (values) – arrivals/hour
  - Departure capacity (values) – departures/hour
- Arrival demands (database – also used as input to ASAC Airport Capacity Model)
- Departure demands (database – also used as input to ASAC Airport Capacity Model)

Output:

- Arrival delay statistics (table or chart)
- Departure delay statistics (table or chart)

ASAC Flight Segment Cost Model – Cost Translator, or COSTMOD (C)

Input:

- From ASAC Airport Delay Model:
  - Arrival delay statistics (table?)
  - Departure delay statistics (table?)
- Airport mix (database)
- Cost factors (user or table input – also used as input to ASAC Flight Segment Cost Model – Mission Generator and ASAC Air Carrier Investment Model):
  - Fuel (price per pound)
  - Wages
  - Ownership costs
  - Insurance costs
  - Indirect operating costs
  - Passenger time factor
- Number of seats (lookup)
- Load factors (lookup)

Output:

- Airport delay costs:
  - Arrival delay cost
  - Departure delay cost
  - Arrival delay time
  - Departure delay time
  - Passenger delay time
  - Passenger delay cost

References:

From the ASAC QRS:

Definitions of key terms
Source descriptions.

Figure 1.0. Aircraft and System Technologies

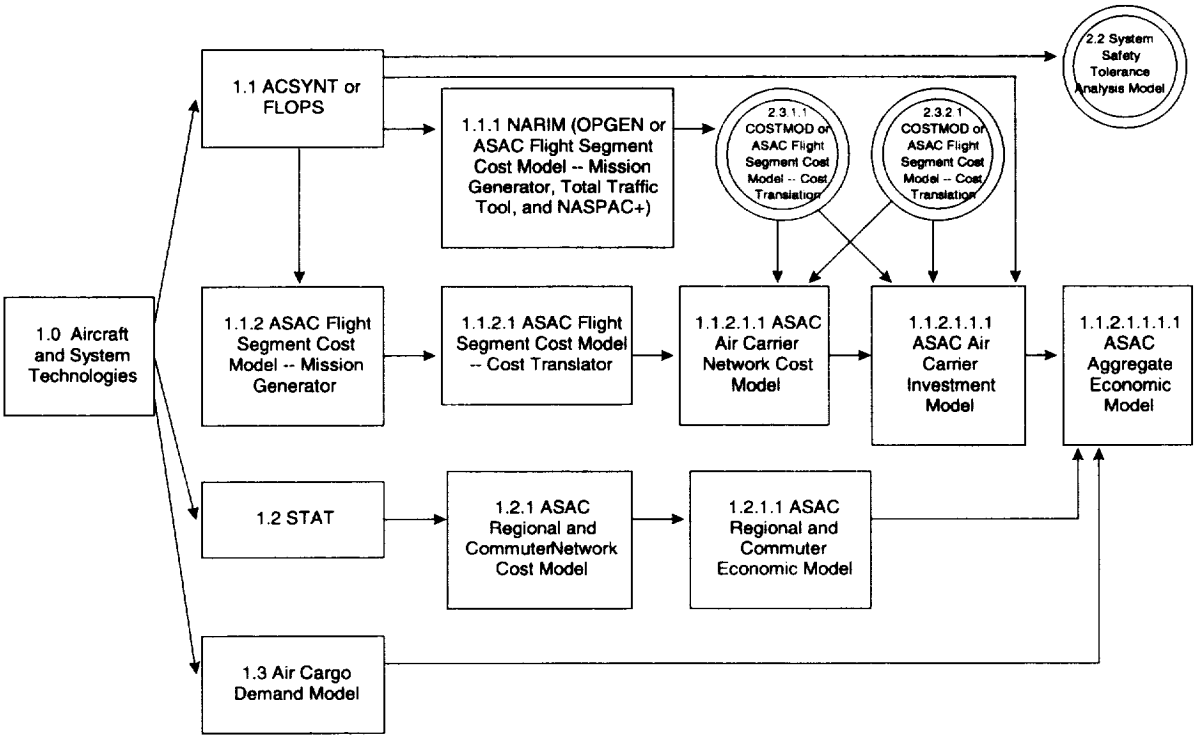


Figure 2.0. FAA Air Traffic Management

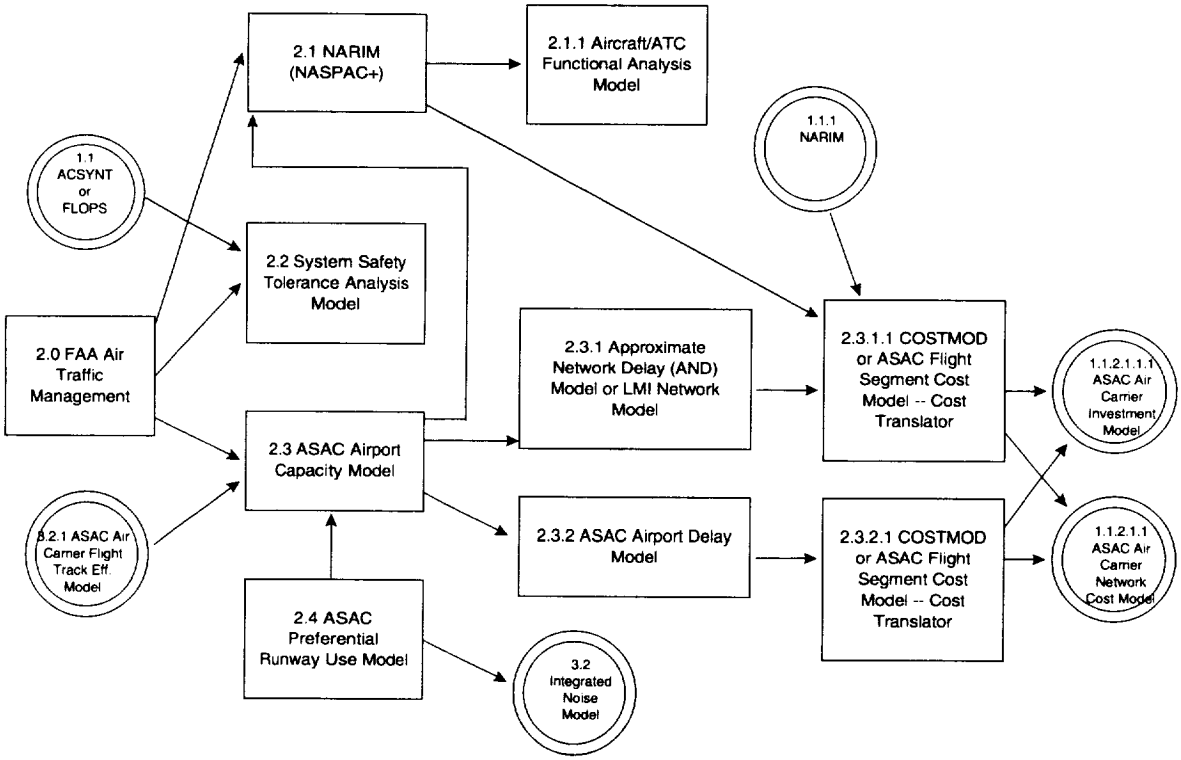


Figure 3.0. Environment

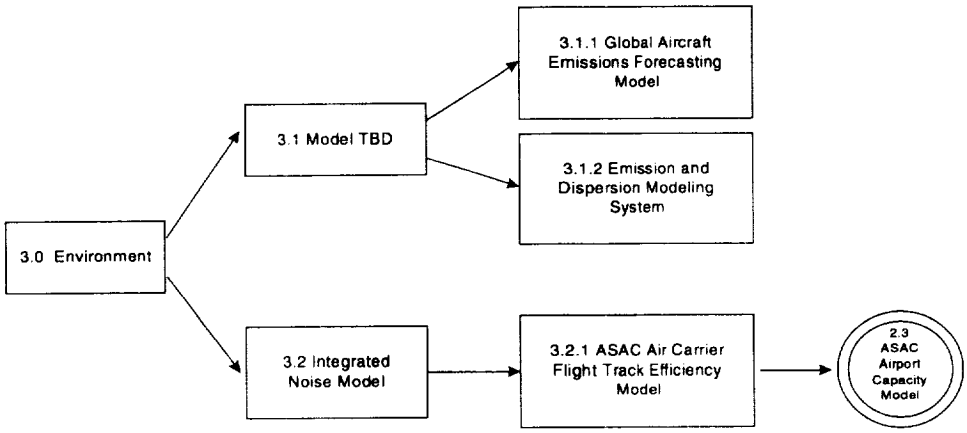


Figure 4.0. ASAC General Aviation Economic Model

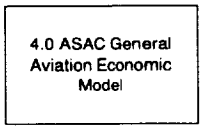


Figure 1.0. Aircraft and System Technologies
ATC Technology Concepts and Investment

Data

•Aircraft technology concepts

Figure 1.1. Aircraft and System Technologies Analytical Requirement: Preliminary Aircraft Design Model

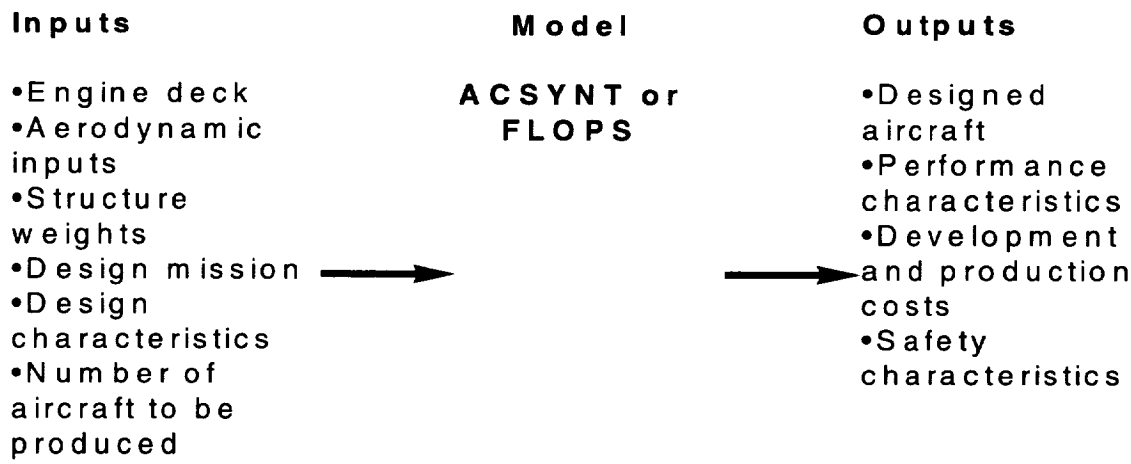


Figure1.1.1. Aircraft and System Technologies Analytical Requirement: System Delay Model

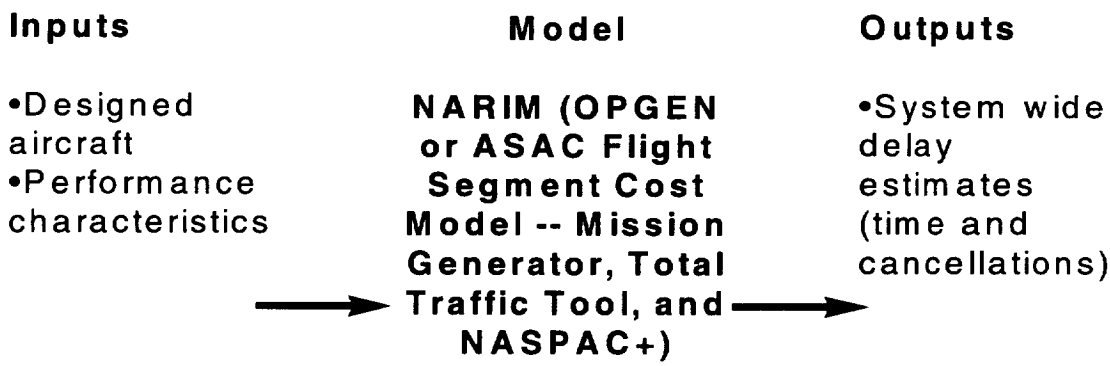


Figure 1.1.2. Aircraft and System Technologies Analytical Requirement: Mission Generator

Inputs	Model	Outputs
•Designed aircraft •Performance characteristics •City pairs •Latitude and longitude •Wind data	**ASAC Flight Segment Cost Model -- Mission Generator**	•Block fuel •Block time •Trajectory

Figure 1.1.2.1. Aircraft and System Technologies Analytical Requirement: Cost Translator

Inputs	Model	Outputs
•Block fuel •Block time •Trajectory	**COSTMOD or ASAC Flight Segment Cost Model -- Cost Translator**	•Block fuel •Block time •Flight segment cost

Figure 1.1.2.1.1. Aircraft and System Technologies Analytical Requirement: Air Carrier Cost Model

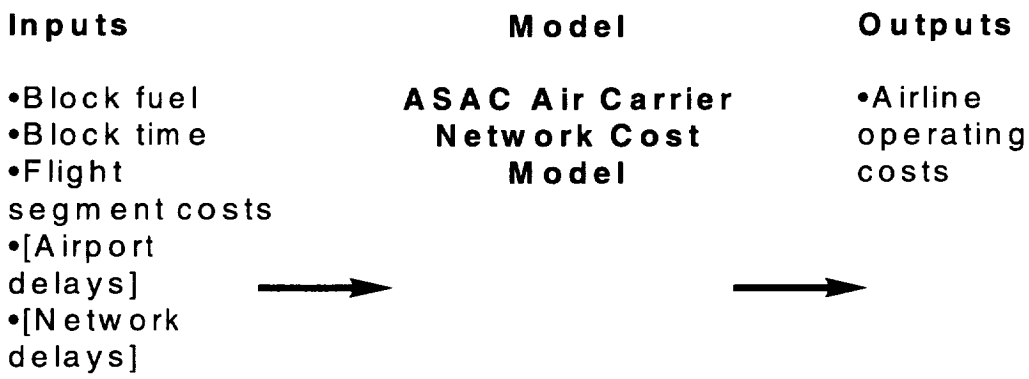


Figure 1.1.2.1.1.1. Aircraft and System Technologies Analytical Requirement: Air Carrier Economic Model

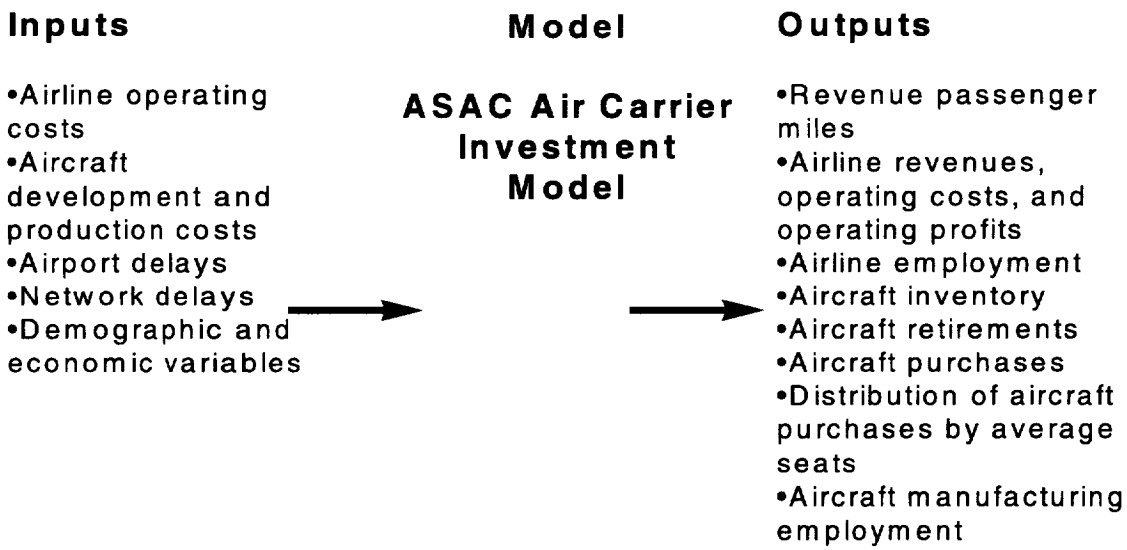


Figure 1.1.2.1.1.1.1. Aircraft and System Technologies Analytical Requirement: Aggregate Economic Model

Inputs	Model	Outputs
•Revenue passenger miles •Airline revenues, operating costs, and operating profits •Airline employment •Aircraft inventory •Aircraft retirements •Aircraft purchases •Distribution of aircraft purchases by average seats •Aircraft manufacturing employment •Forecasted aggregate air cargo demand by region •Change in aircraft inventory •Sales of air transportation services	**ASAC Aggregate Economic Model**	•Net economic impact from aviation technology •GDP •Net exports •Labor demand by industry and occupation •Intermediate demand for suppliers

Figure 1.2. Aircraft and System Technologies Analytical Requirement: Update Designs and Inputs and Inputs for Regional and Commuter Aircraft

Inputs	Model	Outputs
•Aircraft physical performance measures •Utilization profile	**STAT**	•Aircraft operating costs (DOC, IOC)

Figure 1.2.1. Aircraft and System Technologies Analytical Requirement: Regional and Commuter Network Cost Model

Inputs	Model	Outputs
•Aircraft operating costs (DOC, IOC) •Flight schedules →	**ASAC Regional and Commuter Network Cost Model**	→ •Regional and commuter airline operating costs

Figure 1.2.1.1. Aircraft and System Technologies Analytical Requirement: Regional and Commuter Economic

Inputs	Model	Outputs
•Regional and commuter airline operating costs •Demographic and economic variables →	**ASAC Regional and Commuter Economic Model**	→ •Forecasted aggregate air passenger travel demand by region •Aircraft inventory •Employment

Figure 1.3. Aircraft and System Technologies
Analytical Requirement: Air Cargo Economic Model

Inputs	Model	Outputs
•Economic variables	**Air Cargo Demand Model**	•Forecasted aggregate air cargo demand by region

Figure 2.0. FAA Air Traffic Management
ATC Technology Concepts and Investments

Data

•Change in ATM regulations
•Change in ATM automation and technology

Figure 2.1. Air Traffic Management
Analytical Requirement: NationalAirspace System Simulation

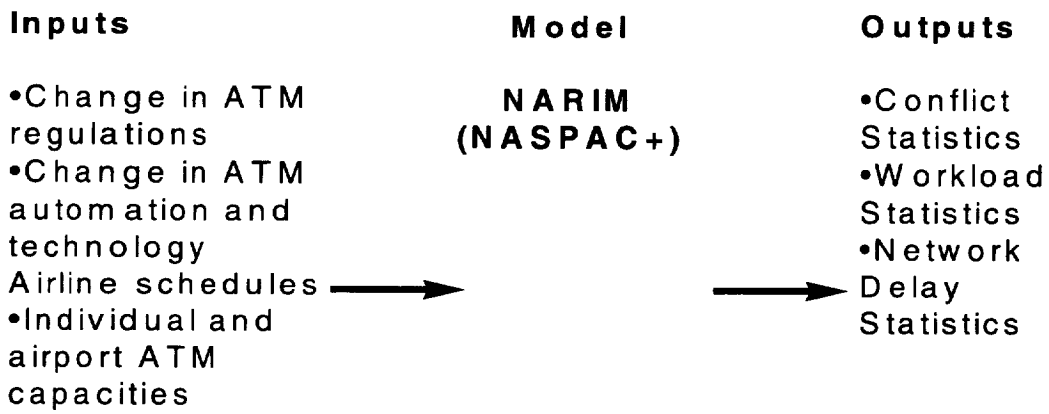


Figure 2.1.1. FAA Traffic Management Analytical
Requirement: Aircraft/ATC Cost-Benefit Model

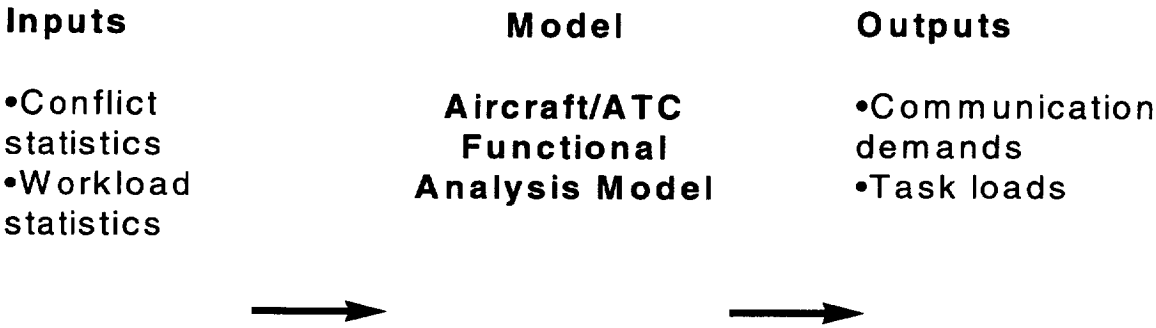


Figure 2.2. Air Traffic Management Analytical Requirement: Safety Analysis Model

Inputs	Model	Outputs
•Change in ATM regulations •Change in ATM automation and technology Safety design characteristics	**System Safety Tolerance Analysis Model**	•Measure of system safety

Figure 2.3. FAA Air Traffic Management Analytical Requirement: Airport Capacity Model

Inputs	Model	Outputs
•Change in ATM regulations •Change in ATM automation and technology •Runway usage •Number of miles before turn to destination (flight tracks)	**ASAC Airport Capacity Model**	•Airport capacity •Airport delay statistics

Figure 2.3.1. FAA Air Traffic Management Analytical Requirement: Network Delay Model

Inputs	Model	Outputs
•Airport capacity	**Approximate Network Delay (AND) Model or LMI Network Model**	•Network delay statistics

Figure 2.3.1.1. FAA Air Traffic Management Analytical Requirement: Network Delay Cost Model

Inputs	Model	Outputs
•Network delay statistics	**COSTMOD or ASAC Flight Segment Cost Model -- Cost Translation**	•Network delay costs

Figure 2.3.2. FAA Air Traffic Management Analytical Requirement: Airport Delay Model

Inputs	Model	Outputs
•Airport capacity	**ASAC Airport Delay Model**	•Airport delay statistics

Figure 2.3.2.1. FAA Air Traffic Management Analytical Requirement: Airport Delay Cost Model

Inputs	Model	Outputs
•Airport delay statistics	**COSTMOD or ASAC Flight Segment Cost Model -- Cost Translation**	•Airport delay costs

Figure 2.4. FAA Air Traffic Management Analytic Requirement: Runway Use Model

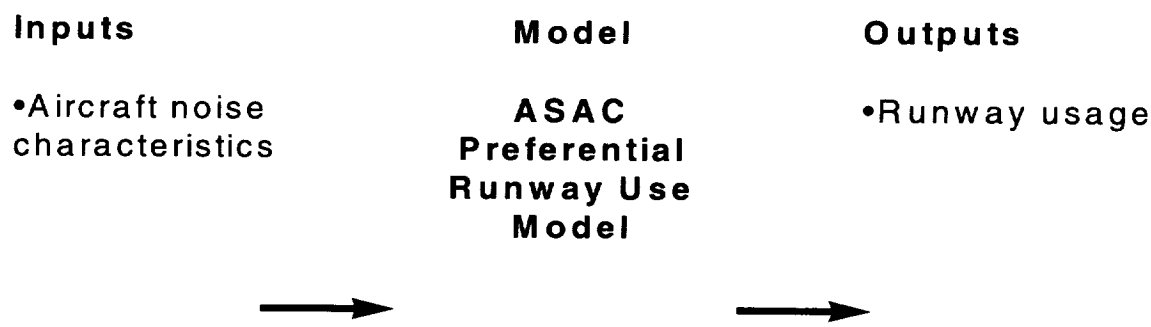


Figure 3.0. Environment

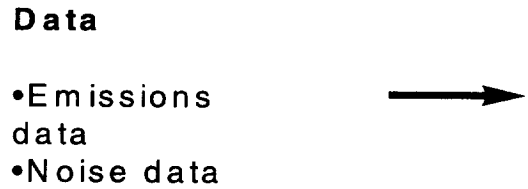


Figure 3.1. Environment–Emissions
Analytical Requirement: TBD

Inputs	Model	Outputs
•TBD	**Model TBD**	•Engine emissions data

Figure 3.1.1. Environment–Emissions
Analytical Requirement: Global Emissions Tracking

Inputs	Model	Outputs
•Engine emissions data •Flight schedules	**Global Aircraft Emissions Forecasting Model (GAEF)**	•Emissions produced enroute

Figure3.1.2. Environment–Emissions
Analytical Requirement: Emissions Dispersion Model

Inputs	Model	Outputs
•Engine emissions data •Flight schedules	**Emission and Dispersion Modeling System (EDMS)**	•Emissions produced around airports

Figure 3.2. Environment—Noise
Analytical Requirement: Noise Impact Model

Inputs	Model	Outputs
•Airport geography and demographics •Aircraft noise characteristics •Flight schedules •Runway usage	**Integrated Noise Model (INM)**	•Noise produced •Noise impact on surrounding community

Figure 3.2.1. Environment—Noise
Analytical Requirement: Noise Cost Model

Inputs	Model	Outputs
•Noise produced •Noise impact on surrounding area	**ASAC Air Carrier Flight Track Efficiency Model (AFTEM)**	•Time savings •Distance savings •Airline costs •Number of miles before turn to destination (flight tracks)

Figure 4.0. General Aviation
Analytical Requirement: General Aviation Economic Model

Inputs	Model	Outputs
•Cost sensitivities •Competitive pressures •Manufacturing analyses	**ASAC General Aviation Economic Model**	•Demand for general aviation aircraft

Appendix D
Client/Server architecture

This section describes what we believe to be the architectural style that will be used in the EA design.

WHAT IS CLIENT/SERVER ARCHITECTURE?

There are various definitions of client/server systems. Ken Avellino[1] of Learning Tree International defines a client/server system as a "distributed cooperative processing environment that provides a single system image to the user." Basically, client/server architecture allows a client to request services from a server.

WHAT IS A CLIENT?

A client is a customer or the calling module. Most people think of their personal computers and workstations as clients. Clients are not defined by hardware but by the function performed. A client can be a workstation requesting services from a LAN or a server requesting services from another server.

WHAT IS A SERVER?

A server provides services to a client. It is the service called by a client. As with clients, servers are not defined by hardware, but by the function performed. A workstation can call a service from a mainframe or other large computer, a server computer, another workstation, a personal computer, or another process on itself.

[1] Avellino, Ken. "Introduction to Client/Server Computing."

APPLICATION COMPONENTS

Most applications are composed of three types of components:

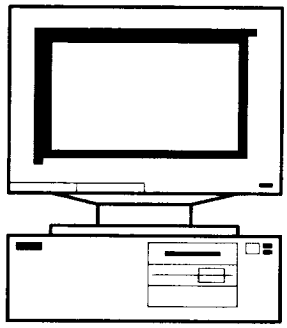
- Presentation
- Business
- Data.

The presentation component contains the logic that presents information to an external source and obtains information from that source. The business component contains the application logic that controls the business functions and processes performed by the application. The business functions usually manipulate data. Typically, business logic applicable to a single user is located on the client and to multiple users on the server. The data component contains the logic that interfaces with the data repository system. Client/server architectures are labeled by how they distribute the presentation, business, and data components.

ONE-TIER CLIENT/SERVER ARCHITECTURE

In one-tier client/server architecture, the presentation, business, and data logic are combined into a tightly integrated executable that runs on a single machine. The data files are resident on the same machine. An example of one-tier client/server architecture is an application such as a word processor running on a stand-alone personal computer.

Figure D-1. One-tier Client/Server Architecture

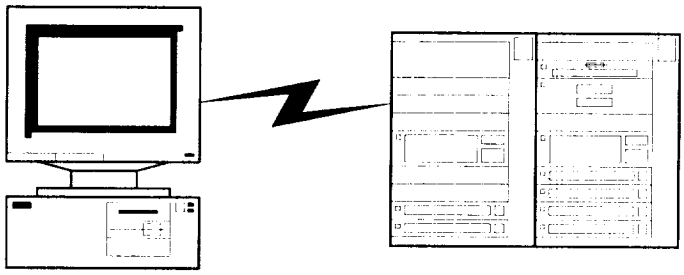


Applications on a one-tier client/server architecture are easy to manage, control, and secure. They are, however, neither flexible, easily scaled, nor ported.

TWO-TIER CLIENT/SERVER ARCHITECTURE

In two-tier client/server architecture, the application is divided into two parts, and the processing is divided between two machines. In this case, it is possible to separate the presentation logic from the application and data logic. An example of two-tier client/server architecture is a workstation that runs presentation logic for a word processor while the actual application and associated data are stored on a common server.

Figure D-2. Two-tier Client/Server Architecture

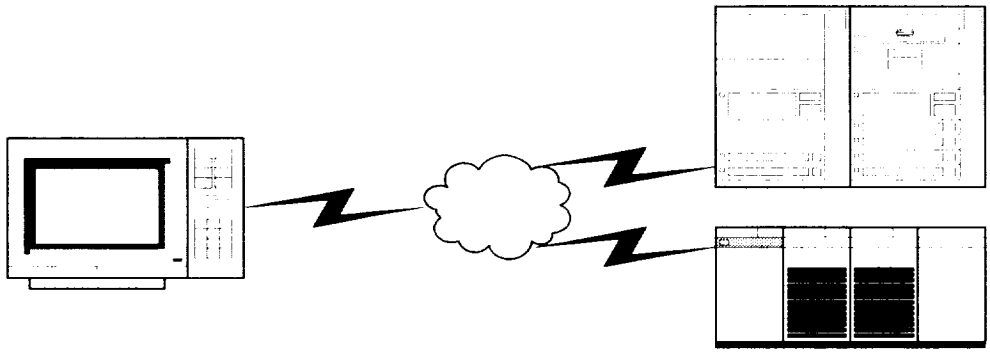


Two-tier client/server architectures are effective for small workgroups. Growth to a larger implementation may lead to performance problems and partitioning difficulty.

THREE-TIER CLIENT/SERVER ARCHITECTURE

In three-tier client/server architecture, the application is divided into three parts. The processing can be divided among three or more machines. The presentation logic is usually completely separate from both the business and data logic, which allows for better platform independence and greater application adaptability. An example of three-tier client/server architecture is an ATM machine that runs presentation logic, requesting services from a server, which runs application logic and provides distributed computing services, which in turn requests data from a bank database server.

Figure D-3. Three-tier Client/Server Architecture



When the processing is divided among more than three machines, it is n-tier or multi-tier client/server architecture.

Three-tier client/server architecture offers better performance and flexibility than two-tier client/server architecture. Application partitioning in multi-tier client/server architectures provides greater reliability and security. Although these architectures are more complex and, therefore, more difficult to design, they are often easier to implement and maintain.

There are many types of three-tier client/server architecture. The following architectures are common, three-tier environments:

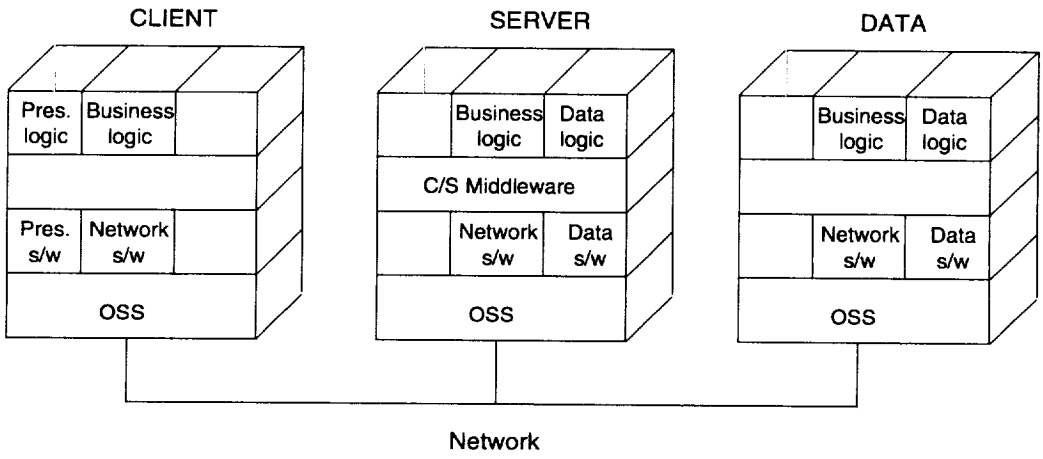
- Transaction processing (TP) monitor lite
- TP monitor heavy

- Messaging server
- Application server
- Object database management system (ODBMS)
- Object request broker (ORB).

An emerging architecture, three-tier client server and the Internet, will shortly be added to the above list.

One common representation of a three-tier client/server architecture is the Open Distributed Application Model (ODAM) from IBM.[2]

Figure D-4. Open Distributed Application Model



Note that presentation logic and software are present on the client. This is the user application. Also, middleware is located on the server, and data logic is present on both the server and data portions of the model.

WHAT IS MIDDLEWARE?

Middleware ties the different components of a multi-tier architecture together. It allows two or more systems to establish a transparent link to exchange data and services. Middleware includes services such as communication, data access and

[2] Open Enterprise Group at IBM Europe. "A Guide to Open Client/Server."

connectivity, scheduling, security, transaction management, and system management.

CLIENT/SERVER DESIGN

Multi-tier client/server architectures are flexible and modular, and they can be combined to satisfy almost any computing need. The ASAC design could use a multi-tier client/server architecture, which could be a combination of the many types of three-tier client server architecture listed above. The design, which could be produced in a follow-on effort, would provide the following benefits:

- Support rapid development and deployment of new application systems
- Support rapid modification of existing application systems
- Support the integration of new and existing application systems
- Support current and emerging technologies and standards
- Encourage reuse of software
- Support dynamic reconfiguration of systems for scalability or networking requirements
- Allow for a choice of GUI programming languages and data sources.

Appendix E
Preliminary ASAC EA Design

This section presents the preliminary design of the ASAC EA. The EA architecture defined in this document lends itself to a client/server design. Five simple data/control flow diagrams depict a preliminary ASAC EA design. They are:

- EA High-level Design
- User Application high-level design
- Analysis Application high-level design
- Model Application high-level design
- Driver Application high-level design.

Ovals represent components that belong to the data/control flow diagram named in the figure title; rectangles represent components that belong to a different data/control flow diagram. Cylinders represent a data repository.

Two additional diagrams have been created:

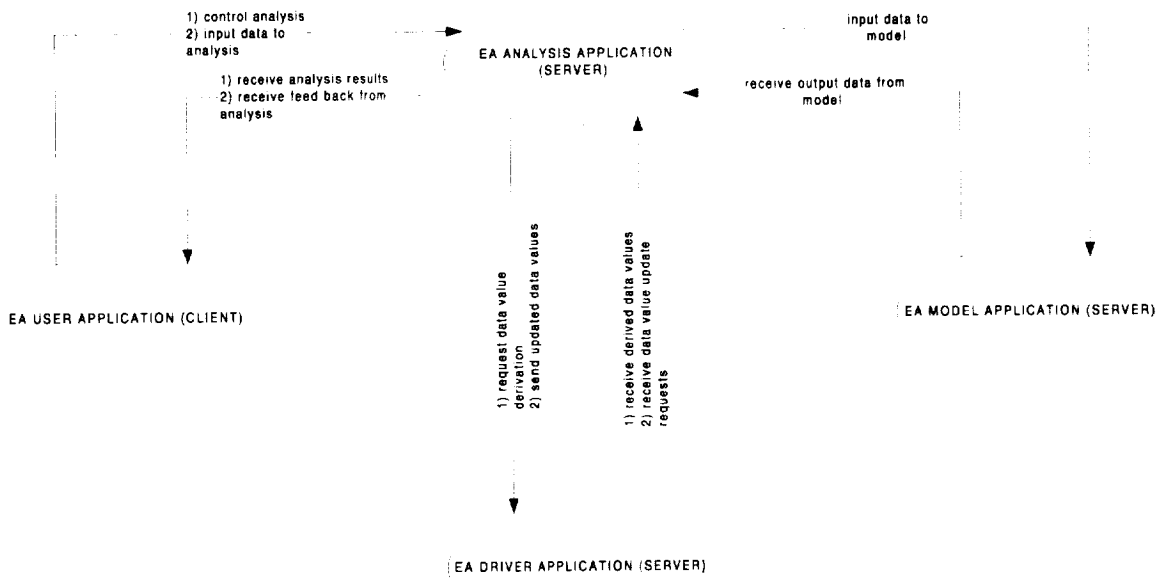
- High-level block diagram
- Entity-relationship diagram.

The high-level block diagram shows another view of the preliminary ASAC EA design. The entity-relationship diagram shows a preliminary look at the ASAC EA database design for storing items such as user ids, analysis templates, analysis history, model and driver configurations, and dependencies.

DATA/CONTROL FLOW DIAGRAMS

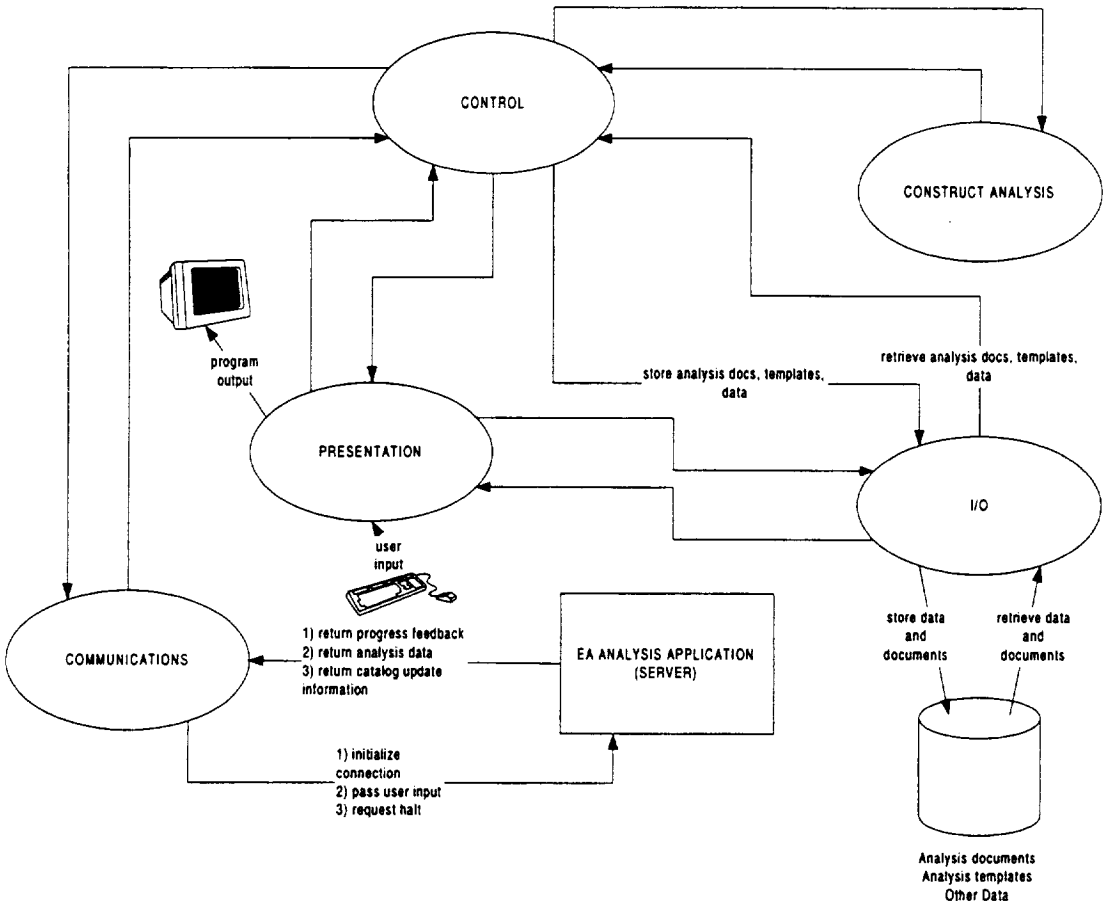
EA High-level Design

Figure E-1. EA High-Level Design



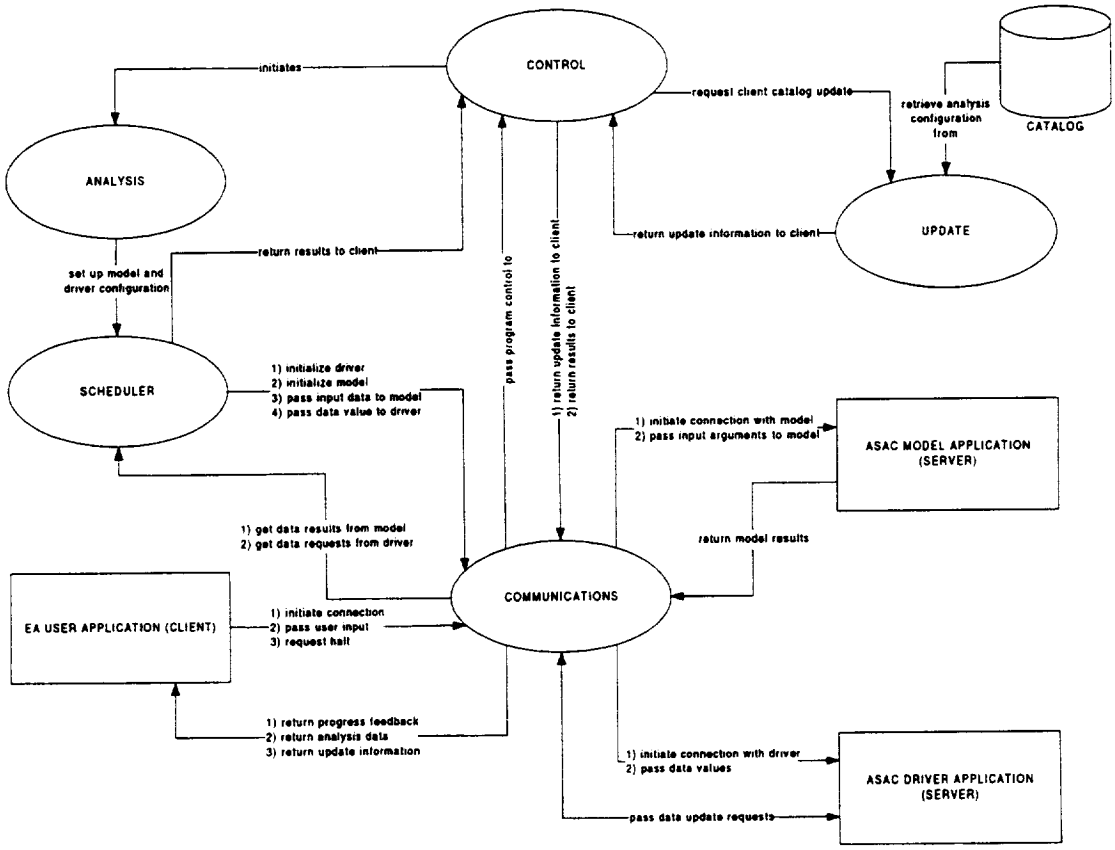
User Application High-Level Design

Figure E-2. User Application High-Level Design



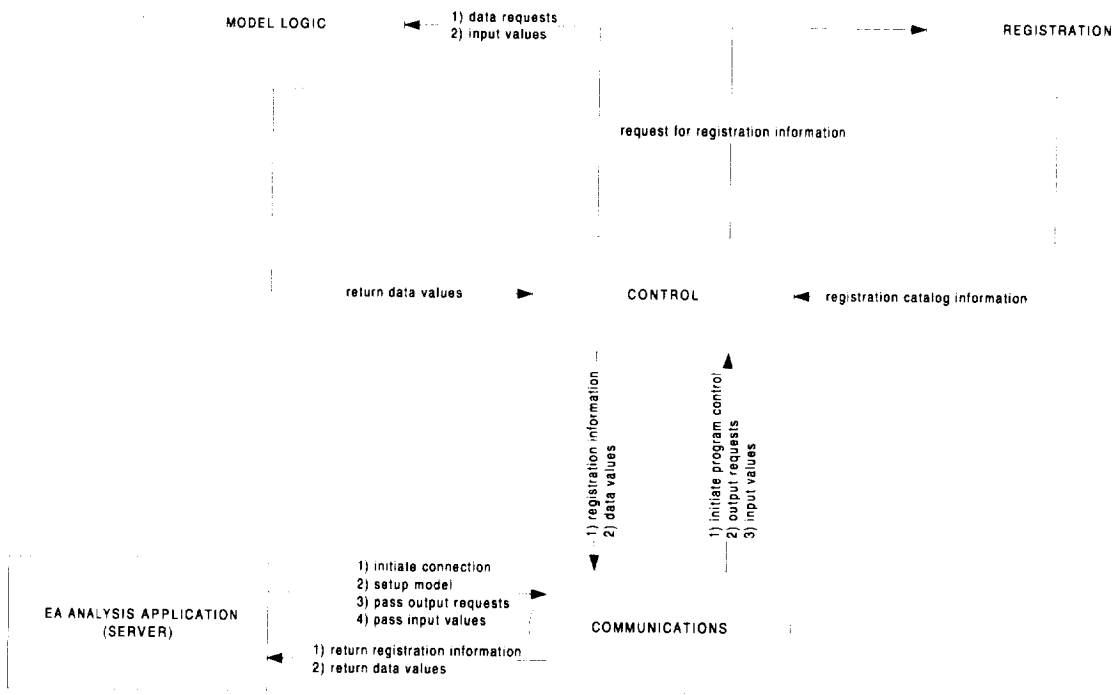
Analysis Application High-Level Design (Server)

Figure E-3. Analysis Application High-Level Design (Server)



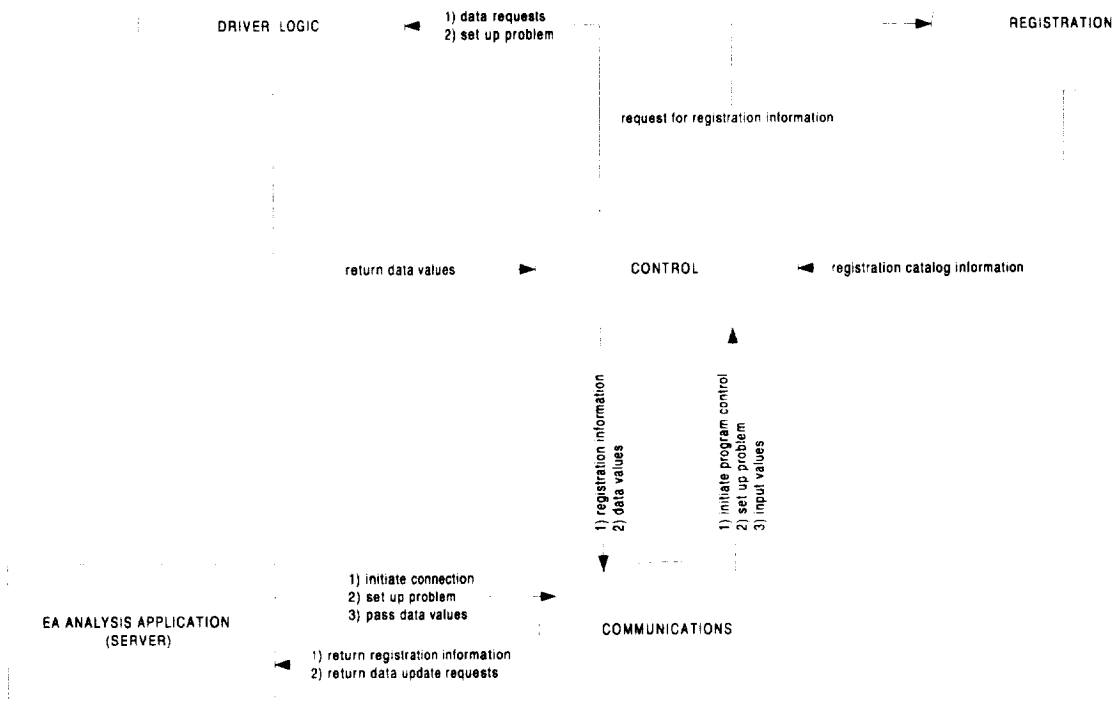
Model Application High-Level Design (Server)

Figure E-4. Model Application High-Level Design (Server)



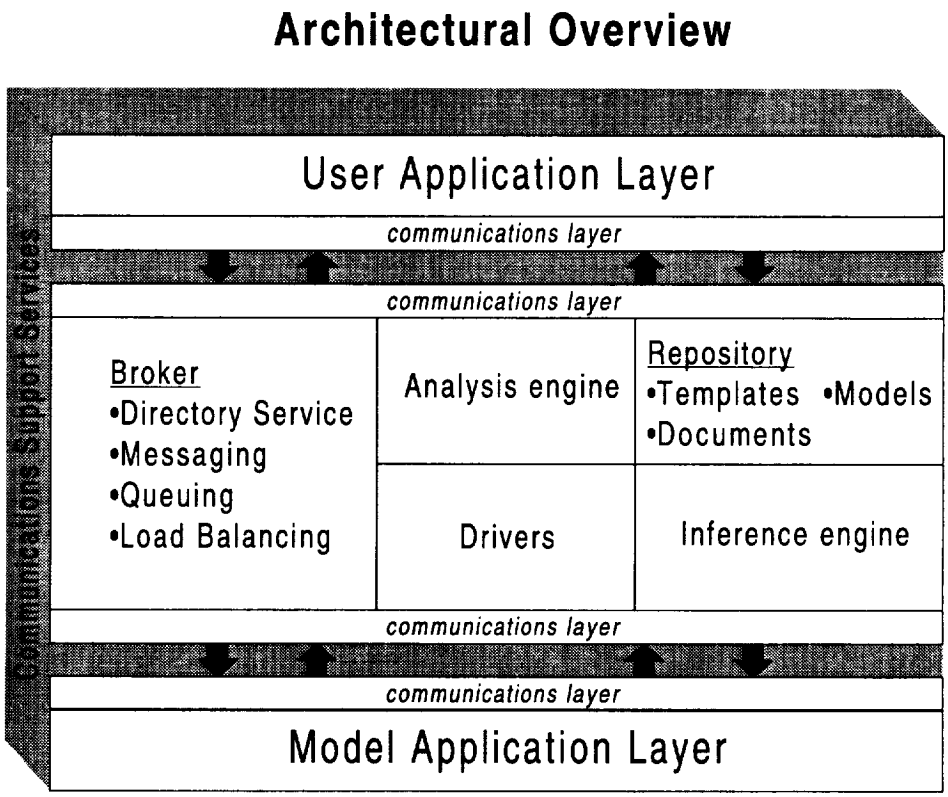
Driver Application High-Level Design (Server)

Figure E-5. Driver Application High-Level Design (Server)

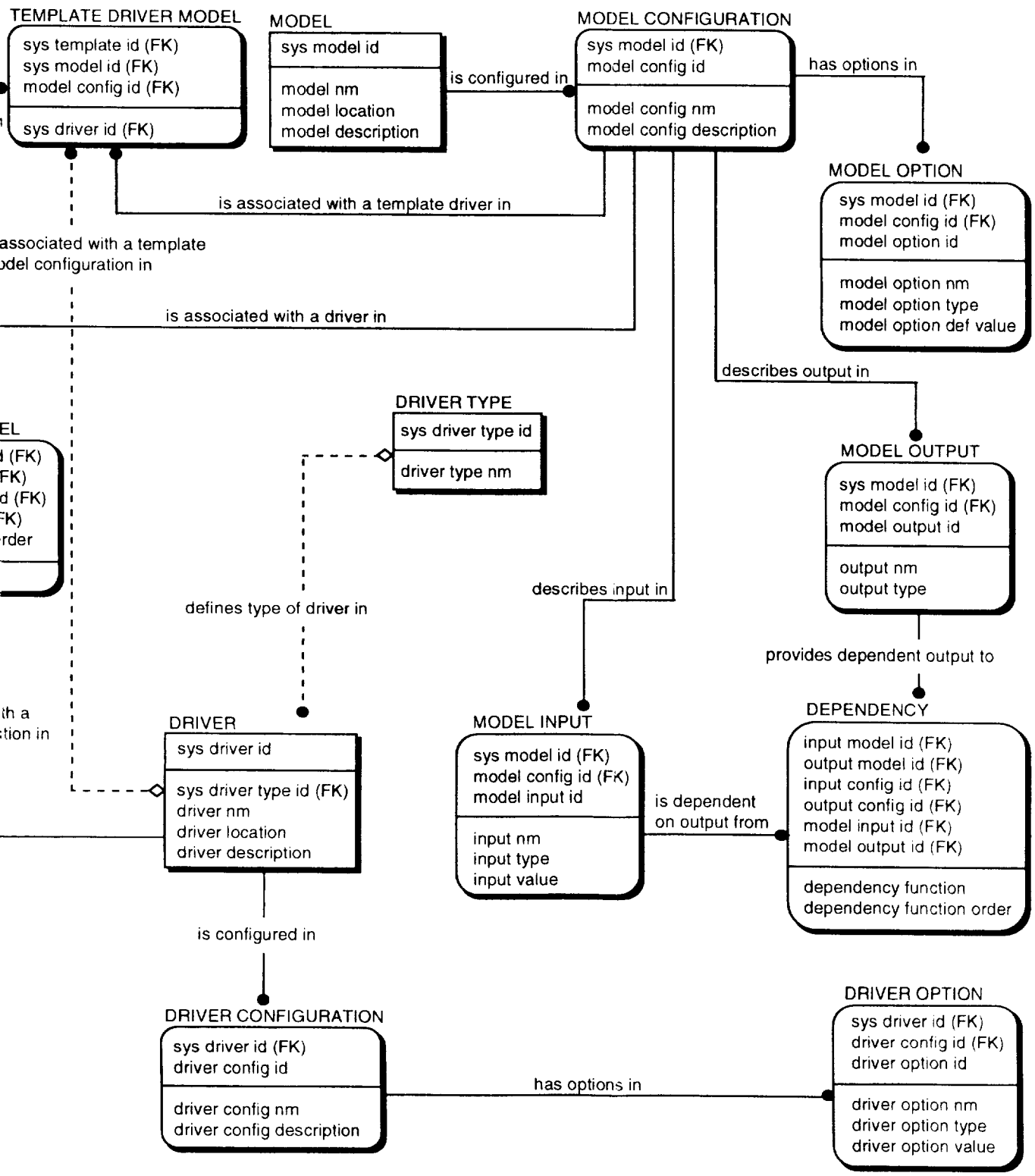


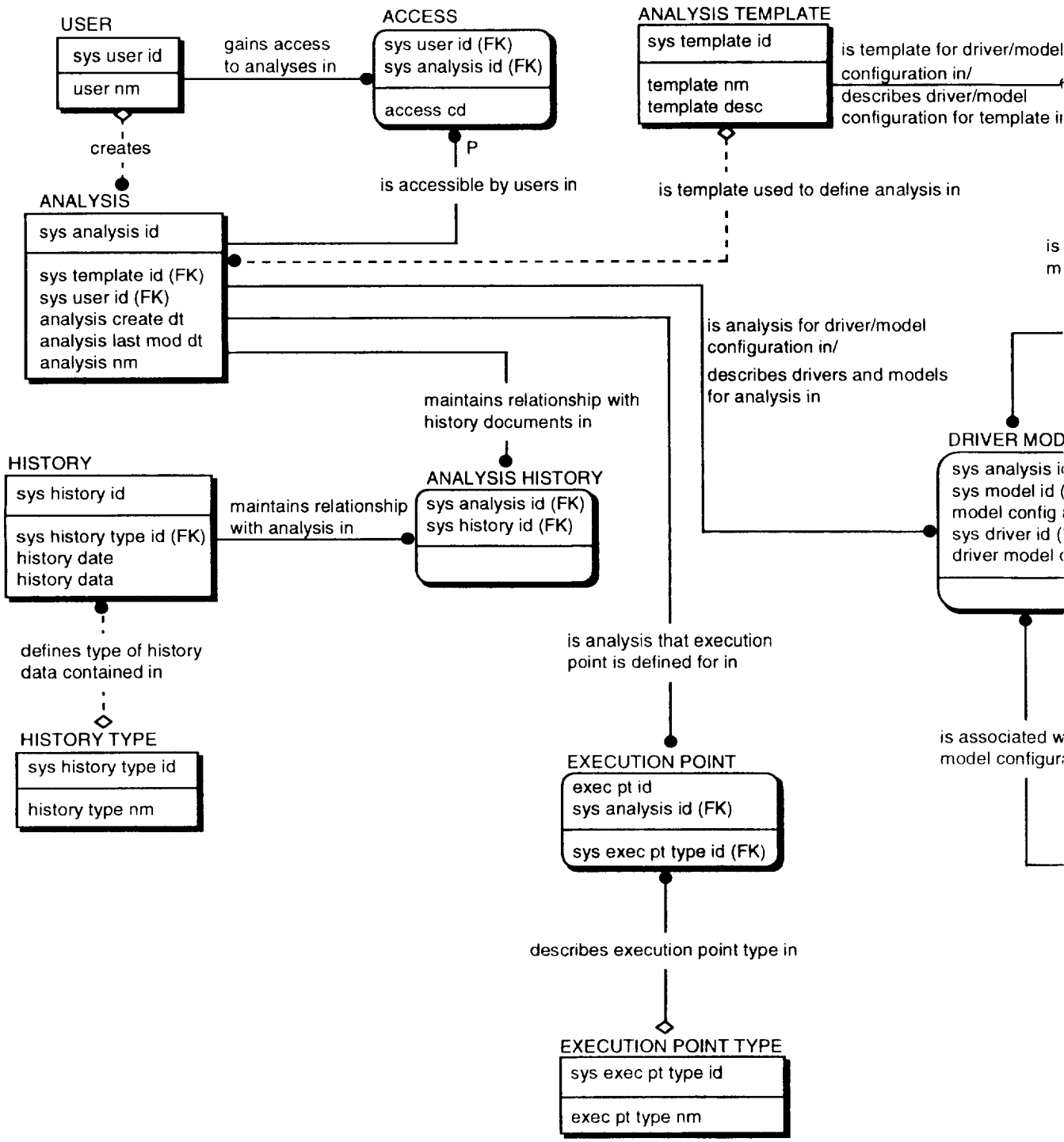
Block Diagram

Figure E-6. High-Level Block Diagram



Database Entity-Relationship Diagram


TEMPLATE DRIVER MODEL
sys template id (FK)
sys model id (FK)
model config id (FK)
sys driver id (FK)
MODEL
sys model id
model nm
model location
model description
is configured in
MODEL CONFIGURATION
sys model id (FK)
model config id
model config nm
model config description
has options in
MODEL OPTION
sys model id (FK)
model config id (FK)
model option id
model option nm
model option type
model option def value
is associated with a template driver in
associated with a template
odel configuration in
is associated with a driver in
describes output in
DRIVER TYPE
sys driver type id
driver type nm
EL
d (FK)
FK)
d (FK)
FK)
rder
MODEL OUTPUT
sys model id (FK)
model config id (FK)
model output id
output nm
output type
defines type of driver in
describes input in
provides dependent output to
th a
tion in
DRIVER
sys driver id
sys driver type id (FK)
driver nm
driver location
driver description
MODEL INPUT
sys model id (FK)
model config id (FK)
model input id
input nm
input type
input value
is dependent
on output from
DEPENDENCY
input model id (FK)
output model id (FK)
input config id (FK)
output config id (FK)
model input id (FK)
model output id (FK)
dependency function
dependency function order
is configured in
DRIVER CONFIGURATION
sys driver id (FK)
driver config id
driver config nm
driver config description
has options in
DRIVER OPTION
sys driver id (FK)
driver config id (FK)
driver option id
driver option nm
driver option type
driver option value


USER
sys user id
user nm
gains access to analyses in
ACCESS
sys user id (FK)
sys analysis id (FK)
access cd
ANALYSIS TEMPLATE
sys template id
template nm
template desc
is template for driver/model configuration in/
describes driver/model configuration for template i
creates
P
is accessible by users in
is template used to define analysis in
ANALYSIS
sys analysis id
sys template id (FK)
sys user id (FK)
analysis create dt
analysis last mod dt
analysis nm
is analysis for driver/model configuration in/
describes drivers and models for analysis in
maintains relationship with history documents in
HISTORY
sys history id
sys history type id (FK)
history date
history data
maintains relationship with analysis in
ANALYSIS HISTORY
sys analysis id (FK)
sys history id (FK)
DRIVER MOD
sys analysis i
sys model id (
model config
sys driver id (
driver model
defines type of history data contained in
is analysis that execution point is defined for in
HISTORY TYPE
sys history type id
history type nm
EXECUTION POINT
exec pt id
sys analysis id (FK)
sys exec pt type id (FK)
is associated w
model configur
describes execution point type in
EXECUTION POINT TYPE
sys exec pt type id
exec pt type nm

REPORT DOCUMENTATION PAGE

Form Approved
OPM No.0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources gathering, and maintaining the data needed, and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Information and Regulatory Affairs, Office of Management and Budget, Washington, DC 20503.

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 1997	Contractor Report

4. TITLE AND SUBTITLE

ASAC Executive Assistant Architecture Description Summary

5. FUNDING NUMBERS

C NAS2-14361

WU 538-08-11-01

6. AUTHOR(S)

Eileen Roberts, James A. Villani

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

Logistics Management Institute
2000 Corporate Ridge
McLean, VA 22102-7805

8. PERFORMING ORGANIZATION REPORT NUMBER

LMI- NS601T2

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

National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23681-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA CR-201681

11. SUPPLEMENTARY NOTES

Langley Technical Monitor. Robert Yackovetsky
Final Report

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified-Unlimited
Subject Category 01
Availability: NASA CASI, (301) 621-0390

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

In this technical document, we describe the system architecture developed for the Aviation System Analysis Capability (ASAC) Executive Assistant (EA). We describe the genesis and role of the ASAC system, discuss the objectives of the ASAC system and provide an overview of components and models within the ASAC system, discuss our choice for an architecture methodology, the Domain Specific Software Architecture (DSSA), and the DSSA approach to developing a system architecture, and describe the development process and the results of the ASAC EA system architecture. The document has six appendices.

14. SUBJECT TERMS

ASAC, NASA, Architecture, Executive Assistant

15. NUMBER OF PAGES

191

16. PRICE CODE

A09

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