

FLIGHT TRANSPORTATION LABORATORY REPORT R87-5

AN AIR TRAFFIC CONTROL SIMULATOR FOR THE EVALUATION OF FLOW
MANAGEMENT STRATEGIES

by

JAMES FRANKLIN BUTLER

MASTER OF SCIENCE
IN AERONAUTICS AND ASTRONAUTICS

FLIGHT TRANSPORTATION LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS 02139
FEBRUARY, 1987

FTL COPY, DON'T REMOVE
33-412, MIT 02139

AN AIR TRAFFIC CONTROL SIMULATOR FOR THE EVALUATION OF
FLOW MANAGEMENT STRATEGIES

by

JAMES FRANKLIN BUTLER

Abstract

An air traffic control (ATC) simulator for the evaluation of flow management strategies was designed and implemented in Lisp. The ATC system is modeled as a network of queueing systems. Flights are generated from a flight schedule, a text file created by the simulation scenario designer. The simulator supports the following flow control actions: ground holds, airborne holds, speed control, and rerouting. Tools are included for schedule generation and statistical analysis. The design of a scenario requires a working knowledge of Lisp.

This thesis includes an overview of the flow management problem, a discussion of modeling issues and design considerations, a description of the implementation of the simulator in moderate detail, results from a sample simulation scenario which includes a simple flow management strategy, and suggestions for possible improvements to the simulator.

Acknowledgements

I would like to express my sincere gratitude and appreciation to the following people: Professor Amedeo Odoni, my thesis advisor, for his suggestions and guidance; Dr. John Pararas, for all of his help and for coding several modules which went into the simulator; Dr. Antonio Elias, for suggesting this project and aiding me in the early stages of this work; Flight Transportation Lab graduate students Ronnie LaJoie, Mark Kolb, Lyman Hazelton, and Tom Trikas, for their assistance; Bob Bruen, for keeping the VAX system in good working order; Professors Abelson and Sussman, for teaching a wonderful course on the structure and interpretation of computer programs; and finally, my friends and family for their support.

I would also like to thank the NASA/FAA Tri-University Program for funding this research.

Contents

Abstract	ii
Acknowledgements	iii
1 Introduction	1
1.1 What Flow Management Is and Why It Is Sometimes Desirable	1
1.2 Previous Research Concerning Flow Management Strategies	2
1.3 The Simulator	2
1.4 Outline	3
2 Modeling and Design Considerations	4
2.1 The Flow Management Problem	5
2.2 Aircraft	8
2.2.1 Individual Aircraft	8
2.2.2 Aircraft Dynamics	9
2.3 The ATC System	9
2.4 Flights	10
2.4.1 Flight Restrictions	10
2.4.2 Unscheduled Flights	11
2.4.3 The Effect of Arrival Delays at an Airport on Departure Delays	12
2.5 Flow Control	13
2.6 Performance Measures	14
3 The Air Traffic Control Simulator and Its Use	16
3.1 An Overview of the Simulator	16
3.2 The Simulator Objects	17
3.2.1 Queueing Systems	18
3.2.2 Network Objects	20
3.2.3 Aircraft and Related Objects	25
3.2.4 Other Objects	28
3.3 Flights, Flight Schedules, and Random Schedule Generation	30
3.3.1 The Flight Schedule	30
3.3.2 Random Schedule Generation	32
3.4 The Implementation of Flow Management Strategies	33

3.4.1	Arrival Demand Management	34
3.4.2	Arrival Metering	35
3.4.3	Strategies for Reducing Airspace Congestion	36
3.5	Data Collection and Analysis	37
3.5.1	Data Collection	37
3.5.2	Tools for Statistical Analysis	37
4	An Example Simulation Scenario Design	39
4.1	The Flow Management Strategy	39
4.2	The ATC System Network	45
4.3	Schedule Generation	46
4.4	Data Collection	49
4.5	Results of the Simulation Runs	49
5	Conclusions and Recommendations for Future Work	53

Chapter 1

Introduction

1.1 What Flow Management Is and Why It Is Sometimes Desirable

The high air traffic densities in some areas of the world cause congestion in certain terminal areas and en route sectors during peak traffic periods. This congestion makes the air traffic controllers' task of ensuring the safe, efficient flow of air traffic more difficult, and causes costly airborne and ground delays of aircraft and their passengers. Flow control refers to the control of traffic so as to avoid unacceptable traffic densities, minimize the cost of delays, and hopefully attain efficient traffic flow as well.

One common problem in the United States and Europe is that at selected airports, the number of desired landings during peak periods may exceed the number of landings that the airport can safely accomplish. Arrival Flow Management is the task of matching the arrival traffic demand at a destination airport to the capacity of the airport, and involves two related tasks: Arrival Demand Management, in which demand and capacity are predicted well in advance and any unavoidable delays are taken on the ground before departure; and Arrival Metering, in which the actual demand is continuously monitored and delays which must be taken in the air are taken at high altitudes rather than in terminal areas, where congestion is higher and aircraft are much less fuel-efficient due to flying at low altitudes.

At times, en route sectors may become congested as well. Sector congestion may be avoided through departure metering or the rerouting of aircraft, as appropriate.

In 1969, a Central Flow Control Facility (CFCF) and Local Flow Control (LFC) positions

in en route control facilities were established in the United States. The CFCF is currently responsible for performing Arrival Demand Management, and LFC positions perform Arrival Metering [1].

1.2 Previous Research Concerning Flow Management Strategies

Surprisingly, there has been very little research done in the area of flow management strategies (specifically concerning the tradeoffs between ground delays and air delays) considering the potential for delay reduction, cost reduction, and possibly increased safety levels due to reduced congestion. To our knowledge there are only three published papers in this area: one by Dr. Balraj Sakkappa of MITRE, and two by Professors Giovanni Andreatta and Giorgio Romanin-Jacur of the University of Padova.

Sakkappa's paper [1] proposes the discounting of ground delays issued by flow control, so as to reduce the amount of wasted airport capacity. The paper also discusses Arrival Metering.

The two papers by Andreatta and Romanin-Jacur [2,3] are very analytical. They present an algorithm to solve a much simplified flow management problem in which a known group of aircraft will desire landings at a destination airport during a single time period and the airport capacity during that time period is a known random variable, while in subsequent periods the airport capacity is infinite.

1.3 The Simulator

Because of the analytical complexity of the overall flow management problem, we believe that a simulator is needed as a way of evaluating alternative flow management strategies. Such a simulator was built for this thesis and is described herein.

The design goals for the simulator were the following:

- The simulator should be able to model a portion of the ATC system at many levels of detail.

- The simulator should be designed specifically for the evaluation of a wide variety of flow management strategies; in particular, the simulator should allow the following flow control actions: ground holds, speed control, airborne holds, and rerouting.
- The collection of statistics on system performance should not be difficult.
- Above all, the simulator should be extremely flexible and easy to modify.

1.4 Outline

Chapter 2, entitled “Modeling and Design Considerations”, first describes the flow management problem, then discusses the basis on which it was decided to model certain characteristics of the ATC system and how the complexity of the system affected these modeling decisions.

Chapter 3, entitled “The Air Traffic Control Simulator and Its Use”, describes the simulator and its associated tools in moderate detail.

Chapter 4, entitled “An Example Simulation Scenario Design”, describes how a model of a small portion of the ATC system and a simple Arrival Demand Management strategy were implemented, and presents the results obtained from simulation runs with and without flow control.

Chapter 5, entitled “Conclusions and Recommendations for Future Work”, discusses the strengths and weaknesses of the simulator in regards to the evaluation of flow management strategies, and presents opinions concerning what features and tools should be added to improve the simulator.

Chapter 2

Modeling and Design Considerations

A simulator for the evaluation of flow management strategies was developed on a Lisp Machine. The simulator has the following general characteristics:

- The ATC system is modeled using a network of queueing systems as specified by the simulation scenario designer.
- Individual aircraft with very simple flight dynamics travel through the network. The aircraft are limited to two speed levels (cruise and minimum consumption) and three altitude levels (ground, low, and high).
- Four types of flow control actions can be applied to aircraft: ground holds, speed control, airborne holds, and rerouting.
- Flights are generated from a flight schedule, which is a text file. Flights must start at an airport and end at another airport, with no intermediate stops.
- The simulator includes tools to aid the simulation scenario designer in creating flight schedules and compiling statistics, but at present there are no tools to aid in the implementation of flow management strategies.

This chapter first describes the flow management problem, then discusses how modeling issues and design considerations affected several important characteristics of the simulator.

2.1 The Flow Management Problem

The flow management problem is the task of controlling the flow of aircraft through the ATC system so as to satisfy some goal, such as minimizing the value of some cost function. The possible solutions of the flow management problem are flow management strategies, which specify certain control actions for aircraft based on the current and predicted future states of the ATC system. These control actions include ground holds, airborne holds, speed control, and rerouting, all of which are allowed (with certain restrictions) by the simulator described in this thesis. The flow control is done by one or more flow control elements, which can be distributed throughout the ATC system. These flow control elements may be able to communicate with one another.

The reduction of congestion in the ATC system is a goal of all good flow management strategies, since congestion usually causes delays. Congestion is most costly when it causes aircraft to be delayed at low altitudes (such as near airports) where aircraft are not very fuel-efficient.

The effectiveness of all flow management strategies is limited by the availability and location of information concerning the current and future states of the ATC system. Because some information may only be available in certain areas within the ATC system, it is often desirable to distribute geographically the flow control elements. In the United States, flow control responsibility is divided between the Central Flow Control Facility and Local Flow Control positions in en route control facilities (see Chapter 1). When a flow management strategy is formulated, the following must be considered: what types of information are available, where that information is available, the accuracy and timeliness of that information, and constraints on communication between flow control elements. It may not be useful to do certain types of flow control if the information necessary to make reasonable flow control decisions is not available to the appropriate flow control elements.

As mentioned in the introduction, selected airports have a demand for landings which exceeds the number of landings the airport can accommodate (the airport's capacity) during certain parts of a typical day. Let us assume that airport D (see Figure 2.1) expects more

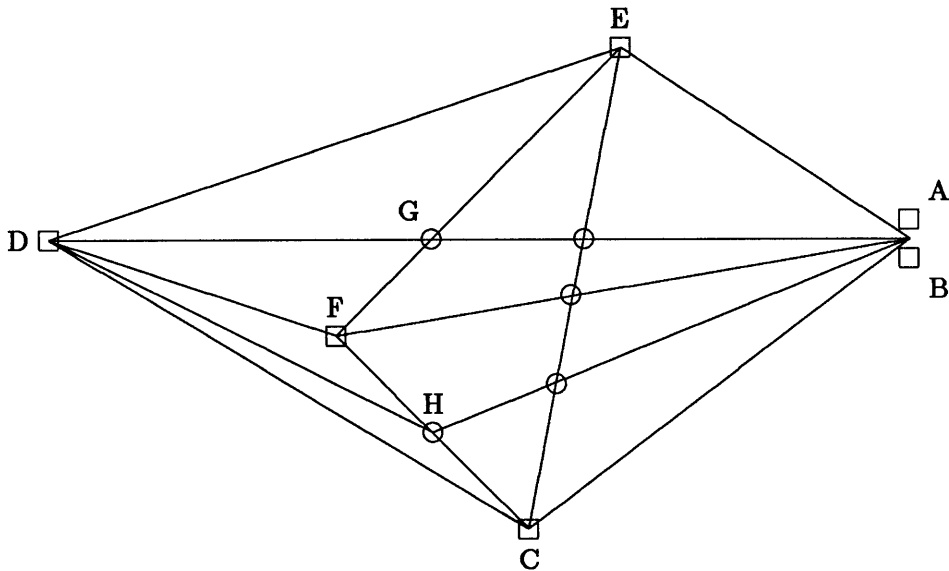


Figure 2.1: This is a network which could represent part of an ATC system; squares are used to represent airports (A through F), circles represent airway intersections (G and H), and lines between airports and intersections represent airways. Airports A and B are close enough to each other to share a terminal area.

landings during a certain part of the day than it can accommodate. A flow management strategy which attempts to avoid congestion in the terminal area surrounding airport D might meter departures headed for D by issuing ground holds at the flight origins so as to distribute the arrivals at D over a longer period of time and cause unavoidable delays to be taken on the ground; this is called Arrival Demand Management. Here are some complications that should be considered when such a strategy is formulated:

- The number of flights arriving at D during the time period in which demand is expected to exceed capacity is uncertain.
- The capacity of airport D during the same time period may also be uncertain, partially due to uncertainty in weather forecasts.
- The flight time from A to D is much longer than the flight time from F to D, so the departure metering of flights from A to D must be done long before the departure

metering of flights from F to D, at which time the capacity and demand can be estimated more accurately.

- A similar complication exists for slower air traffic, which will request earlier departures than jets with the same expected arrival time.

If it might be desirable to meter flights departing from A, then planning must be done before any of those flights depart, which may be several hours before the flights are to arrive at D. One possible strategy would be to only meter flights departing from airports which are not very far away, since capacity and demand can be better estimated at that time. However, the discrimination based on travel distance may be unacceptable for non-technical reasons (the short range flights would take all of the ground holds). Aircraft may also be metered while airborne when they get reasonably close to D by commanding speed reductions or airborne holds; this is called Arrival Metering. If done well, Arrival Metering can even further reduce airborne delays in D's terminal area which are costly in terms of fuel consumption.

It is also possible to have congestion in en route sectors. Such congestion is sometimes caused when terminal areas become very congested with aircraft desiring landings, which may cause the air traffic to back up into neighboring sectors. When busy airports are located close to each other (for example, A and B in Figure 2.1), the terminal area and neighboring en route sectors may become congested during periods when there are many arrivals at or departures from the airports. Strategies to reduce that congestion might again involve departure metering (ground holds). Sectors containing crossing airways can also become congested. For example, the sector containing node G (in Figure 2.1) may become congested if traffic between A and D and between F and E is heavy. Some of that congestion can be eliminated using speed control, but rerouting may also be necessary. Traffic between A and D can be routed through node H instead of node G if traffic between F and C is not too heavy.

It is quite obvious that a strategy may give good results when evaluated with one cost function but may not be very good when other cost functions are used. One simple system cost function is the average flight delay for all aircraft. A good flow management strategy for

that cost function would not give many ground holds, since a minute of air delay costs the same as a minute of ground delay. A more reasonable cost function is a weighted average of air delay and ground delay, with air delay weighted more heavily. This reflects the fact that it is cheaper and safer to keep an aircraft on the ground than it is to fly it. A good strategy for this cost function might issue a large number of ground holds, since a minute of air delay may be equivalent to several minutes of ground delay (depending on the weighting).

There is also a distinction between static strategies and dynamic strategies. In a static strategy, a daily plan is created and that plan is adhered to no matter what happens during the day. Dynamic strategies have some underlying structure, but certain parameters of the strategy may change based on changes in the state of the ATC system. Note that almost all worthwhile strategies are at least partially dynamic; for example, any good strategy which does en route metering must be dynamic. However, static strategies may be effective if they are based on excellent predictions about the traffic that will materialize and airport capacities. A static strategy that has been implemented on our simulator is described in Section 4.1.

Because of the uncertainty in airport and sector capacity predictions and inaccuracies in flight information, good flow management strategies will be dynamic; their flow control actions will be based on a priori estimates of the accuracy of their information and will be able to react quickly as better information becomes available. Hopefully, the simulator described in this thesis will be helpful in accurately evaluating such strategies.

We shall now discuss the rationale for a number of rather critical choices made in the design of the simulator. Although certain characteristics of the ATC system are not modeled as accurately as they could be by the simulator, it is our belief that those modeling decisions will not significantly alter the effects of most flow management strategies.

2.2 Aircraft

2.2.1 Individual Aircraft

The decision to keep track of individual aircraft was made at a very early stage of this

work; other possibilities were not considered. The primary advantage of this approach is that it is very realistic; any real-world flow management strategy must specify flow control actions on individual aircraft, and this requirement is maintained in the simulator. The existence of individual aircraft also makes the collection and evaluation of statistics straightforward.

2.2.2 Aircraft Dynamics

Because of the potentially large scale of a simulation scenario, it is not practical to model the dynamics of aircraft very precisely. It is our strong feeling that this is also not necessary; the modeling does not need to be perfect, just good enough to permit the accurate evaluation of flow management strategies. The two aircraft state variables we think are most important are speed and altitude, since they affect both flight time (at lower altitudes, speed restrictions may apply) and the cost of operating the aircraft. Flight time affects delay and operating cost, performance measures which would almost certainly be used to compare flow management strategies. To keep things simple, we decided to limit the simulator to three altitudes, ground, low, and high, and two speed levels, cruise and minimum consumption (Maximum Endurance Speed). The speed levels can represent different speeds at different altitudes. Aircraft of different types can have different speeds, as well. Of course, the simulation scenario designer is free to selectively ignore the differences between speeds (in terminal areas, for example). Another benefit of having the three altitude levels is that flow management strategies which cause necessary delays to be taken on the ground or at high altitudes can be rewarded.

2.3 The ATC System

A network model is used to represent the ATC system. This was an obvious choice; networks are very flexible and easy to specify. A more difficult decision was how to model the transit of aircraft through the network elements. The two options considered were simple queueing systems or a more complicated system in which an aircraft's rate of travel through a region is some function of the number of aircraft in that region (which can change as aircraft enter and leave the region). Both systems could have been implemented, but this

was considered unnecessary.

The latter system is clearly the more complicated of the two, since every time an aircraft enters an area or exits from an area, the travel rate of every other aircraft in the area must be recomputed. Of course, a queueing system could potentially have varying service rates depending on the number of customers in the system, but the queueing systems we had in mind were simple: they have one waiting queue and several servers which are unaffected by the length of the waiting queue, for example. As to which system more accurately modeled en route sectors and runway systems, the evidence supporting the hypothesis that the transit time of an aircraft passing through an en route sector is significantly affected by other aircraft entering or leaving the sector is not very strong. In addition, it is not at all clear how such a function relating travel rate to the number of aircraft in an area would be determined.

In the end, the queueing systems approach was chosen. The service time function for the queueing system can depend on the state of the queueing system (and potentially other parameters of the network), but once the servicing of a customer begins, changes in the state of the queueing system (for example, a change in the number of customers in the queueing system) do not affect the service rate of the customer unless the customer's service rate changes for some other reason (for example, a speed change by the aircraft). Because of that restriction, the implementation of queueing systems is relatively easy.

2.4 Flights

2.4.1 Flight Restrictions

In the simulator, flights are restricted to go from their airport of origin to their destination airport without stopping at any other airports. One reason why that restriction was imposed is the following: the allocation of aircraft to flights can be modeled very accurately or just approximately, but it was decided that doing it accurately would be quite difficult and would require some arbitrary decisions about how the allocation is done; once the decision to do the allocation in an approximate manner was made, to then allow flights to make intermediate stops would mean that precise aircraft allocation would be done for those flights and approx-

imate aircraft allocation would be done for the flights which do not make intermediate stops. This may not seem like a serious inconsistency, but the prohibition of intermediate stops also simplifies several aspects of the simulator, as well as the modeling of the effects of arrival delays at an airport on departure delays at the same airport, as described in Section 2.4.3.

If a simulation scenario designer is creating a flight schedule based on airline schedules containing flights which make intermediate stops, then those flights can be broken up into flights with no intermediate stops. However, a complication occurs when delays and cost are calculated: there may be some system-wide double counting of delay time. To illustrate this, consider the case in which a flight makes one intermediate stop, arriving late at the intermediate stop and late at the destination; we might say that the total flight delay was the amount of time after the estimated time of arrival at the destination, or we might say that it was the sum of the amount late at the intermediate stop and the amount late at the ultimate destination, or some intermediate quantity; if we were to break the flight into two flights to accommodate the simulator's prohibition of intermediate stops, then the delay of each of the two flights is independently computed. But now the possibility exists that the second flight might depart before the first one arrives; this complication is discussed in Section 2.4.3.

2.4.2 Unscheduled Flights

Two factors limit the accuracy of any projections of air traffic demand at any point in the ATC network: airline schedules are not always accurate, and information on non-airline flights is frequently not available well in advance. In the simulator, we try to model that uncertainty by having two types of flights: scheduled and unscheduled. Scheduled flights, which may be thought of as airline flights, are usually known to the system (file flight plans) before the simulation starts and are subject to aircraft conservation restrictions (see Section 2.4.3). Unscheduled flights are generally not known to the system until shortly before departure and are not subject to aircraft conservation restrictions.

2.4.3 The Effect of Arrival Delays at an Airport on Departure Delays

Delays in arrivals at an airport cause delays in departures. As mentioned in Section 2.4.1, flights in the real world often make intermediate stops, and if a flight arrives late at an intermediate stop then it is likely to depart late as well. In addition, many airlines have hub-and-spoke route structures; at hubs, departures may be delayed by the airline if some arrivals are late, so as to allow passengers on late arriving flights to make connections on departing flights. Even when those two cases do not apply, airlines have a limited pool of aircraft at each airport and plan on using some aircraft from arriving flights for departures.

Modeling those effects accurately is not easy, unless each flight is inflexibly assigned an aircraft. As discussed in Section 2.4.1, it was decided not to use any precise aircraft allocation scheme, so we were forced to come up with an approximate method for modeling the effect of arrival delays on departure delays. One proposed scheme is the following: a particular aircraft which wants to depart from some airport will be delayed for a period of time based on the arrival delays of aircraft scheduled to arrive shortly before the scheduled departure time of the particular aircraft; however, it is not clear how to take into account aircraft which were expected to arrive shortly before the particular aircraft's scheduled departure time but have not.

The method we ended up using is based instead on the conservation of aircraft at each airport. A pool of aircraft is maintained at each airport. Each aircraft may optionally be tagged as belonging to a particular airline. When the estimated departure time of a flight arrives, the flight is allocated an aircraft if an appropriate one is available in the aircraft pool at the origin airport. There are two modes of operation: one in which flights must be assigned an aircraft which is from the same airline as the flight, and another without that restriction (this can be set by the simulation scenario designer). If the flight cannot be assigned an aircraft, then the flight is placed in a waiting line. Each arriving flight will relinquish its aircraft after some turnaround time. Unscheduled flights will be exempt from equipment allocation, as we are assuming that such flights will usually be made by operators which do not plan a flight without knowing that an aircraft will be available.

The advantages of the aircraft conservation method are that it is conceptually simple and very easy to implement. A disadvantage is that aircraft must be very carefully allocated to airports before a simulation run starts; this allocation will be dependent on the flight schedule. One can also question the accuracy of this method, but we believe that it is good enough to make reasonable evaluations of flow management strategies.

2.5 Flow Control

Since this simulator was intended for flow management research, we wanted to make the implementation of flow management strategies as simple as possible. The simulator supports the following flow control actions: ground holds for departure metering, speed controls and airborne holds for en route metering, and rerouting in order to prevent sector congestion. Due to the existence of only two speed levels (see Section 2.2.2), the imposition of speed control on an aircraft means that the aircraft's speed is reduced to the minimum consumption level. Holding an airborne aircraft means that the aircraft will progress no further until the hold is ended by flow control. In order to simplify the calculation of transit times, rerouting is permitted only at certain times (see Section 3.2.2).

Another part of facilitating the implementation of flow management strategies is to make the information necessary for planning easily available. In the simulator, it is easy to obtain information on flights which have filed flight plans. The information available includes an aircraft's flight plan, location, and estimated time of departure. In addition, all network elements keep lists of the aircraft they contain, and airport entities know about all aircraft that will arrive or depart from the airport (and have filed a flight plan). Information is also available on the expected transit times through network elements.

Tools for facilitating the implementation of flow management strategies would be a useful addition to the simulator. At present, all flow management strategies must be coded in Lisp. Methods for implementing various types of flow management strategies are discussed in Section 3.4. An example simulation which implements a simple Arrival Demand Management strategy is discussed in Chapter 4.

2.6 Performance Measures

The simulator has the ability to automatically collect a large amount of data during simulation runs. The following data is automatically collected for each flight: the amount of time accumulated at each altitude and speed level, delay time on the ground due to flow control and other causes, times associated with takeoff and landing, and a detailed diary of the flight (see Section 3.2.3). The simulation scenario designer has the option of having queues keep statistics on themselves, as well. A scenario designer is also free to add scheduler events which will collect other statistics (see Section 3.5.2).

Aircraft delays are computed based on a nominal flight time which is calculated when an aircraft files a flight plan. The nominal flight time is calculated using nominal transit time functions associated with queueing systems (see Sections 3.2.3 and 3.2.1). The delay types are the following:

- airline delay, which is the difference between the amount of ground delay not directly caused by flow control and the nominal airline delay (set by the simulation scenario designer)
- ground delay, which is the amount of time spent on the ground prior to departure due to a flow control ground hold
- departure delay, which is the delay incurred while waiting to takeoff and possibly delay in the departure airport's terminal area
- arrival delay, which is the delay incurred while waiting to land
- air delay, which is the difference between the actual flying time (not including departure delay and arrival delay) and the nominal flying time (computed using nominal transit time functions)
- total delay, which is the difference between the actual arrival time and the predicted arrival time (which in turn is based on the estimated departure time and the nominal flight time).

Note that the total delay for an aircraft should equal the sum of the other types of delay. Due to the flight restriction prohibiting intermediate stops, there may be some double counting of system-wide flight delays (see Section 2.4.1), but the extraneous delay will be entirely in the airline delay category. For precise definitions of the delay types, see Section 3.5.2.

Various statistics can be used to create a system performance function, if desired. A cost function can be assigned to each aircraft type; these costs can depend on any characteristics of a flight (see Section 3.2.3). One system performance function included in the simulator simply adds the costs of individual flights using the cost functions for the appropriate aircraft types (see Section 3.5.2).

Chapter 3

The Air Traffic Control Simulator and Its Use

3.1 An Overview of the Simulator

As discussed in Chapter 2, the simulator models the ATC system as a network of queueing systems. Aircraft start and end their trips at terminals, and pass through runway systems and airspace, which are modeled as queueing systems. Flights are generated from information contained in a flight schedule, which is a text file. The simulator consists of the entities which model the ATC system, and tools for flight schedule generation and statistical analysis.

As mentioned in the introduction, one major goal of the design of the simulator was the flexibility to model the ATC system with varying levels of detail and perform a wide variety of flow management experiments. With that in mind, the simulator was implemented in Lisp on a Texas Instruments Explorer Lisp Machine using an object-oriented style of programming. The Explorer's flavor system allows the definition of abstract objects and object methods which are used in message passing [4].

In object-oriented programming, programs are built around entities (called objects) which represent various items in the real world. Each object has a local state and operations that can be performed on it. Objects can be grouped into types, and objects of the same type will have similar behavior. For example, the simulator models airports using an object type called an airport. In the Explorer's flavor system, an operation associated with an object type is called a method, which is invoked by sending a particular message to an object of

that type. Objects of different types may share similar operations, in which case it is possible to allow the object methods to be invoked by the same message; therefore certain operations can be generic. For example, area objects and runway system objects both have an operation which returns the number of aircraft objects that the area or runway system object contains. Section 3.2 discusses the objects used in the simulator in moderate detail.

3.2 The Simulator Objects

The simulator objects can be divided into a few groups:

- The scheduler is of the discrete time variety, and is usually invisible to the simulation scenario designer.
- Network objects are the objects which model the ATC system; they include airports, each of which contains a terminal and a runway system, areas, which are used to model airspace, and ports, which connect the other network objects.
- Aircraft objects represent flights; they contain information such as the flight plan, aircraft type, and various statistics. Related to aircraft objects are aircraft type objects, which contain aircraft type characteristics and a cost function for that type, and aircraft equipment objects, which are used in modeling departure delays from an airport due to arrival delays.
- Queueing system objects are used to model the transit of aircraft through runway systems and areas.
- Flow controllers are also objects, but their implementation is dependent on the flow management strategy being evaluated, so the flow controllers currently supplied with the simulator are only there as an example for simulation scenario designers; a scenario is not required to have any flow controllers.

There are also various utility objects, which include flight plans, queues, sampling objects, lookup tables, and random variates. For a summary of the main simulator objects and the modeling they are associated with, see Table 3.1.

<u>Object:</u>	<u>Serves to Model:</u>
Queueing System	transit through airspace and runway systems
Airport	an airport
Terminal	a terminal
Runway System	a runway system and (optionally) the associated terminal area airspace
Area	airspace (usually en route sectors)
Port	a control transfer point
Aircraft	a flight
Aircraft Equipment	departure delays caused by arrival delays
Aircraft Type	aircraft flight speeds and operating costs
Flow Controller	a flow control element
Flight Plan	a flight plan

Table 3.1: A summary of the main simulator objects and of the entities they model.

3.2.1 Queueing Systems

Queueing systems are used to model the transit of aircraft through airspace (area objects) and runway systems. Two different types of queueing systems are provided in the simulator: a standard single queue multiple server queueing system and a variable capacity queueing system. A null queueing system is also provided for cases in which it is desired that selected portions of the ATC system are never congested. A simulation scenario designer is free to implement other queueing systems as long as they are compatible with the simulator.

The two types of queueing systems have the following things in common: each queueing system has a waiting queue; each queueing system has a service time function (provided by the simulation scenario designer), which may depend on the customer, the state of the queueing system, and time; each queueing system has a nominal service time function (used as a reference and for planning, provided by the simulation scenario designer), which may depend on the customer and the customer's projected time of entry into the queueing system; when the queueing system enters a customer into a waiting queue or removes a customer from a waiting queue, the customer is so informed; the queueing system has the capability of rescheduling the customer's exit from the queueing system (usually due to a speed change by the customer, which is an aircraft in this simulator) assuming that the customer keeps track of its own progress through the queueing system (which aircraft objects do).

Single queue multiple server queueing systems have one waiting queue and a fixed number of servers. All of the servers have the same service time function, and optionally a setup time function. The waiting queue is implemented as a double ended queue (see Section 3.2.4). The default queue discipline is first-in-first-out, but other queue disciplines can be used if defined by the simulation scenario designer.

Variable capacity queueing systems differ from the single queue multiple server variety described previously in that they can service a variable number of customers (possibly an unlimited number), and there is no setup time function. The service capacity can be changed at any time using the appropriate message. If the service capacity is lowered below the number of customers currently being serviced, the servicing of the excess customers is not stopped, but when servicing is completed those excess customers are not replaced by new customers or customers from the waiting queue. Customers being serviced are placed in a processing queue, which like all queues, can maintain statistics on itself.

When a customer enters a queueing system, the customer servicing will begin immediately if the queueing system has unused capacity (idle servers in the case of a single queue multiple server queueing system), otherwise the customer is entered into a first-come-first-serve waiting queue. The service time function is called with the customer and the queueing system as parameters, returning a number which is the service time for the customer. A scheduler event is added which will terminate the service of the customer at the appropriate time. The speed of an aircraft may change due to speed control or an airborne hold, in which case the time at which the customer servicing will be completed may change. When such an event occurs, the aircraft computes the fraction of the service time in the queueing system which has passed. The remaining service time is the fraction of service remaining times the value returned by the service time function based on the aircraft's new speed. In the case of an airborne hold, the fraction of service completed is computed before the hold starts, and a new service completion time is computed after the hold ends. Queueing system objects and aircraft objects have special operations which allow this rescheduling of the completion of service to occur. When the servicing of a customer is complete, the customer is passed back to the queueing system's associated runway system or area.

The nominal service time function is used by the simulator for two things: first, all flight delays must be based on some reference, and the nominal service time function is used to calculate that reference for each aircraft which will traverse the area associated with the queueing system; second, flow management strategies can use the nominal service time functions for making predictions about the future state of the system given some set of circumstances. As a result, it may be desirable to have the nominal service time function change with simulation time in a fairly sophisticated simulation, reflecting changing estimates of the properties of the service time function at some future time.

3.2.2 Network Objects

The network objects are used to model elements of the ATC system. Characteristics of some of these objects are summarized in Figure 3.1.

Ports

A port may be thought of as a point where the control of aircraft is transferred from one group of air traffic controllers to another. In the simulator, ports are used to connect together two network objects. A port can connect two areas, or one area and one airport. Flight plans are specified in terms of airports and ports; every flight plan begins and ends with an airport, and the rest of the waypoints are ports. Thus, ports define the entry point and exit point from an area. That information is often used in calculating the transit time of an aircraft through an area; that transit time is usually dependent on the entry and exit points (see Section 4.2).

Areas

An area can be used to model a region of airspace such as an en route sector. The principal components of an area object are a queueing system, which models the transit of aircraft through the region of airspace, a flow controller (optional), and a port-area table. Areas also have the following features: they have an altitude, which is either ground, low, or high; they notify an aircraft as the aircraft enters the area; they keep track of the aircraft they contain,

Object Characteristics: Areas, Terminals, and Runway Systems

Area:

- models a region of airspace (usually an en route sector)
- contains the following objects:
 - queueing system
 - port-area table
- has an altitude (low or high)
- is optionally associated with a flow controller
- has an aircraft count threshold, above which the flow controller is notified

Terminal:

- models a terminal at an airport
- is associated with the following objects:
 - airport
 - runway system
 - flow controller (optional)
- maintains a pool of aircraft equipment objects used in simulating the effects of arrival delays on departure delays
- contains a queue of aircraft waiting for appropriate aircraft equipment objects
- maintains lists of aircraft which start or end their flights at the terminal

Runway System:

- models a runway system and possibly airspace surrounding the runway system (the terminal area)
- is associated with the following objects:
 - airport
 - terminal
 - flow controller (optional)
- is otherwise almost identical to an area object

Figure 3.1: A summary of the characteristics of certain network objects.

and if the number of aircraft exceeds a certain threshold, the area's flow controller (if any) is informed.

When an aircraft enters an area, the area notifies the aircraft that it is entering the area, the area's flow controller (if any) is given the opportunity to change the aircraft's flight plan, then the aircraft is passed to the queueing system. The queueing system "services" the aircraft, then the aircraft is passed back to the area. The area then queries the aircraft as to its exit port, then that port is used as a key for the port-area table, which returns the next area (or possibly an airport) the aircraft will enter. The current area then passes the aircraft to that next area.

Airports

Airport objects are used to model airports. An airport object's only purpose is to link together a terminal object and a runway system object, both of which will be discussed next.

Terminals

Terminals are the actual sources and sinks of aircraft. Their main components are a flow controller (optional) and a pool of aircraft equipment. In addition, terminals are associated with a single runway system and a single airport.

One important function of a terminal is to keep track of every aircraft which will depart from that terminal or arrive at that terminal at some time. When an aircraft files its flight plan, the origin and destination terminals are informed of the planned departure or arrival. This information is available to any flow controller for planning purposes. In Arrival Demand Management, it is necessary to estimate how many aircraft will be arriving at each airport and when; the terminal's list of aircraft which are expected to arrive at a later time is very useful for that purpose.

Each terminal has an equipment pool. As mentioned in Chapter 2, we wanted to have arrival delays at a terminal cause departure delays at the same terminal, and the chosen method was to allocate at each terminal a pool of aircraft equipment objects, which are assigned to scheduled departures and reclaimed from scheduled arrivals. There are two

different ways this equipment assignment can be done: all equipment can be available to any airline or each aircraft equipment object can be tagged with an airline and that aircraft equipment object can only be assigned to a flight of that airline. When a flight (aircraft object) arrives at a terminal, its aircraft equipment object is relinquished and goes into the terminal's equipment pool. However, in order to model the turnaround time for aircraft, the equipment need not be relinquished immediately; the simulation scenario designer can specify the turnaround time for each aircraft equipment object to be a constant or a random variate.

When an aircraft's estimated departure time arrives, the aircraft is activated and its departure terminal is informed. If the aircraft is a scheduled flight, it must be assigned equipment from the equipment pool. If there is an appropriate aircraft equipment object in the terminal's equipment pool it is assigned to the aircraft and removed from the equipment pool, otherwise the aircraft is put into a queue containing aircraft waiting for equipment, and each time a new aircraft equipment object is relinquished, it will be assigned to the first aircraft (flight) in the queue with which it is compatible, at which point the aircraft can proceed. After equipment assignment, the aircraft is assigned some random delay, which is intended to simulate the delays in departure times due to airline-related problems such as late boarding of passengers, mechanical problems with the aircraft, and the late loading of luggage; like the equipment turnaround time, the random delay can be specified by the simulation scenario designer, and can be either a constant or a random variate. After the random delay, the aircraft requests a departure. The origin terminal asks its flow controller (if any) whether the aircraft can depart, and the flow controller can answer in one of three ways: the aircraft can depart immediately, the aircraft can depart at some specified time, or the aircraft should request a departure at some specified time. If the aircraft is not cleared for an immediate departure, then the terminal informs the aircraft that it has been given a ground hold. When the aircraft gets a departure clearance, the terminal passes the aircraft to its associated runway system.

In order to take into account the random delay after equipment assignment for planning and delay calculation, the simulation scenario designer may specify a nominal airline delay,

which generally is the mean value of the random delay distribution. The nominal airline delay is subtracted from an aircraft's airline delay time when airline delay is computed by function `aircraft-delay-statistics` (see Sections 3.2.3 and 3.5.2). If not specified by the simulation scenario designer, nominal airline delay is zero.

When an aircraft is passed to a terminal (by its associated runway system), the aircraft is notified that it is entering the terminal. Then, a scheduler event is added which will make the aircraft relinquish its aircraft equipment object at the appropriate time.

Runway Systems

A runway system object is used to model a runway system and optionally airspace around a runway system. A runway system object is very similar to an area object. They both have the same components: a queueing system, which models the transit of an aircraft through the runway system, a flow controller (optional), and a port-area table. Runway system objects are also associated with a terminal and an airport. Runway system objects differ from area objects in one way: when an area is ready to pass an aircraft off to the next area or runway system, that object is always located in the port-area table; however, a runway system often passes aircraft to its associated terminal, which is not connected to the runway system through a port and thus is not in the port-area table. One possible motive for having a runway system object model both a runway system and the surrounding airspace (the terminal area) is that it may be desirable to have time accumulated in the surrounding airspace be accumulated in the departure time or arrival time categories instead of in the low altitude time categories (see Section 3.2.3).

A function called `make-simple-runway-queueing-system` is provided in the simulator. This function creates a queueing system which is a simplified model of a runway system. The queueing system has one server and a first-in-first-out waiting queue. Takeoffs and landings take different amounts of time; the ratio of the service times defaults to 10. This value was chosen as the default because we did not want waiting departures to affect arrivals very much in the simulation scenario described in Chapter 4. In the most elementary use of this function, the number of operations per hour and the nominal number of operations per hour

are specified. The nominal number of operations per hour is used to create a nominal service time function, which is used to calculate an aircraft's nominal departure and arrival times (see Section 3.2.3) and is available to flow controllers for planning purposes. The takeoff and landing times are calculated based on a 50/50 mix of landings and takeoffs, with the actual times being taken from a normal distribution whose standard deviation is 10% of the mean. The simulation scenario designer has the following options: the operations and nominal operations parameters can be time functions; values for the landing to takeoff time ratio, the assumed mix, and the service time standard deviation can be specified; the normal random variate can be initialized with a seed. This function is used to create queueing systems for the runway systems in the simulation scenario described later (see Section 4.2).

3.2.3 Aircraft and Related Objects

Aircraft

An aircraft object represents one flight, not an actual aircraft (which is modeled by an aircraft equipment object and an aircraft type object). Aircraft objects contain the following: a flight name, a flight plan, a speed (cruise or minimum consumption), an altitude (ground, low, or high), a type, and an aircraft equipment object (some of the time). An aircraft (flight) is either scheduled, or unscheduled; scheduled flights must be allocated equipment before departure. Scheduled flights will generally file flight plans before the flow control planning phase, but the default for unscheduled flights is that the flight plan is filed at the estimated departure time; this can be specified in the flight schedule (see Section 3.3.1). The convention used by the random schedule generator is that unscheduled flights have airline abbreviation "N" (see Section 3.3.2). Aircraft also maintain a large amount of information about themselves. The characteristics of an aircraft object are summarized in Figure 3.2.

In order to allow the types of flow control discussed in Chapters 1 and 2, aircraft objects have the following capabilities: an aircraft's flight plan can be changed by flow control when an aircraft requests a departure, enters an area, or enters a runway system; if an aircraft is in an area, a flow controller can impose speed control, causing the aircraft to reduce its speed to minimum consumption, or the flow controller can give the aircraft an airborne hold, in

Object Characteristics: Aircraft, Aircraft Type, and Aircraft Equipment

Aircraft:

- models a flight
- has the following contained and associated objects:
 - flight name
 - aircraft type
 - flight plan
 - aircraft equipment (some of the time)
- maintains the following information:
 - speed (cruise or minimum consumption)
 - altitude (ground, low, or high)
 - location (a network element)
 - flight phase
 - progress through the current network element
 - accumulated time in various flight phases
 - a diary
- is either scheduled or unscheduled

Aircraft Type:

- contains a cost function
- contains the following speeds:
 - low altitude cruise
 - low altitude minimum consumption
 - high altitude cruise
 - high altitude minimum consumption

Aircraft Equipment:

- used in the modeling of how arrival delays cause departure delays
- may belong to a particular airline

Figure 3.2: A summary of the characteristics of objects discussed in the section entitled “Aircraft and Related Objects”.

which case the aircraft will progress no farther until the flow controller tells the aircraft to continue.

The aircraft object's speed will be cruise under normal conditions. However, the speed will be changed to minimum consumption under any of the following conditions: while in a waiting queue, after being so ordered by a flow controller (speed control), or after being given an airborne hold by a flow controller. The aircraft object's altitude is just the altitude of the area, terminal (ground), or runway system the aircraft is currently in.

Aircraft objects keep track of time accumulated during several flight phases:

- airline delay time, which is the amount of time between the estimated departure time and when the aircraft first requests a departure; this includes time spent waiting for equipment and any random delay after equipment is assigned (see Section 3.2.2)
- ground hold time
- departure time, which is the amount of time spent in the departure runway system
- low altitude cruise time, which is the amount of time spent cruising at low altitude
- low altitude minimum consumption time which is the amount of time spent at low altitude and minimum consumption speed
- high altitude cruise time
- high altitude minimum consumption time
- arrival time, which is the amount of time spent in the arrival runway system.

Upon filing a flight plan, an aircraft object will compute nominal times in a few modes of operation based on the nominal transit time functions of the queueing systems contained by the areas which the aircraft will traverse. These times include nominal departure time, nominal low altitude cruise time, nominal high altitude cruise time, and nominal arrival time. A flow controller can use those nominal times, the nominal airline delay, and the aircraft's estimated departure time to predict the aircraft's nominal arrival time for planning purposes.

The nominal times are used in conjunction with the accumulated times to compute various types of delay (see Sections 2.6 and 3.5.2). The same information is often used by the cost functions contained in aircraft type objects (see Section 3.2.3).

An aircraft object maintains a diary, which includes important events and their times. The events recorded include the following: filing of the flight plan, activation at the estimated departure time, equipment assignment (for scheduled flights), the beginning of a ground hold (if any), entry into each area, the beginning of an airborne hold imposed by flow control, the end of an airborne hold, the beginning of speed control imposed by flow control, the end of speed control, entry into a waiting queue, exit from a waiting queue, and deactivation upon arrival at the destination. This diary could be used to generate many additional statistics, although at present no such facility has been developed.

Aircraft Equipment Objects

Aircraft equipment objects were conceived in order to try to model the effects of arrival delay at an airport on departure delays at the same airport (see Section 2.4.3). Aircraft objects are extremely simple, only containing an airline name, which is only used if the simulation scenario requires that flights from an airline only use aircraft equipment from that airline.

Aircraft Type Objects

Aircraft type objects have two functions: they contain speeds characteristic of that aircraft type (cruise and minimum consumption speeds at low and high altitudes), and they contain a cost function which is used to compute the cost of flights which use that aircraft type. The cost function is generally a function of the amount of time accumulated by an aircraft at various speeds and altitudes. The simulation scenario discussed in Chapter 4 uses only one aircraft type, a "Medium Jet".

3.2.4 Other Objects

The simulation designer may not need knowledge of the following objects, but they are

listed here for completeness.

Flight Plans

The flight plans used in this simulator have the following format: an airport, then several ports, then another airport, each of which specifies a waypoint along the route of travel. A flight plan object consists of a list of those waypoints and a pointer to the current waypoint, which is generally the entry port. In order to run a simulation, a default flight plan must be defined for every origin-destination airport pair for which there will be flights.

Random Variates

The simulator includes a random variates package written by Dr. John Pararas. Several types of random variate objects have been defined: deterministic, discrete, uniform, exponential, normal, and lognormal. Discrete and exponential random variates are used in the generation of random schedules, described in Section 3.3.2. Exponential, normal, and lognormal random variates are used in the simulation scenario described in chapter 4. All random variates may be initialized with a seed so that experiments may be repeated exactly.

Queues

The simulator includes a double-ended queue package written by Dr. Pararas. These queues are used extensively by the simulator. With appropriate initialization, these queues will maintain their own statistics using sampling objects described later. The default queue discipline is first-in-first-out, but the simulation designer is free to add other queue disciplines.

Sampling Objects

The simulator includes a sampling package written by Dr. Pararas. At present, the only statistics the sampling objects can compute are moments, but extensions are planned. Sampling objects are used by queues to keep their own statistics, and by the function `aircraft-delay-statistics` (which is described in Section 3.5.2).

Lookup Tables

The simulator includes table objects of one and two keys. The port-area tables used in areas and runway systems are of this type, as is the default flight plan table. The simulation scenario described in Chapter 4 uses tables which contain distances between ports, for the calculation of transit times. The Lisp system also provides hash tables, which are used for the recall of many simulator objects by name.

3.3 Flights, Flight Schedules, and Random Schedule Generation

As discussed previously, a flight begins at an airport, ends at an airport, and makes no stops along the way. An aircraft object is associated with each flight. A flight is specified by the following: an origin airport, a destination airport, an estimated departure time, an aircraft type, the amount of time prior to departure that the flight plan is filed (this is optional), and whether or not the flight is scheduled.

A flight name consists of one or more letters followed by one or more numbers. The letters in the flight name designate an airline. The random schedule generation function assumes that airline “N” designates an unscheduled flight.

3.3.1 The Flight Schedule

A flight schedule is a text file containing data which specify a group of flights. Schedule files can be created by hand or by a program, then edited and possibly combined with other schedule files. Each line of the text file defines one flight. The flights need not be in any particular order. The first line of a flight schedule file may set a default aircraft type; this is almost always a good idea.

Each line of the flight schedule should have the following format: the estimated departure time (in seconds), the flight name, the origin airport’s name, the destination airport’s name, then optional information. Optional information includes the aircraft type (if different from the default aircraft type), whether or not the flight is scheduled (the default is scheduled), and the amount of time before the estimated departure time the flight plan is filed (for

```

:default-aircraft-type "Medium Jet"
57156 "EA7492" "BOS" "NY"
57340 "AA3254" "BOS" "NY"
57412 "UA5962" "BOS" "CHI" :type "Heavy Jet"
57467 "PA326" "BOS" "NY" :type "Heavy Jet"
57575 "PA8706" "BOS" "NY"
58034 "DL6940" "BOS" "NY"
58109 "NW1995" "BOS" "NY"
58155 "N2086" "BOS" "CHI" :unscheduled :advance-notice 1921 :type "Light Jet"
58340 "UA1489" "BOS" "NY"
58375 "DL257" "BOS" "NY"
58405 "PE4580" "BOS" "NY"
58747 "N4044" "BOS" "NY" :unscheduled :advance-notice 3504 :type "Light Jet"
59033 "N3458" "BOS" "CHI" :unscheduled :advance-notice 10027 :type "Prop C"
59187 "NW7220" "BOS" "CHI" :type "Heavy Jet"
59237 "N8751" "BOS" "CHI" :unscheduled :type "Prop A"
59293 "AA2797" "BOS" "CHI"
59487 "UA6003" "BOS" "CHI" :type "Heavy Jet"
59553 "EA5653" "BOS" "NY"
59572 "UA2623" "BOS" "NY"
59967 "AA8709" "BOS" "CHI"
60014 "DL4037" "BOS" "NY" :type "Heavy Jet"
60339 "N9675" "BOS" "CHI" :unscheduled :advance-notice 5489 :type "Light Jet"
60354 "DL6375" "BOS" "NY"

```

Figure 3.3: A portion of a flight schedule file is displayed above. The first line of the file defines a default aircraft type. The columns are the estimated time of departure (seconds), the flight name, the origin airport name, and the destination airport name.

scheduled flights, the default is that the flight plan is filed before the initial flow control planning phase, which is before the simulation starts; for unscheduled flights, the default is that the flight plan is filed at the estimated departure time). A portion of a flight schedule is shown in Figure 3.3.

Schedule files are read into the simulation by invoking function `read-schedule`, which does many consistency checks, including insuring that every airport and aircraft type mentioned in the schedule exists and checking for duplicate flight names (which are not permitted).

3.3.2 Random Schedule Generation

The simulation includes tools for generating random schedules. The tools are very flexible in that it is possible to create a simple random schedule quite easily by omitting many options, yet quite complicated random schedules can be created as well. When specifying a random schedule, there are only four required parameters: the earliest departure time, the latest departure time, the schedule file name, and specifications for the flight generators. It is also highly recommended that a default aircraft type be specified, as that will almost always make the schedule file shorter and easier to read. Random schedules are created by invoking function `make-random-schedule`.

For each airport which has departures, one or more flight generators must be defined. Each generator requires the following parameters: an origin airport, a discrete or deterministic random variate which chooses destination airports, and a demand rate function (of time). By having multiple generators for each departure location, it is possible for origin-destination pairs to have different demand rate functions. The generation of flight times goes as follows: the demand rate is looked up at the time of the last flight; the base interval is computed from the demand rate, then multiplied by a value sampled from the unit interdeparture random variate (the default is negative exponential), then added to the time of the last flight giving the next flight time. Note that no correction is made for the possibility that the demand rate may have changed over the interval between two flights; as a result, it is not a good idea to have demand rate functions which change very quickly compared to the average interdeparture time. Function `make-demand-function` can be used to create demand rate functions which have steps.

After a flight destination is chosen, its airline is selected. The selection is done by a function which may depend on the origin and destination. A default function is provided, or the schedule designer may write some other function to do the same task. The default function ignores the origin and destination; it just samples from a discrete random variate. The schedule generation tools assume that airline "N" indicates an unscheduled flight, so that should be kept in mind when the function which chooses airlines is written. The selected

airline is used to create a random flight name, which is guaranteed to be unique within that file.

After the flight's airline has been determined, the aircraft type is selected. The selection is done by a function which may depend on origin, destination, and airline. The default function simply distinguishes scheduled flights from unscheduled flights, then samples from one of two discrete random variates, one for scheduled flights and one for unscheduled flights. If the aircraft type selected is the same as the default aircraft type, then the aircraft type is not written into the schedule file.

If the flight is unscheduled, then the amount of time the flight plan will be filed prior to departure is determined. The default function in the present system always returns one hour. The keyword in the schedule file for this information is advance-notice. Unscheduled flights are also noted as such in the schedule file (recall that scheduled is the default).

3.4 The Implementation of Flow Management Strategies

The simulator does not have a friendly input system for devising flow control strategies. As a result, flow management strategies must be coded in Lisp at the present time. In addition, flow management strategies must create the Lisp objects necessary to do the actual control of aircraft on an individual basis. The mechanics of issuing instructions to aircraft is quite simple; however, the translation of general strategies into control actions on individual aircraft can be a difficult task. This section suggests methods for the implementation of such strategies.

The simulator allows for several different types of flow control actions. Ground holds can be issued in order to meter departures to certain airports. Speed control and airborne holds can be used to implement airborne metering strategies. Congested sectors or bad weather can be avoided by rerouting aircraft.

It is likely that many flow management strategies will require a planning phase before the simulation starts. At that time, the planner may use any information known by the system before the starting time of the simulation. Complete information on flights which

have already filed flight plans is available; this includes estimated departure times, flight plans, and nominal flight times. Nominal transit time functions should reflect the system's best guess (before the simulation starts) of the future state of the ATC system.

3.4.1 Arrival Demand Management

To do Arrival Demand Management, airport capacities and the runway system demand profiles must be estimated. That information is used to devise a departure metering strategy.

The landing capacity of an airport at some time can be estimated by the following technique, if the runway system object's queueing system was made using function `make-simple-runway-queueing-system` (see Section 3.2.1): construct an aircraft object whose flight plan ends with the desired airport; set the flight plan's current waypoint pointer to the second to last waypoint; call the runway system object's nominal transit time function with the aircraft object and the desired time as arguments- this returns the time it takes for a landing; the estimated maximum number of landings per hour (ignoring departures) is one hour divided by the landing time. Similar techniques can be developed for runway system objects which include other types of queueing systems, as long as the runway system object's queueing system models only a runway system (see Section 3.2.2).

The number and timing of takeoffs and landings at each airport can be estimated based on flight plans which have already been filed; each terminal keeps track every aircraft which has filed a flight plan and which plans either a takeoff or a landing at that airport. As a first approximation, the estimated arrival time for a flight which is before its estimated departure time is the sum of its estimated departure time and its nominal flight time. For an aircraft which is en route, the transit time through future areas can be calculated by constructing an aircraft object with the same flight plan and adding the nominal transit times through the future areas, and the remaining transit time through the aircraft's current area can be estimated by asking the aircraft what fraction of the area it has traversed, then multiplying that fraction by the nominal transit time through the current area; of course, this ignores any congestion in areas along the way, but it should be a good first approximation. Flights which have not departed from their origin terminals due to a lack of equipment or random

delay are somewhat more difficult to handle, since the time remaining until departure and any congestion in the departure runway system should be considered, as well as the time in the air; a first approximation might be to estimate the remaining departure delay with some constant and the flight time using the method for aircraft which are en route.

To perform departure metering, flow controller objects must be designed and implemented. One method is to use one flow controller object for each airport whose departures must be metered. The flow controller's task is to give departure clearances and ground holds based on the metering strategy and the quantity and times of actual departures.

The implementation of static Arrival Demand Management strategies is not very difficult; one such strategy which has been implemented is described in Section 4.1. However, the implementation of strategies which change depending on the actual state of the system may be quite tricky (as of this writing none has been attempted). Such strategies would involve updating the runway capacity and demand estimates. The former would not be too difficult, but some of the aircraft would be en route, making the prediction of time of arrival at the destination runway system a bit trickier; approximate methods such as the ones described previously would be required. Planning might be done at some constant time interval, but it might not be necessary to plan the strategy for the entire simulation run each time if the run is longer than the longest expected nominal flight time. One might have to be careful to only make modifications to the strategy which would not cause undesirable transients in the system.

3.4.2 Arrival Metering

Arrival Metering involves controlling the flow of aircraft which are in the air so as to have those aircraft take any unavoidable delays away from the terminal area and preferably at high altitude. Any reasonable strategy for arrival metering must be dynamic, so methods of capacity and demand prediction described previously for Arrival Demand Management must be used. However, Arrival Metering is somewhat different in that the metering of an aircraft will occur not long before the aircraft enters the terminal area. This means that capacity estimates should be pretty accurate, but capacity and demand estimates need to be quite

accurate for this method to be effective.

Arrival Metering for a particular airport might be implemented as follows: whenever an aircraft enters one of a few areas, the area asks the flow controller whether the aircraft should be rerouted (see Section 3.2.2); besides the obvious rerouting question, this is a signal to the flow controller that an aircraft is entering that area; if the aircraft is headed to the particular airport, the flow controller adds a scheduler event which will trigger metering of that aircraft when the aircraft gets a certain distance away from its destination; when that event occurs, the flow controller estimates the aircraft's arrival time at the terminal area and estimates the congestion at that time; if the aircraft is quite likely to be delayed in the terminal area, speed control might be imposed on the aircraft for some period of time; in any event, the aircraft's estimated arrival time at the runway system can be used to update the prediction of congestion at the runway system in the future.

3.4.3 Strategies for Reducing Airspace Congestion

Airspace congestion can be reduced by some combination of rerouting aircraft around (potentially) congested areas and departure metering. These actions can be based on periodic projections of future traffic levels or may be triggered by some event in the ATC system such as a particular area signaling that it is becoming congested. When an aircraft requests a departure, enters a runway system, or enters an area, the network element in which the aircraft is located gives the flow controller the opportunity to reroute the aircraft. Departure metering is done by just giving an aircraft a ground delay, as described previously.

Areas and runway systems can notify a flow controller when the number of aircraft in the area or runway system exceeds a certain threshold, which is set by the simulation scenario designer (see Section 3.2.2). This is probably the easiest way to trigger measures to reduce congestion, if that approach is taken.

Strategies which try to predict potentially congested areas will have to project the future locations of aircraft, then either do rerouting or departure metering to prevent potential congestion. In order to estimate the future locations of aircraft, nominal transit time functions must be used, as described previously. Rerouting will probably require flow control objects

for areas in which rerouting will occur. Departure metering requires flow control objects at appropriate terminals (see Section 3.4.1).

3.5 Data Collection and Analysis

3.5.1 Data Collection

A large quantity of data is collected automatically during a simulation run. Aircraft objects maintain a diary and keep track of time accumulated in various phases of a flight (see Section 3.2.3). Queueing systems can keep statistics on queue lengths if the appropriate option is selected when the queueing system is created. This data is accessible using functions described later.

To collect other forms of data, the simulation designer has two options: for data which can be obtained by sampling the system state at times determined in advance, scheduler events can be added which will do the sampling; the collection of other data may require extensions or modifications to simulator objects.

3.5.2 Tools for Statistical Analysis

Function `aircraft-delay-statistics` will compile the following aircraft delay statistics for a group of aircraft:

- total delay, which is defined to be the difference between the actual flight time (arrival time minus estimated departure time) and the nominal flight time (calculated when the aircraft files its flight plan)
- airline delay, which is defined to be the difference between the time from estimated departure time to the first departure request (this is the airline delay time computed by an aircraft object) and the nominal airline delay, which may be set by the simulation scenario designer (see Sections 3.2.3 and 3.2.2)
- departure delay, which is defined to be the difference between the time spent in the origin airport's runway system and the nominal departure time

- arrival delay, which is defined to be the the difference between the time spent in the destination airport's runway system and the nominal arrival time
- air delay, which is defined to be the difference between the time spent in areas and the nominal time in areas
- ground delay, which is defined to be the ground hold time.

The data returned is the average and standard deviation of each type of delay over the group of aircraft selected. The group of aircraft for which the data is compiled is specified using filters. A filter is used to select only certain aircraft for which statistics are to be compiled. For example, we may be interested in the delays experienced by aircraft that were scheduled to land in Boston between 1800 and 2000 hours. Several filters have been defined and it is not difficult to implement others, although this must be done in Lisp. The filters which have been defined include scheduled flights, unscheduled flights, airline, origin, destination, nominal flight time range, scheduled departure time range, actual departure time range, scheduled arrival time range, and actual arrival time range. The conjunction of filters is also possible.

Function `aircraft-cost-statistics` returns the sum of the costs of each flight. Those costs are calculated using cost functions contained in the aircraft type objects associated with each aircraft (see Section 3.2.3).

Three functions exist to display statistics about network objects. Function `terminal-statistics` lists statistics compiled by two queues: the queue of aircraft waiting for equipment and the queue of available equipment. Function `runway-system-statistics` list statistics compiled by the waiting queue of the runway system's queueing system. Function `area-statistics` displays statistics compiled by the waiting queue of the area's queueing system, and if the queueing system is of the variable capacity type, statistics compiled by the processing queue. At present, those statistics are limited to the average and the standard deviation of the queue lengths.

Chapter 4

An Example Simulation Scenario Design

As discussed in Chapter 3, a scenario contains several related elements: a network, which represents the ATC system, flight plans between airports in the network, aircraft type definitions, one or more flight schedules, initial equipment pools for each airport, Lisp code which implements a flow management strategy, and values for various system parameters including the equipment turnaround time and the random delay after equipment allocation. The flight schedules are dependent on the network, as each airport mentioned in the flight schedule must be defined in the network, and on the defined aircraft types. Initial equipment pools should depend on the flight schedules and the equipment turnaround time.

This chapter details a simulation scenario which was implemented and tested. The scenario includes a simple flow management strategy. Results from simulation runs are also presented.

4.1 The Flow Management Strategy

To show that it is possible to implement a flow management strategy in connection with the simulator, a very simple Arrival Demand Management strategy was developed. This strategy is static; that is, while the simulation is running, no changes are made to the strategy based on new information. In order to implement this strategy, several new Lisp objects were designed and implemented.

The approach chosen was to meter departures by issuing ground holds, in order to better match the number of aircraft wanting to land to the arrival runway system capacity over a period of time. The strategy does not discriminate by origin or airline, but would in general favor slower aircraft (see below). However, we were only using one type of aircraft (“Medium Jet”) in our simulation, so the speed discrimination did not matter.

The user specifies two parameters: the planning period length and the ratio of landing slots to expected desired landings (the utilization parameter). For each destination, the planning function projects the number of desired landings and the number of landings the destination runway system can handle (ignoring any planned departures) for each planning period. The projections of the number of desired landings are based on the nominal transit time functions for the areas and the departure runway system and the estimated departure times for the aircraft which have filed flight plans (of which almost all are scheduled flights). The projections of arrival runway system capacities are based on the nominal service time functions for the runway systems, since the runway system objects are modeling only runways in this scenario (see Section 3.4.1). In order to estimate the effect of unscheduled traffic, the number of known flights in each time period is multiplied by a constant (which can be set by the user).

Each planning period is considered, in chronological order. If a planning period has more desired landings than landing slots, then a metering period is started. For a one time period example of this calculation, see Figure 4.1. Each origin airport is allocated landing slots at the destination proportional to the expected number of aircraft leaving that airport which desire landings at the destination airport. Any origin airport which has a very small number of aircraft (for example, zero) which desire landings is allocated some minimum number of landings (this can also be set by the user); this is to account for unscheduled traffic, which might otherwise get very long ground holds. Any excess demand is carried over to the next planning period. A metering period ends at the beginning of that planning period when demand for landing slots no longer exceeds the number of landing slots. Figure 4.2 gives an example of how landing slots are allocated.

For each origin airport and metering period, the number of allocated landings at the

Example Metering Calculations, Part 1

Flow control parameters:

$$\text{planning period length} = 1.0 \text{ hours}$$

$$\text{utilization ratio} = 0.9$$

$$\text{unscheduled allowance} = 0.2$$

$$\text{minimum allocated flights per hour} = 1.0$$

Time period parameters:

$$\text{destination runway system nominal landing transit time} = 103 \text{ seconds}$$

$$\text{known flights from origin A} = 1$$

$$\text{known flights from origin B} = 17$$

$$\text{known flights from origin C} = 22$$

$$\text{total known flights} = 40$$

Intermediate calculations:

$$\begin{aligned} \text{total expected flights} &= (\text{total known flights}) * (1 + (\text{unscheduled allowance})) \\ &= 40 * 1.2 \\ &= 48 \end{aligned}$$

$$\begin{aligned} \text{destination airport capacity} &= \frac{\text{planning period length}}{\text{destination runway system nominal landing transit time}} \\ &= \frac{1.0 \text{ hours}}{103 \text{ seconds}} \\ &= 35 \end{aligned}$$

$$\begin{aligned} \text{maximum allocated landing slots} &= (\text{destination airport capacity}) * (\text{utilization ratio}) \\ &= 35 * 0.9 \\ &= 31.5 \end{aligned}$$

Since the total number of expected flights (48) exceeds the maximum number of landing slots that can be allocated (31.5), departures expecting to arrive during the planning period must be metered.

Figure 4.1: An example calculation of whether metering is required during a planning period. It is assumed that there is no carryover from the previous planning period. Landing slot allocation is shown in the next figure.

Example Metering Calculations, Part 2

fraction of known flights from A = .025

fraction of known flights from B = .425

fraction of known flights from C = .550

Landing slot allocation for A:

$$.025 * 31.5 = 0.8$$

Since that is less than the minimum number of landing slots allocated per planning period (1.0), we will allocate 1.0 slots to A, with no carryover to the next period.

Landing slot allocation for B:

$$.425 * 31.5 = 13.4$$

Excess known landings from B carried over to the next planning period:

$$17 - 13.4 = 3.6$$

Landing slot allocation for C:

$$.550 * 31.5 = 17.3$$

Excess known landings from C carried over to the next planning period:

$$22 - 17.3 = 4.7$$

Since excess known landings from the current planning period will be delayed to the next period by ground holds, they will be added to the known landings for the next period.

Figure 4.2: An example of landing slot allocation for one time period, based on the example from the previous figure.

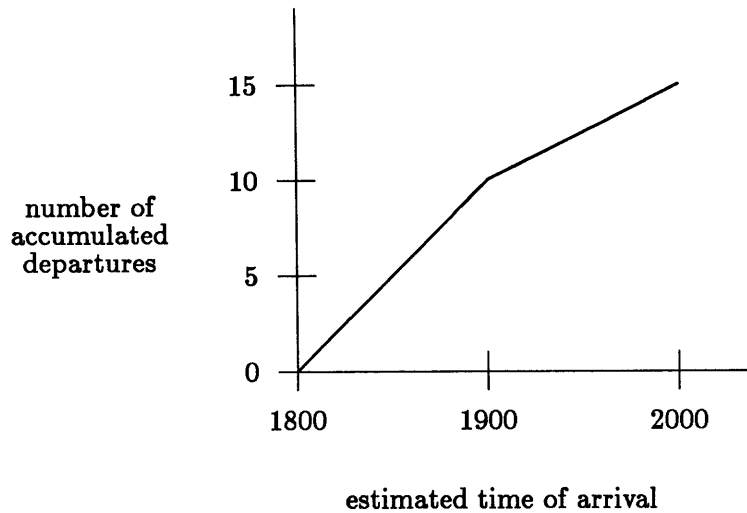


Figure 4.3: A function which relates accumulated departures during a metering period from an origin airport to the estimated arrival time at the destination airport's runway system; the origin airport has been allocated ten landings between 1800 and 1900 hours, and five landings between 1900 and 2000 hours.

destination airport in each planning period is used to create an increasing continuous function which relates the maximum accumulated number of departures during the metering period from the origin airport to the estimated arrival time at the destination airport's runway system. The function is composed of line segments, one for each planning period in the metering period. A line segment's time range is the length of a planning period, and its departures range is the number of allocated landings at the destination airport over that time period. The maximum number of accumulated departures at the beginning of the metering period is zero. For an example, see Figure 4.3. This function is packaged into a metering object, which also counts the number of aircraft which request departures and have an estimated time of arrival within the metering period (the cumulative departure count). For each metering period at each destination airport, a metering object is created for each possible origin airport, and the metering object is assigned to the flow controller for the origin airport.

After the planning phase, the simulation starts. When an aircraft is ready to leave

the terminal, the terminal's flow controller is queried as to whether the aircraft can leave the terminal. The flow controller projects the aircraft's estimated time of arrival at its destination's runway system based on the aircraft's previously computed nominal transit times through the departure runway system and areas along the aircraft's flight path. If the flow controller does not have a metering object for the aircraft's destination and estimated time of arrival, the aircraft is cleared for an immediate takeoff. If there is an appropriate metering object, then the metering object's cumulative departure count is looked up in the function relating the maximum number of accumulated departures to arrival times, yielding an arrival time; if the resulting arrival time is before the aircraft's estimated arrival time, then the aircraft is cleared for an immediate takeoff, otherwise the aircraft is given a ground hold whose duration is the difference between the two times. If the metering object's aircraft count exceeds the maximum number of departures at the end of the metering period, then the ground hold duration is the difference between the time the metering period ends and the aircraft's estimated arrival time. It would be simple to modify the flow management algorithm to discount such ground hold delays as well [1]. Here is an example: aircraft X, which intends to fly from airport A to airport B, requests a departure at 1700 hours, which gives X an estimated arrival time (at B's runway system) of 1840 hours; assume that departures from A to B are controlled by a metering object whose function relating accumulated departures to arrival times is the function in Figure 4.3 and whose cumulative departure count is 10 at the time of X's departure request; 10 previous departures corresponds to an arrival time of 1900 hours; therefore, B will be given a ground hold of 20 minutes, and the metering object's cumulative departure count will become 11.

One characteristic of the strategy is that slower aircraft will get smaller ground holds; this is because they will ask to depart before a faster aircraft which would arrive at the same time, and would therefore be more likely to get an immediate departure clearance or a small ground hold. During the runs we performed using this strategy, only one aircraft type was used, so speed discrimination was not a factor. Note that there is no discrimination by aircraft origin, as the number of landings at some destination airport allocated to an origin airport during some time period is proportional to the number of aircraft expected to depart

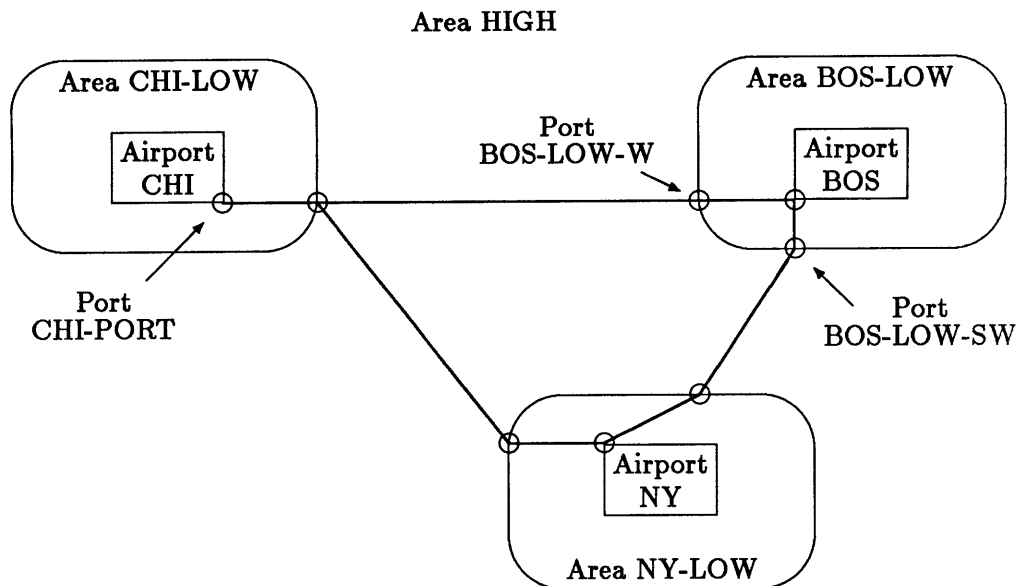


Figure 4.4: This is the network used in the demonstration scenario. The small circles are ports.

the origin airport en route to the destination airport during that time period.

4.2 The ATC System Network

The network used consists of three airports (“CHI”, “NY”, and “BOS”), three low altitude areas (“CHI-LOW”, “NY-LOW”, and “BOS-LOW”), one high altitude area (“HIGH”), and eight ports (see figure 4.4). Each terminal is connected through a port to its respective low altitude area. For example, airport “CHI” is connected to area “CHI-LOW” through port “CHI-PORT”. Each low altitude area is also connected to the high altitude area “HIGH” through one or more ports. For example, area “BOS-LOW” is connected to area “HIGH” through two ports: “BOS-LOW-W” and “BOS-LOW-SW”.

The queueing systems for the runway systems associated with each airport were constructed using the `make-simple-runway-queueing-system` function described in Section 3.2.2. The arrival to departure service time ratio was 10 (the default), and the assumed mix was 50% departures and 50% arrivals. To model the changing capacity of runway systems, the

operations and nominal operations parameters were chosen to be step functions of time. For “BOS”, the capacity was 55 operations per hour until 1800 hours, and 40 afterward; for “NY”, the capacity was 70 per hour until 1800 hours and 55 afterward; for “CHI”, the capacity was 70 per hour until 1900 hours and 55 afterward. For 40 operations per hour and a service time ratio of 10, the average landing time is 2:44 and the average takeoff time is 0:16; for 55 operations per hour, the times are 1:59 and 0:12; for 70 operations per hour, the times are 1:34 and 0:09. Note that for this simulation, the nominal service time function and the actual service time function had the same average value for each type of operation; this means that the system had accurate prior knowledge of the capacity of each of the runway systems at planning time.

The area network objects were quite simple. Since it was decided not to try to model congestion en route, each of the areas had an associated queueing system which had unlimited service capacity (the variable capacity queueing system with unlimited capacity). Transit times were determined using the aircraft’s speed and the “distance” between the entry and exit ports. The actual transit time had a normal distribution with a mean of the quotient of the aircraft’s speed and the distance and a standard deviation of five percent of the mean. The nominal transit time was the quotient of the aircraft’s speed and the distance. The distances were stored in a table for each area, the keys being the entry and exit ports. The distances through each low altitude area were set at 50 nautical miles. The distances in the high altitude area were 100 miles between “BOS-LOW-SW” and “NY-LOW-NE” (either direction), 600 miles between “CHI-LOW-E” and “NY-LOW-W”, and 700 miles between “CHI-LOW-E” and “BOS-LOW-W”. For a standard jet, the nominal transit times are 0.6 hours from “NY” to “BOS”, 1.6 hours from “CHI” to “NY”, and 1.8 hours from “CHI” to “BOS”.

4.3 Schedule Generation

A schedule was generated using the random schedule generation tools described in Section 3.3.2. Takeoffs were scheduled between 1500 hours and 2200 hours. One generator

<i>Flight Generation Rate (per hour)</i>							
<i>for Hour Starting At</i>							
<u>Origin</u>	<u>1500</u>	<u>1600</u>	<u>1700</u>	<u>1800</u>	<u>1900</u>	<u>2000</u>	<u>2100</u>
BOS	24	30	42	36	24	18	6
NY	30	36	48	42	30	18	6
CHI	12	12	18	30	36	30	12

Table 4.1: Flight generation rates used to create the 504 aircraft schedule used with the example simulation.

was used for each airport, so the destination probability distribution was not time-varying. The interdeparture time probability distribution was negative exponential. Function make-demand-function was used to produce a stepped demand rate function for each generator. From “BOS”, 180 flights were specified with a peak between 1700 and 1800 hours. From “NY”, 210 flights were specified with a peak between 1700 and 1800 hours. From “CHI”, 150 flights were specified with a peak between 1900 and 2000 hours. See Table 4.1 for details.

The destination distributions were set up so that the expected values are 240 flights between “BOS” and “NY”, 120 flights between “BOS” and “CHI”, and 180 flights between “NY” and “CHI”, with half of the flights going in each direction. Figure 4.5 shows the landing capacities and the demand rate for landings (based on the flight generation rates used to create the schedule) at each airport. The schedule that was actually generated had 133 flights from “BOS” to “NY”, 101 from “NY” to “BOS”, 59 from “BOS” to “CHI”, 55 from “CHI” to “BOS”, 75 from “NY” to “CHI”, and 81 from “CHI” to “NY”, for a total of 504 flights, compared to an expected total of 540.

For the rest of the schedule generation options, the defaults were taken; these included the default airline function, which has an unscheduled traffic fraction of 0.161, the default aircraft type function, which always chooses “Medium Jet”, and the default unscheduled traffic advance notice function, which always returns 1 hour. The unscheduled traffic fraction in the schedule generated was 0.176.

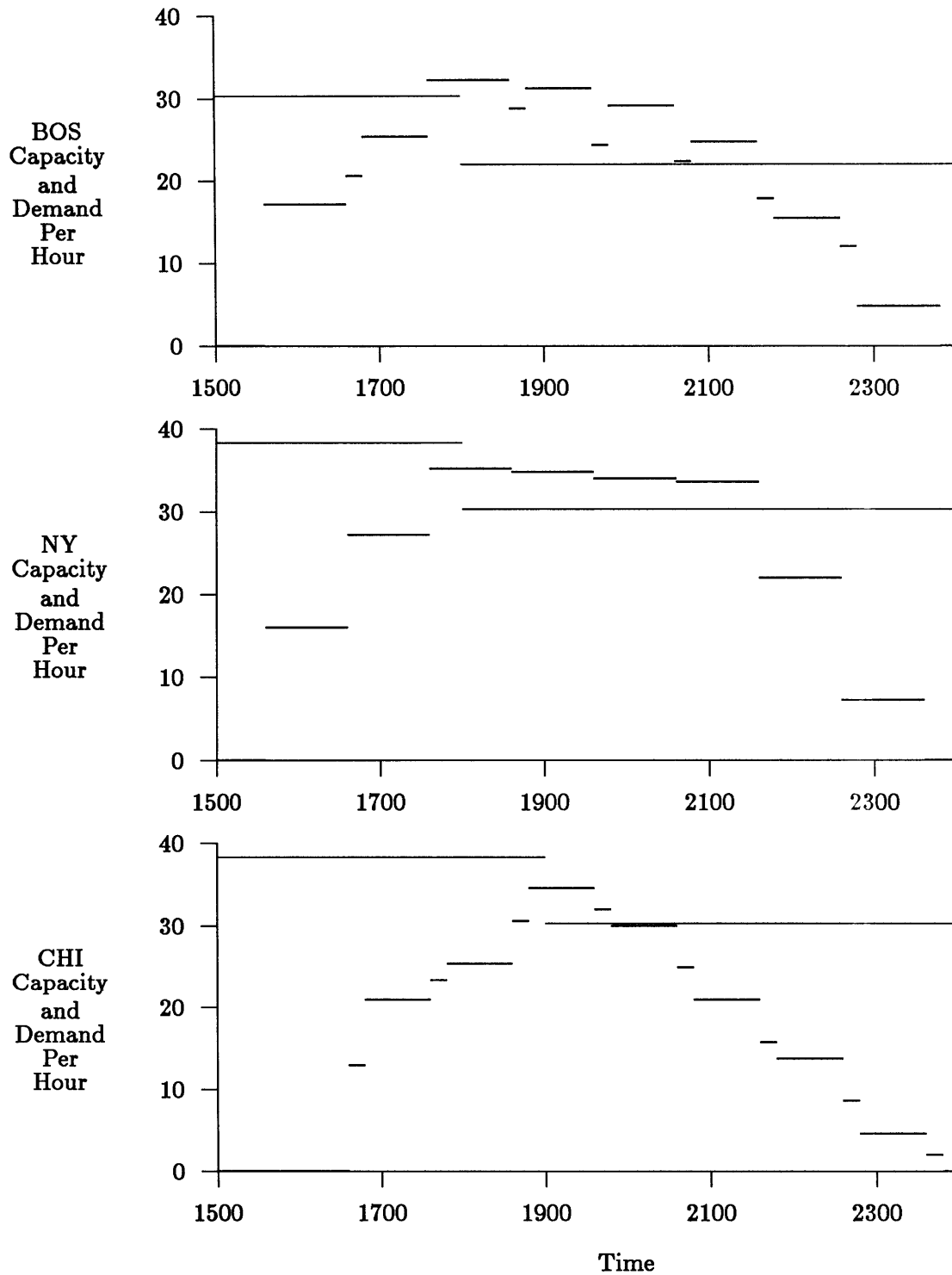


Figure 4.5: Airport landing capacities and demand rates (per hour); demand rates are based on the flight generation rates used to create the schedule; demand is indicated by the thick lines, and capacity by the thin lines.

4.4 Data Collection

We were most interested in the ability of the flow management strategy to reduce delays and replace delays in the air by ground delay. No special modifications were required to obtain those statistics. In order to look at the lengths of the runway system waiting queues, the queue time statistics option was specified when the runway systems were specified.

Delay data was processed using the `aircraft-delay-statistics` function and various filters (see Section 3.5.2).

4.5 Results of the Simulation Runs

For the 504 aircraft schedule discussed previously, a liberal amount of initial equipment was allocated in hopes of not causing too much airline delay. In addition, airline equipment compatibility was turned off. The initial equipment quantities were 72 for “NY”, 60 for “BOS”, and 48 for “CHI”, which are roughly proportional to the proportion of flights leaving each airport that one would expect the random schedule generator to create, given the specified demand rates. However, the schedule that was created actually had more flights leaving from “BOS” than from “NY”. In a realistic study of the properties of a particular strategy, it would be important to insure that equipment was well allocated at the start of a simulation run, so as to not distort the results.

For the simulation runs, the equipment turnaround time was assigned by sampling a lognormal random variate with a mean of 30 minutes and a standard deviation of 5 minutes. The random delay after equipment assignment (one of the two components of airline delay) was assigned by sampling a negative exponential random variate with a mean of 2 minutes, and the nominal airline delay was set to 2 minutes. One aircraft type was created, called “Medium Jet”. It had a high altitude cruise speed of 500 knots, and a low altitude cruise speed of 250 knots.

The scenario was run with a variety of parameter settings, concentrating on the effects of changing the utilization parameter. Each simulation run took less than ten minutes on an Explorer. At least five runs were done with each set of parameters. The results are

summarized in Table 4.2. The system was fairly congested, with the average total delay being almost 35 minutes when there was no flow control. However, the demand management strategy was able to significantly reduce delays directly caused by runway congestion (departure delay and arrival delay) for suitable values of the utilization parameter. In addition, for values of the utilization parameter above 0.90, the total expected delay was reduced as well. For values less than about 0.95, further reductions in the delays due to runway congestion come at the expense of increasing the average total delay but may still be desirable if system performance is evaluated based on a cost function such as the one used to compute the cost column in Table 4.2. Since the nominal airline delay is the same as the mean of the random delay distribution, the airline delay shown in Table 4.2 is time when flights were waiting for equipment. Airline delays have a slight tendency to increase as the utilization parameter decreases. It should be noted that there may be system-wide double counting of airline delay (see Section 2.4.1). The effect of changing the planning period length seems to depend on the value of the utilization parameter; for utilization parameters of 0.90 and above, a 60 minute planning period seems to work better than a 15 minute period, but for a utilization parameter of 0.80 the shorter planning period seems to work better. The variability of results from run to run was substantial for some values of the utilization parameter and the planning period length. Runs with a utilization ratio of 0.95 with a planning period of 15 minutes showed the largest variations: the average total delay for a run had a standard deviation of 1.4 minutes (six runs). Other simulation results are given in Table 4.3.

A few runs were done using a schedule which had about ten percent fewer aircraft than in the original schedule. The average total delay without flow control was about 17 minutes. Runs with flow control gave results similar to the previously mentioned runs, but the effects of flow control were not as large. This simply demonstrates the fact that flow control is most effective with congested systems.

The above results are quite encouraging. They show that given accurate information about the future capacities of the runway systems, nominal transit times through areas, and the amount of unscheduled traffic, even quite simple arrival demand management strategies may be quite effective in converting congestion delay (most importantly arrival delay) into

<u>Utilization</u> <u>Ratio</u>	<u>Planning</u> <u>Period</u> <u>(min)</u>	<u>Ground</u> <u> Holds</u>	<u>Ground</u>	<u>Average Delay (min)</u>			<u>Total</u>	<u>Cost</u>
				<u>Departure</u>	<u>Arrival</u>	<u>Airline</u>		
no flow control		0	0.0	11.0	12.2	11.0	34.2	40.9
1.00	15	210	9.4	5.4	6.8	11.3	32.7	31.1
1.00	60	183	8.4	5.5	6.6	11.2	31.8	29.9
0.95	15	233	12.4	4.1	5.0	11.1	32.4	28.6
0.95	60	197	11.0	4.4	5.5	11.2	32.2	28.6
0.90	15	256	16.0	3.2	3.3	11.5	33.9	27.4
0.90	60	217	13.5	3.6	3.7	11.6	32.3	26.3
0.80	15	327	26.4	1.9	1.6	12.7	42.7	32.5
0.80	60	297	26.1	2.3	2.1	12.8	43.2	33.8

Table 4.2: The affects of flow control on delay (504 aircraft schedule); delays were calculated using function aircraft-delay-statistics; cost is the sum of 2.0 times the arrival delay, 1.5 times the departure delay, and the ground delay.

<u>Filter</u>	<u>Delay (min)</u>
destination: "BOS"	38.1
destination: "NY"	35.2
destination: "CHI"	28.0
unscheduled	18.7
scheduled	37.5

Table 4.3: Total delay without flow control, averaged using various filters.

ground delay, and possibly even reducing the average total delay. Of course, this assumes that our model of the ATC system is reasonably accurate.

The demand management strategy was conceived and implemented in a few days. However, more complex strategies would probably take much longer to implement. At present, there is no friendly input system to aid the implementation of these strategies; all strategies must be directly coded in Lisp. Until such an input system is devised, the implementation of complex strategies may be quite time consuming and must be done by an experienced Lisp programmer.

Chapter 5

Conclusions and Recommendations for Future Work

Based on limited use by this writer, the simulator seems to have generally satisfied the design goals listed in the introduction. A simulation scenario which includes a simple flow management strategy was successfully implemented and tested using the simulator. Flexibility in the modeling of the ATC system appears to be a strength of the simulator. At least two significant weaknesses exist: it is not easy to collect certain types of data (in an elegant way), and no pre-programmed flow control tools are provided.

The data collection capabilities of the simulator are substantial, but it was discovered that some data was not easy to obtain. At one time, we wanted to look at the times aircraft were arriving at each runway system to see what the effects of departure metering were; to do this we would have had to either write a routine which would obtain the information from aircraft diaries (not too difficult to do, but possibly an inefficient approach), modify the runway system object type to record those times, or assign to each runway system a flow controller which records the time it receives a rerouting query for an aircraft which is landing. Another time, we were interested in a plot of how the queue lengths in each runway system were changing with time, but that data was not being recorded anywhere. In particular, the following additions to the simulator would make data collection and analysis much easier: sampling objects which record the values of state variables over time, extensions to allow the times of certain events to be recorded, and tools which can extract useful information from

aircraft object diaries.

The lack of pre-programmed flow control tools is another problem. It is clear that different types of flow management strategies share some common types of operations, such as the prediction of aircraft positions in the future, and tools could be provided to perform some of those operations. Until those tools are developed, a simulation designer will have to implement flow management strategies from scratch (in Lisp).

After the two previously mentioned weaknesses have been sufficiently corrected, a better user interface should be devised. At present, a simulation designer needs at least a working knowledge of Lisp, but it would be nice if that was not necessary.

The simple runway system model used in the example simulation scenario (see Section 4.2) would probably not be adequate for realistic studies of flow management strategies. A better queueing system model of a runway system should be developed; this model might also include the terminal area, which in the example simulation scenario was contained by the low altitude area surrounding each airport.

Certain characteristics of air traffic and the ATC system are only roughly approximated by the simulator (see Chapter 2). It would be interesting to study how those design decisions affect different types of flow management strategies.

In conclusion, the simulator could develop into a powerful tool for the evaluation of flow management strategies after certain weaknesses have been corrected. If nothing else, we hope that our experience in designing and implementing the simulator will be useful to others who wish to construct similar tools.

Bibliography

- [1] Sökkappa, Balraj G., “Arrival Flow Management as a Feedback Control System”, The MITRE Corporation, 1985.
- [2] Andreatta, Giovanni and Romanin-Jacur, Giorgio, “The Flow Management Problem”, University of Padova, 1986.
- [3] Andreatta, Giovanni and Romanin-Jacur, Giorgio, “Flow Management Problem and Optimal Priority Rule”, University of Padova, 1986.
- [4] Explorer Lisp Reference, Texas Instruments Inc., 1985.