AIRPORT LANDSIDE PLANNING AND SIMULATION MODEL (ALPS)

by
Mohit Kulkarni

Thesis submitted to the faculty of
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
Master of Science
in
Civil Engineering

APPROVED:

A.A. Trani, Ph.D., Chairman

D.R. Drew, Ph.D.
R.D. Walker, Ph.D.

February 14, 1994
Blacksburg, Virginia
c.2
AIRPORT LANDSIDE PLANNING AND SIMULATION MODEL (ALPS)

by

Mohit Kulkarni
Committee Chairman: Dr. A.A. Trani
Civil Engineering

(ABSTRACT)

The increasing demand on the air transportation system is causing delays due to congestion, leading to monetary losses and passenger inconvenience. Traditionally, research has been conducted to improve the airside component of the airport. This led to improvements in the airside component in turn leading to increased demand. The landside was not considered as a serious threat to the capacity of the airport. However, the increased demand and inconsistent improvements to the landside has made this airport component critical at some facilities.

Research in recent years has produced many methods to assess landside capacity and to predict the behavior of the system under different demand scenarios. Many of these tools are cumbersome and are not suitable for every day use of planning professionals. This research is aimed at developing a computer based simulation model (ALPS) to estimate or predict the capacity of the landside components under varying scenarios.

ALPS is a discrete-event simulation model developed using EXTEND (version 2 © Imagine That inc., 1992) a simulation software based on the 'C' programming language. This model is designed to be able to simulate a given airport in a short time with accurate results. The model runs on a Apple Macintosh computer and needs no special programming for effective use. The model is well suited for every day use of planning professionals.
ACKNOWLEDGMENTS

I sincerely wish to express my gratitude to my advisor Dr. A.A. Trani for the guidance he has given me from the conception to conclusion of this research and my stay at Virginia Tech. His constant support, encouragement and thoughtful suggestions have made it possible for the successful completion of this research. I have enjoyed his teaching and appreciate his enthusiasm in helping the students.

I am thankful to Dr. Drew for serving in my committee and appreciate his suggestions for my research. I was fortunate to be his student and have always enjoyed his teaching and especially appreciate his free spirit.

I am also thankful to Dr. Walker for serving on my committee and appreciate his suggestions for my research.

I thank my parents, brother and my sisters for their love, support, and confidence in me. It would not have been possible to pursue my graduate studies in the United States without their support and encouragement.
Table of Contents

1. Introduction .................................................................................................................. 1
   Background .................................................................................................................. 1
   Airport Environment Description .............................................................................. 2
   Airside ......................................................................................................................... 2
   Landside ..................................................................................................................... 3
   Research Scope, Objective, and Approach .............................................................. 4

2. Literature Review ....................................................................................................... 6
   Introduction .................................................................................................................. 6
   Defining Capacity ...................................................................................................... 6
   Airport Design process .............................................................................................. 7
   Description of Selected Simulation Approaches ....................................................... 15

3. Methodology ............................................................................................................... 20
   Introduction ................................................................................................................ 20
   Simulation .................................................................................................................. 20
   Simulation in The Airport Landside Environment .................................................... 25
   Extend as a Simulation Tool ....................................................................................... 28
   Model Construction Process ..................................................................................... 30

4. Model Description ...................................................................................................... 31
   Introduction ................................................................................................................ 31
   Model Environment .................................................................................................. 31
   Model Assumptions .................................................................................................. 32
   ALPS Model Features .............................................................................................. 33
   Enplaning Model Flow Description .......................................................................... 35
   Deplaning Model Flow Description .......................................................................... 37
   Alphabetical Block Reference for ALPSLIB Library .............................................. 42

5. Model Results and Analysis ....................................................................................... 67
   Introduction ............................................................................................................... 67
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Year Scenario</td>
<td>67</td>
</tr>
<tr>
<td>Horizon Year Scenario</td>
<td>67</td>
</tr>
<tr>
<td>Assumptions</td>
<td>67</td>
</tr>
<tr>
<td>Results</td>
<td>69</td>
</tr>
<tr>
<td>Result Graphs</td>
<td>72</td>
</tr>
<tr>
<td>6. Conclusions and Recommendations</td>
<td>76</td>
</tr>
<tr>
<td>Conclusions</td>
<td>76</td>
</tr>
<tr>
<td>Recommendations</td>
<td>77</td>
</tr>
<tr>
<td>Appendix A Generic Library Blocks</td>
<td>78</td>
</tr>
<tr>
<td>Appendix B Discrete Event Library Blocks</td>
<td>80</td>
</tr>
<tr>
<td>Bibliography</td>
<td>85</td>
</tr>
</tbody>
</table>
List of Figures

1. Landside Capacity Assessment, Management and Planning.......................... 26
2. Enplaning Passenger Model (ALPS) .................................................. 34
3. Deplaning Passenger Model (ALPS) .................................................. 38
4. Alphabetical List of ALPSLIB Library ............................................. 41
5. Structure of Baggage Claim Block .................................................... 50
6. Structure of Circulation Lobby Block ................................................. 51
7. Structure of Curb Delay Block .......................................................... 52
8. Structure of Customs Counters Block ............................................... 53
9. Structure of Deplaning Gate (Heavy) Block ....................................... 54
10. Structure of Deplaning Gate (Medium) Block .................................... 55
11. Structure of Deplaning Gate (Small) Block ...................................... 56
12. Structure of Gate Assignment Block ................................................. 57
13. Structure of Holding Room/Gate (Heavy) Block ................................. 58
14. Structure of Holding Room/Gate (Medium) Block ............................... 59
15. Structure of Holding Room/Gate (Small) Block .................................. 60
16. Structure of Immigration Block ...................................................... 61
17. Structure of Passenger Generator Block .......................................... 62
18. Structure of Security Block ............................................................. 63
19. Structure of Ticket Counters Block ................................................. 64
20. Structure of Transfer Passenger Block ............................................. 65
21. Structure of Vehicle Parking Block ................................................ 66
22. Deplaning Model (Roanoke Airport) ............................................... 70
23. Enplaning Model (Roanoke Airport) ............................................... 71
24. Model Results Base/horizon Year Scenario (Baggage Claim area)........ 72
25. Model Results Base/Horizon Year Scenario (Ticket Counters).............. 73
26. Model Results Base/Horizon Year Scenario (Security Counters) .......... 74
27. Model Results Base/Horizon Year Scenario (Circulation Area) Queuing .. 75
28. Model Results Base/Horizon Year Scenario (Deplaning Gate) Queuing ... 76
29. Model Results Base/Horizon Year Scenario (Baggage Claim Area) Queuing... 77
30. Model Results Base/Horizon Year Scenario (Ticket Counters) Queuing ... 78
31. Model Results Base/Horizon Year Scenario (Security Counters) Queuing .. 79
1. INTRODUCTION

Background

Ever since the historic flight of the Wright Brothers in their 'Flyer' in the year 1903, the aviation industry has seen very rapid and varied growth. However, the first daily aircraft passenger service did not start until 1914. The first commercial airliner in America was the Benoist Aircraft Company that began the airline services on New Year's day in 1914 with their type XIV flying boat [David Jefferies, 1988]. This began the era of commercial air transportation that has influenced the growth of industry and economy of the entire world like nothing else ever before.

Many airports were built during the war that were later modified to suit the needs of the modern civil aviation aircraft. By the end of the World War II, commercial airlines was a booming business. The fast travel mode of transport supported the growth of other businesses as well. This growth in industry attracted people to the cities and consequently the city started to expand. This expansion caused a scarcity of land around the business centers of the city and people were forced move to the outskirts of the city. Soon the land around the airports was all occupied by private owners restricting the airport expansion. Over the years with the improvement in technology, bigger aircraft were introduced which carried more passengers and cargo. This increase in demand required large airports to process the increase in passenger and cargo. However, expansion of many of the airports was not possible because of the private land ownership around them. These restrictions caused an increase in processing time of the passengers, implying increase in delays resulting in loss of revenue to both the users and suppliers of the air transportation. This trend continues and the losses today due to delays amount to millions of dollars. Statistics provided by the Federal Aviation Administration [FAA, 1992] which, is the main agency
in the US for regulating and certifying air carriers indicates that a delay of one hour costs an airline about $1600 [FAA, 1991]. Furthermore, the loss of revenue to the airports and passengers when added to the airline losses will amount to billions of dollars a year. FAA statistics indicate that 23 airports experience delays of over 20,000 hours annually with this figure growing to over 33 such airports as early as the year 2002. The growth in annual enplanements is forecasted to increase by 90% over the present levels in the next 15 years at an average rate of 6% a year. To accommodate such a tremendous growth with restriction on airport expansion, airports will have to make the best use of the existing facilities. This will present a big challenge to the nations air transportation system. The airport capacity improvement process will involve the enhancement of both the "Airside and the Landside capacities" against the backdrop of land scarcity, limited financial resources, environmental pollution impacts, and many more restrictions.

**Airport Environment Description**

The airport may be considered as a system consisting of two main components.

1) The Airside component, and
2) The Landside component

These two main components are further divided into respective sub components as follows:

**Airside**

1) Runways
2) Taxiways
3) Air traffic control systems
Landside

1) Aircraft parking positions and Gates
2) Terminal buildings
3) Baggage services
4) Parking
5) Access roadways

These sub components are further divided into basic elements. The capacity of each of these basic elements influence the capacity of the entire airport system and therefore should complement each other. However, the capacity of an airport may be evaluated from the most critical capacity of the two main components. Improvement in technology introduced larger and faster aircraft that needed longer runway lengths for takeoff and landing. The runway occupancy time [ROT] i.e., the time required to takeoff or land an aircraft increased because of increase in the size, speed and a few other aircraft features. This increase in the runway occupancy time of the aircraft implied that the capacity of the runway was limited. Thus, the airside became the critical component dictating the capacity of the airport. This limitation on the capacity of the airside was given due attention and until recently much of the research was concentrated on improving the runway capacity. Improvement programs were introduced successively to increase the capacity of the runway which in turn resulted in increased traffic. The runways are better equipped to handle this increase in traffic than ever before however, the landside, which was not improved consistently with the airside improvements, would soon become the restricting factor in the handling of the increased traffic in many airports around the world. In fact until recently at the Bangkok Airport in Thailand, the capacity of the airport was restricted by the poor ground access system. Such restriction in the capacity has prompted researchers to study the problem [FAA, 1987; Transport Canada, 1988; BAA, 1982]. These studies have
resulted in the development of some solutions for the problem however, unlike the standard methods developed for the airside component, the methods proposed for the landside improvement are not so widely accepted despite more than a decade of continuing research.

**Research Scope, Objective, and Approach**

The landside capacity is influenced by a number of elements such as the terminal building, ground access, parking, aircraft parking positions and gates, and so on. The first measure in alleviating the capacity problem is to identify the critical element or elements. This element (s) can then be studied to find the reason for inefficiency and a solution proposed. This process could be time consuming and may prove to be very expensive if the conventional methods of analytical/mathematical solutions are applied. The advancements in the field of computers offers a solution which could satisfy the requirements of the problem solving process and yet be not very expensive. A computer simulation model can very easily simulate the actual environment of the airport and may produce accurate results in less time. It is also possible to use the computers to simulate a wide variety of scenarios thereby checking the system for its responsiveness to these variations. However, a computer simulation model may have its own limitations. For example, accuracy of the output may depend on the accuracy of the input and erroneous processing may be hard to identify. Also excessive details may make the use of these models very difficult to program and usage will be limited to those who know programming. Even with these drawbacks, computer simulations are possibly the only means of getting quick and accurate results. The FAA [ALSIM] and several other private consultants [Peat Marwick Mitchell and Company, Mitre] have developed computers programs to simulate the landside environment but, these models have drawbacks such as the ones mentioned above and so are not very suitable for the general use of planning professionals.
This research aims at developing a quick assessment computer based simulation model *Airport Landside Planning and Simulation (ALPS)* using a software called *EXTEND* (Version 2, © Imagine That, Inc., 1992). This model is developed for the conceptual phase of the landside planning. Measures of performance are given for each facility planned in terms of the average wait times, average length of the queue, and the level of service (LOS) given as the area per passenger.

The airport may be defined as a system consisting of several components, whose capacity determines the capacity of the system. This makes the airport system very complex because the effect of one component may be felt by all the components thereby decreasing the efficiency of the entire system. This interaction of the airport components can be studied during the planning process by testing the environment for its reaction to various demand patterns. For a quick assessment model the requirement is that it should be effective in identifying the inefficient component or components within a very short time and with minimum information input for different demand scenarios. Some of the existing methods used by the FAA are based on analytical/mathematical approaches that may not be very useful in reasonably assessing the dynamic behavior of the airport system in a short time. This dynamic system may be simulated on a computer using a suitable simulation program. ALPS is programmed for determining the airport capacity at the preliminary design level and help in predicting potentially critical elements of the landside thus useful in proper planning of the terminal landside.
2 LITERATURE REVIEW

Introduction

The objective of this literature review is to present some background on past and current research on the landslide capacity determination methods. The methods reviewed here are those developed by the FAA [ALSIM], by private consultants [Peat Marwick Mitchell and Company] and by some international organizations [Transport Canada, BAA]. These methods range from analytical/mathematical to graphical and computer simulation models. This chapter briefly describes some of these methods. But before going into this discussion it is necessary to define certain terms such as capacity of a system/ component, level of service, flow rates, and crowding.

Defining Capacity

Landside capacity may be defined as the capability of an airport landside infrastructure to accommodate passengers, visitors, air cargo, ground access vehicles, and aircraft. Of the above, the capability to serve passengers is of greatest concern. Other aspects of landside operations such as cargo shipments and aircraft maintenance do not involve direct interaction with the passengers but serve to directly influence passenger service. A variety of demands are imposed by the passengers on the parking facilities, ticketing facilities, baggage claim, waiting/lounge area, and other landside facilities. These demands vary with the pattern of passenger arrivals at the airport, the number and size of luggage they carry, age of the passengers, purpose of the trip, number of visitors accompanying each passenger, and other external factors.

In terms of quantity, the landside capacity may be defined as the number of passengers who can be served by the airport landside components in a given period of
time. Passenger capacity is indicated by either 'Flow Rates' defined as number of passengers per unit of time, or 'Crowding' defined as number of passengers in a specific area during a given time. Flow rate capacity is indicated by 'Maximum Throughput' or a lower 'Service Volume' that results from 'Level Of Service' considerations, demand characteristics and some other internal factors. Crowding capacity is indicated by the 'Crush Conditions' which occur at maximum throughput or by a lower service volume that maintains service levels consistent with passenger comfort, convenience, safety and health.

**Airport Design Process**

The design process which has evolved through research over the years consists of two phases: 'The Conceptual Definition Phase' and 'The Detailed Design Phase'.

The conceptual definition phase involves the selection of an overall airport configuration and also addresses temporal matters such as what should be the terminal configuration (Whether it should be a box, or should it have finger piers, or should it have satellites, and so on). These decisions will be based on the type of airport i.e., International/Domestic, Transfer/Through, Originating/Terminating. Also the number of airlines expected to use the facility, the type of gate assignment (exclusive or common gate operation), proposed design horizon, and the phases of construction influence the decisions made in the conceptual definition phase.

While the conceptual definition phase mainly deals with the broad spatial and functional issues, the detailed design phase deals with design of specific elements whose configuration has already been defined in the conceptual definition phase. This phase addresses issues like determining the exact and final dimensions of the various airport components and allocation of available space to these elements. Special attention will be given to the custom design of facilities such as a baggage handling or security check system.
which need to fit within a given space or a particular orientation. Thus the objective of
analysis in this phase is to come up with precise answers which will provide specifications
for construction contracts, purchase of new equipment, and so on.

The design process involves the following four steps:

1) Forecasting traffic levels for peak hours,
2) Specification of level of service standards,
3) Flow analysis and determination of server and space requirements, and
4) Configuration of servers and space.

The following discussion briefly reviews the above mentioned design steps.

1) **Forecasting of Traffic Levels at Peak Hours**: To plan airport facilities to
meet future demands, it is essential to predict or forecast the level and the distribution of the
demand on the various components of the airport system. Thus the objective of this
exercise is to obtain detailed peak design hour demand scenarios for the design day of the
horizon year. These figures serve as a basis for an actual design.

**Methods of Forecasting**

Of the various methods of forecasting available, a particular method is selected
based upon factors such as use of the forecast, availability of data, degree of precision
required, resources available, time frame in which the forecast is required and is to be used,
and so on. However, in general the forecasting process involves estimation of the
aggregate traffic for the horizon year and design day which is normally taken as the 30th or
40th busiest day of the year. This is done using a set of conversion factors, partly based on
historical data. It is assumed that the pattern of traffic over the design year is predictable
and the design year is taken as a round figure for the sake of convenience.
Further assumptions are made concerning the type of aircraft, their origin or destination, load factors, percentage of transfer passengers, and so on, in order to develop hour-by-hour traffic scenarios for the design day, down to the level of a specific schedule of flights. The various methods available for this type of forecasting are forecasting by judgment of knowledgeable persons, by trend projection and extrapolation, by market analysis methods and by econometric modeling methods. A discussion of the details of these methods is out of the scope of this research report.

2) Specification of Level Of Service (LOS) Standards: The specification of level of service standards serve as a basis for translating the forecasts into an architectural program. Therefore, the objective here is to specify explicitly Level Of Service (LOS) standards for waiting times and space allocation that is the number of square feet or square meters per person at processing facilities, holding areas, passageways. These LOS standards are set through the agreement of the airport designer, owner or operator through consultations. Higher standards imply more space and hence, more cost. Thus, the standards are set in accordance with the financial objectives of the airport owner or operator. and therefore vary for each individual airport. However, the FAA provides for minimum standards for a facility which the designer has to comply with because these requirements are necessary for safety of passengers. Apart from the FAA, several other organizations such as International Air Transport Association (IATA), Airport Associations Coordinating Council (AACC), Transport Canada, etc., specify LOS standards which may also be adopted if applicable. These standards may range from LOS A (Best) to LOS E (Worst) as shown in table 1.

For passengers flowing through passageways or corridors standards set by Fruin/Benz (1971/1986) may be used. Typical values are for LOS(C)-12 Persons Per Foot Width Per Minute (PFM) and for LOS(D)-16 PFM.
<table>
<thead>
<tr>
<th>SUB SYSTEM</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Holding Areas With Baggage</td>
<td>1.6</td>
<td>1.4</td>
<td>1.2</td>
<td>1.0</td>
<td>0.8</td>
</tr>
<tr>
<td>Check-In, Baggage Claim Areas</td>
<td>1.6</td>
<td>1.4</td>
<td>1.2</td>
<td>1.0</td>
<td>0.8</td>
</tr>
<tr>
<td>Holding Areas Without Baggage</td>
<td>1.4</td>
<td>1.2</td>
<td>1.0</td>
<td>0.8</td>
<td>0.6</td>
</tr>
<tr>
<td>HoldRoom Pre inspection</td>
<td>1.4</td>
<td>1.2</td>
<td>1.0</td>
<td>0.8</td>
<td>0.6</td>
</tr>
<tr>
<td>Waiting Areas / Circulation passageways</td>
<td>2.7</td>
<td>2.3</td>
<td>1.9</td>
<td>1.5</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Table 1

**LOS STANDARDS (SQ.MT./OCCUPANT) [AACC/IATA 1981]**

[Odoni and Neufville, 1990]
In setting the above mentioned standards it is assumed that the space provided for a particular activity such as ticket counters, lounges, etc., will be useful no matter how or where it is provided, and that the occupants of a space make uniform use of the entire area. Therefore it is evident that no consideration is given to people collecting at specific places such as news-stands, ticket counters, information booths, etc., which might result in discrepancies in the design.

3) Flow Analysis and Determination of Server-Space Requirement:

The different methods used to analyze the flows and determine the amount of space and the number of servers required are as follows:

- Application of Queuing Theory
- Graphical Analysis
- Computer Simulation

Queuing Theory is widely used for estimating the processing and delay times from which the required sizes of facilities and operating costs may be derived. Almost all the passenger handling systems may be modeled as service facilities using queuing models. Since passenger arrival pattern is a random process and because steady state conditions seldom occur at airports, probability theory is used for generating mathematical functions such as Poisson distribution, representing the arrival and service performance of the system. These mathematical formulations may not be very precise because of the variability associated with the passenger behavior and also because the queuing theory is more accurate for steady state conditions whereas, the airport environment is highly dynamic. The only exceptions where steady state queuing formulae do prove useful are cases in which either parallel processing facilities are shared by a number of airlines and roughly constant rates of demand at high server utilization levels are achieved for significant periods.
of time or the facilities are quite underutilized. However, reasonable estimates of delay and
the corresponding queue lengths may be obtained using simple formulations such as:

Measure of Effectiveness for a multi-server queuing system with poisson arrival
distribution and exponential or constant service time are:

\[ L_q = \frac{(\rho S)^s - \rho P_0}{S! (1-\rho)^2} ; \quad W_t = \frac{L_q}{\lambda} \]

- \( L_q \) = Average queue length
- \( W_t \) = Average delay or waiting time
- \( \lambda \) = Arrival rate (passengers / unit of time)
- \( \mu \) = Service rate (passengers / unit of time)
- \( \rho = \frac{\lambda}{S\mu} \) (utilization factor), which is less than 1.
- \( S \) = Number of Servers
- \( P_n \) = Probability of a Queue occurring; \( \frac{(\rho S)^n P_0}{n!} \) for \( n \leq S \);
  \( \frac{(\rho S)^n}{S!} P_0 \) for \( n \geq S \)
- \( P_0 \) = Probability of a Queue not occurring; \( P_0 = \sum_{n=0}^{S-1} \frac{(\rho S)^n}{n!} + \frac{1}{S!(1-\rho)} \)

- \( n \) = Different States of a Queuing System

Graphical Analysis of the cumulative arrivals and service have proven to be effective
in analyzing and designing specific components of the airport system such as, departure
lounges, ticket counters, security counters, and so on. The basic assumption in this process
is that the pattern of loading is known, hence this method is best suited for the redesign of
an existing processing facility mainly for expansion purposes. The major disadvantage of
this method is that it cannot be useful in designing facilities where the arrival flows into
each facility is interdependent often in complicated ways.
Modeling and simulations in principle provide the means for investigating the flows through an entire processing facility. Simulation modeling can be used as very effective tool for developing solutions for the dynamic environment of airport planning process. In the last couple of decades simulation models were developed by academic universities [ACAP, 1978; AIRSIM, 1973; AIR-Q, 1972], by government agencies [ALSIM FAA, 1982; Transport Canada, 1973; British Airport Authority (BAA), 1982] by industry consulting firms [TAMS, 1972; Bechtel, 1974; Battelle, 1974; ATSIM, 1987; and Peat Marwick Mitchell & Company, 1987]. Most of these models were developed for the specific requirements of a particular airport and their database could be useful for only a few handful scenarios. Moreover, these models were developed based on simulation languages which at that time were not very responsive in the user friendly point of view. These models require extensive programming and change of input in data for the few scenarios they can simulate. Such drawbacks have limited the use of these models and forced researchers to look for better methods of terminal simulations.

One of the important reasons for the failure of the above mentioned simulation models is the choice of a proper programming language for the model construction. Some of the models mentioned above are programmed in general purpose languages, e.g., FORTRAN, Pascal, or Basic. These languages are not very well suited for simulation purposes. Therefore, it is advantageous to program a simulation model in a simulation language than in a general purpose language. Some of the advantages of simulation languages over the general purpose languages are:

- Simulation languages automatically provide most of the features needed in programming a simulation model, resulting in decrease in programming time.
- Simulation models can be easily modified if they are written in a simulation language.
- Most of the simulation languages have built in error detection features which detect and check errors automatically.
On the other hand, the general purpose languages also have certain benefits of use over the simulation languages. These are:

- More people have knowledge of general purpose languages than a simulation language making it easier to write a program.
- A simulation language requires special infrastructure whereas, a general purpose language is available on any computer.
- A program written in a simulation language is generally designed to model a wide variety of systems with one block whereas, a program in a general purpose language can be written specifically for a particular situation.
- General purpose languages have more flexibility than simulation languages.
- Software costs may be lower with general purpose language models.

These factors should be borne in mind while making a choice of a programming language. Some of the simulation languages in use are GPSS, SIMAN, SIMSCRIPT, and SLAM [Law and Kelton, 1991].

4) **Configuration of Servers and Space Requirement:** This phase involves space allocation for each facility and other non-revenue generating areas such as the rest areas, public service facilities and airport operation and management offices. The sequence in which the servers need to be oriented needs to be studied and the area provided for each facility should have flexibility to modify according to the changing requirements. The building architects need to be consulted for making decisions in this step. The final selection is made after testing the different orientations for flexibility to modify and yet satisfy all the level of service criteria for present and future demand levels.
DESCRIPTION OF SELECTED SIMULATION APPROACHES

Airport Landside Simulation Model (ALSIM)

ALSIM [FAA, 1982] is a airport landside computer simulation model developed by the FAA for simulating the landside environment. ALSIM is a macroscopic, probabilistic, discrete-event, fast-time model capable of producing flow and congestion parameters and statistics on simulated facilities and works on a IBM/370 mainframe. It is a FORTRAN based model and consists of main and auxiliary programs written in GPSS-V, which creates transactions representing passenger and visitor processing and directs them through the model blocks that describe the system. The supporting subprograms provide flight schedule, airport configuration data, matrix manipulation, and assignment of facilities to GPSS-V created transactions. The subprograms of the assembly language of IBM/370 mainframe provides linkages between GPSS-V and FORTRAN subprograms.

The model input consists of: flight schedule, passenger characteristics, airport geometry, and facility information. The first two represent the demand and the last two represent service characteristics of the system. ALSIM produces a statistical report of each facility involved in the simulation. The output report includes total number of persons served, maximum and average number of servers busy, the occupancy, and flow of passengers through the system. The model assumptions are:

• Passenger and visitor facilities are similar and independent of the type of airport,
• Random functions govern transfer flight selection and baggage delivery times,
• Service time distributions are independent of time and server workload,
• Single queue lines are used to represent multi-server queues at each facility,
• People proceed directly from one facility to another,
• Exogenous flight schedule provides the time-varying demand, and
• Arrival rates and operations of facilities are determined by the model itself.
ALSIM has a capability to simulate a 100-gate airport during a busy five-hour period involving 20,000 passengers on 165 flights. The model running time (CPU time) for such a facility is 7 minutes with GPSS-V and FORTRAN IV compilers and requires a 570K bytes of core capacity [Mumayiz, 1992].

**Canadian Airport Planning Models**

The models developed by Transport Canada for assisting planners and designers in assessing demand, capacity and levels of service are useful for airport terminal and groundside facilities (Transport Canada, 1988). The three models developed for this purpose are the gate assignment model, air terminal passenger flow simulation model, and ground transportation simulation model. These models are interactive, interrelated, and mutually compatible separate models and can run on an IBM-AT compatible microcomputer. A brief description of each model follows.

*Gate Assignment Model*: The gate assignment model is a deterministic multi-channel queuing model that operates at five minute intervals. The user specifies the gate assignment strategy and the flight schedule according to which the model assigns gates to flights as they appear in the flight schedule. A maximum of 300 flights can be input in the flight schedule as created by a computer editor and the user specified airport description. The airport description includes aircraft class by seating capacity, carrier preference (max. 30 carriers), gates (up to 100), aircraft service times at a gate, gate conflict (max. 50) defining any intergate restrictions caused by aircraft or gate size incompatibility, and time equal to the aircraft push-out maneuvering time.

The output of the simulation includes a two-way passenger flow through a gate, a Gantt Chart assigning aircraft to gates, statistics on gate utilization and aircraft and passenger delays, and a step chart plotting the cumulative number of aircraft on gate in each five minute interval over the simulation time or the time span of the schedule.
**Air Terminal Passenger Flow Model**: The air terminal passenger flow model is an interactive event-oriented stochastic simulation model that simulates flow of passengers along predefined paths from curb to aircraft and vice versa. The model consists of three interactive modules: the airport terminal description data entry and edit module, the simulation module, and the statistics report generation module. Special data files are created using the data entry and edit module for the input which includes:

- Identification of carriers, sectors, terminal users, aircraft types, passenger types, and baggage claim;
- Description of layout of air terminal building by identifying gates, links, processing facilities, separators and meeting areas, baggage dispensers, waiting, and holding rooms;
- Arrival distribution tables and processing rate distributions;
- Sequential lists of node numbers identifying paths used in the terminal;
- Ratio tables of visitors to passengers by hour of day and by sector; and
- Flight schedule that includes 17 different attributes for each flight.

The information provided by the entry module is used to process the passengers passing through a particular part of the model. The simulation module gets activated and acts on each element of a flow path representing a part of the terminal building. It then assigns a next event time to the simulation and moves the elements through a sequential flow along a predefined path. Each transaction is associated with 19 attributes that are used in the logical control of the transactions when simulation progresses as described by the airport description file. When a transaction enters a processing facility such as a baggage area, a probabilistic match is performed on passenger arrival rates, baggage arrival rates, and bag to passenger ratio. The statistical report generation module then accesses the output files created by the simulation module to print the output data specified by this module. The simulation output includes the following:
• The cumulative number of users entered and exited, present count in facility, maximum accumulation of users, and time of maximum accumulation for each facility;

• A summary of baggage device assignment statistics;

• The queue statistics for each processor including total and maximum user queue time, number of users in queue, frequency distribution of queue times, and flight causing maximum congestion; and

• A summary of delay statistics for each flight.

*Ground Transportation Model*: The ground transportation model is an interactive program used to simulate the flow of vehicular traffic along access and egress road systems of the airport. A description of the road network, vehicular flow patterns and paths, and behavioral characteristics of the vehicles on the road system is used to simulate the movement of traffic on roads, parking lots, and curbs. The simulation approach applies the survey statistics and flow path information to each flight in the schedule, employee schedule, visitor schedule, and cargo schedule. Vehicle transactions are created as they move through the road link system. At each portion of the system transactions are acted upon, assigned a next event time, and moved along their predefined paths. Statistics are gathered whenever a vehicle enters or exits an element of the system of road network. The data input includes:

• The road link system and associated transit times;

• Vehicular and user flow along the road link system;

• Time distribution relating vehicle creation to flight arrival and departure times;

• Visitors to passenger ratios, modal split, average vehicle occupancy by mode, curb processing distribution, and mass transit schedule and demand patterns;

• Aircraft flight schedule; and

• arrival and departure schedules for employees, visitors, and cargo.
The output provides information on any combination of the eight attributes i.e., current position of the transaction (node), next node, path, transportation mode, trip purpose, user type, sector, and flight number. Output statistics include time interval occupancy counts for each link: time, total number of vehicles entering and exiting the time interval, present vehicle count at end of time interval, and maximum vehicle count for time interval and time of its occurrence.

**Alternative Simulation Method for Individual Facilities**

SLAM (Simulation Language for Alternative Modeling, Pritsker and Pegden, 1979) was used to simulate individual processing facilities of the airport terminal and is considered as a microscopic approach requiring minimum input. Each facility is modeled separately by writing a short program consisting of SLAM network mode statements that best describe the operation of the facility. The major input includes number of channels (servers), processing rate of each server, service discipline, and arrival distribution of demand. A SLAM user functions describe arrival distribution, queueing process, and activity process involving random distribution of processing times.

The output includes statistics on processing operations at a facility at any desired time or time interval. Output statistics data such as maximum/minimum length of the queue, average wait times, variation of queue lengths, queue times, server utilization and number of entities in each facility can be gathered.

SLAM may be used for macroscopic modeling also but then it will involve more detailed programming and the structure of the model will become very complex. The macroscopic approach can be used to establish performance measures for models representing individual facilities at varying service volumes and demand levels. Separate FORTRAN segments are needed to relate the various portions of the model and provide the means to input arrival distributions at the system boundaries.
3. METHODOLOGY

Introduction

This chapter briefly discusses the methodology used for conducting this research. As mentioned in a earlier chapter, the aim of this research is to develop a Quick Assessment method for planning and evaluation of the airport landside in terms of its capacity and level of service. It is proposed to use a computer based discrete-event simulation program to build the model. Therefore, this chapter is devoted to the definition of simulation and the description of the modeling process. EXTEND\textsuperscript{1} [© Imagine That, inc., 1992], the software program used for developing the proposed model will be discussed as a simulation tool.

Simulation

Before we discuss any further it is necessary to know the difference between modeling and simulation. A model can be thought of as a ideal representation of a reality, as some subject of inquiry that may be already in existence, or as a conceived idea awaiting implementation [Mumayiz, 1992]. Models are primarily used because they represent a reality with which the model may have a pertinent relationship moreover a model may be easy to manipulate than the real world system. Also manipulating and controlling a real world system may be very expensive and sometimes impossible.

Models should be simple and represent the actual system as accurately as possible in order to obtain a reasonable accuracy in their results. Normally a large number of variables are required to accurately represent a real world system but, only a small number of variables usually account for most part of a model. The reliability of the results obtained

\textsuperscript{1} Extend is a trademark of Imagine That, Inc., San Jose, CA
from a model depend on the validity of the model in representing the assumed real world system. The level of detail also contributes towards the reliability of the results. There are different kinds of models. For example, analogue, heuristic, iconic, simulation and symbolic models. Simulations are models that use mathematical and logical representations of the real world system to convert system descriptions or input parameters into output that describes some of the features of the system. Simulation is widely used in different fields of studies hence, different definitions to explain it. However, it suffices here to say that simulation is a process of designing a model of real world systems and conducting experiments to either understand the behavior of the system or evaluate various strategies to operate the system. Inferences can then be drawn about the system without the need to physically build, disturb, or destroy it.

The functional types of simulation models are

- Analytic queuing models are based on probability and use mathematical expressions derived from the queuing theory.

- Accounting models are time based and deterministic in nature. These models use predefined rules to describe the state of the system.

- Time dependent models are event based and stochastic in nature. These models use dynamic equations with mathematical-logical representations or Monte Carlo methods for fast time reproduction of the state of the system.

The first step in developing a simulation model is to define the conceptual framework which would describe the system to be modeled. In other words the functional relationships of the real-world system are described as they are perceived. This definition of the real-world is called the 'world-view'. Simulation generally involves two types of world-views: 'Discrete' and 'Continuous'. In discrete-event simulation, the system is described by the change in the state of the system occurring at discrete time intervals (event
times). The state of the system is assumed to remain unchanged between these time intervals i.e., the variables describing the system change only at certain time intervals. In contrast, the continuous-event simulation assumes that the variables representing the system change continuously over time. The behavior of the system is characterized by equations for the set of variables whose dynamic nature simulates the real-world system.

The dynamic environment of an airport can only be described by a discrete-event simulation because, the airport involves too many variables and therefore it is almost impossible to derive the set of equations required for the continuous-event simulation. The following paragraphs briefly describe the different types of simulations.

**Discrete-Event Simulation**

Discrete-event simulation is of three general types depending on the specific features of the simulation. These are event-oriented, activity-oriented, and process-oriented simulations. The objects or basic units within the boundaries of a discrete-event simulation are called 'entities'. Each entity has its own characteristic called 'attribute'. Attributes may be shared by different entities who engage in different kinds of 'activities'. A 'process' involves several activities performed in a time ordered sequence. A simulation system (language/program or a package) could be one of these three general types or it could be an object-oriented simulation, or it could be a hybrid simulation. Each type of these simulations will be briefly discussed here.

**Event-Oriented Simulation**

A system is modeled by defining changes occurring at event times and by determining events that can change the state of the system, and then by developing the logic associated with each event type. Simulation is run by executing the logic associated with each event type in a time ordered sequence.
Activity-Oriented Simulation

Models are built by describing activities engaged in by entities of the real world and prescribing conditions that cause each activity to start or end. These conditions must be scanned continuously to ensure that each activity is accounted for. This process of following the activities of each entity may result in a very cumbersome structure and reduce the computer handling capacity.

Process-Oriented Simulation

This type of simulation is a combination of the event-oriented and the activity-oriented simulations. Because of this combination this simulation is very easy to use. This approach involves a model representing a real-world system defining the flow of entities through the system on the basis of sequences of events that occur in a predefined pattern, the logic of which can be generalized and used in a single statement. In this case the event logic is implicit and automatically contained in the corresponding statements. A model may be constructed by combining a set of standard blocks that map the logical structure, of the simulated real-world system.

Object-Oriented Simulation

Model development consists of a highly modular approach to provide simple, unifying programming and prevents extensive intertwined subroutine coding. The basic entities in object-oriented simulation are objects with attributes that have values. Each object has rules and procedures associated with the values. These objects have the capability to communicate with each other through messages. Upon the arrival of messages, the values containing the rules and procedures process the messages and carry out their effects. The major advantage of this approach is that it reduces the amount of programming and coding required for a real-system representation.
Hybrid Simulation

Hybrid simulation systems incorporate the advantages of simulation approaches in a unified modeling framework. In other words, a model built with this system can operate in the discrete-event, network, and continuous-event simulation modes in different models or even in one single model. EXTEND falls under this category. However, it may be noted here that a model can basically be of either discrete-event type or continuous event type with the incorporation of a few ideas common to both types.

Desired Properties Of A Simulation System

In order to build and run simulation models efficiently, the simulation system should have certain minimum requirements which would allow for built-in facilities and capabilities. These minimum requirements are:

1. The change of state of a system during an event should be easy to describe.
2. The scheduled events should occur relative to the independent variable time.
3. Should have a built-in capability to generate random variables and random functions.
4. The system should have general arithmetic capabilities and facility for gathering statistics and controlling experiments in the system.
5. It should have interfacing capabilities with other computer segments.
6. The system should have easy to use debugging features.

Some of more desirable features fast becoming popular are

1. Graphics capability to display output of statistical analysis of experiments.
2. Real-time animated graphics of the simulated system which indicate the entities entering the system, going through different processes and then exiting through the outputs.
3. Programming and editing capability to facilitate entering and manipulating data structures, programs and coding.

4. Help functions to provide brief useful information on various features of the simulation system while it is in use.

SIMULATION IN THE AIRPORT LANDSIDE ENVIRONMENT

Simulation is almost an indispensable tool for planners investigating the problems of an environment such as the airport landside because of its convenience, reliability, and efficiency in analyzing and describing activities in detail and in a manageable fashion. The characteristics of an airport simulation model can vary depending on the objective of the simulation, features of the simulation, and modeling approach. The basic properties of simulation could be analytic versus numeric, static versus dynamic, discrete versus continuous, deterministic versus stochastic, or closed versus interactive. A discussion of these properties is outside the scope of this report however, it should be noted here that the Airport Landside Planning and Simulation (ALPS) model is based on the discrete-event processing.

The central step in the landside capacity assessment, planning and management process is capacity and level of service. The level of detail of a landside simulation model is determined by the data on demand characteristics, operating conditions and community factors. The simplicity, reliability and efficiency of a model depends on the purpose of the simulation, the characteristics of the model, degree of detail in representing the real world system and the level of precision required. The modeling process will depend on achieving a balance between simplicity, speed and ease of use on one hand and accuracy, need for data, and cost on the other hand. Figure 1 shows the conceptual framework of landside capacity assessment, management and planning process as proposed by the TRB[1987].
Figure 1. Landside Capacity Assessment, Management and Planning (Special Report 215, TRB, 1987)
In general for model development, the tradeoff is made between accuracy and cost-effectiveness. The airport landside capacity is generally measured in terms of an airports individual functional components. However, the capacity of individual components may fail to give the important functional linkages within the airport landside.

The airport terminal can be modeled on two scales: (1) macroscopic, as one integral system from groundside to airside and (2) microscopic, where individual facilities are considered separately. ALPS is a microscopic model where each facility is considered separately for capacity estimation. Although it needs a considerable data input for the first run, it will be only a one time data entry and later runs would need only little data modifications reducing the input work load.
EXTEND AS A SIMULATION TOOL

Extend (version 2) is an easy to use advanced simulation tool for decision support. Extend uses building 'blocks' to create models and is similar to SLAM (simulation language for alternative modeling). A process can be described by the creation of a block diagram where each block represents one part of the process. The block diagram can be laid out in a two dimensional drawing environment. The advanced features of the system like animation, graphical interface, dialogs and notebooks make the system equivalent to a moving picture which unlike some other simulation packages does not represent only a snapshot of a system or a program during a simulation run.

Some of the important features of Extend are

1. Rapid construction of models using an array of building blocks
2. Animation of the model for better presentation
3. A customizable graphical interface to show the relationships in the model
4. Hierarchical modeling to make complex systems easy to build and understand
5. Changes in model values through dialogs and notebooks
6. Ability to adjust settings while the system is running
7. Connectivity with other programs through copy / paste, import / export, text files, XCMDs, publish / subscribe, etc.,
8. Monte Carlo, batch mode and sensitivity analysis for system optimization

Additional advanced features of extend are

- Extend allows modeling in continuous, discrete event, linear, and non-linear dynamic models giving it a multi-purpose simulation environment
- Extend has built-in C like scripting language to allow any modifications the user may want to make in the blocks or create customized blocks

28
• The building blocks in extend are provided in libraries and can be reused in some other models. Also the customized blocks can be saved in these or new libraries
  • Building blocks can send messages to other blocks interactively for sub processing
  • Extend is capable of passing data such as values, arrays, or structures composed of arrays of arrays
  • Extend offers direct access to functions for integration, statistics, queuing, animation, math, matrix, sounds, debugging, XCMDs, etc.,
  • On-line Help available from desk accessory menu and from help button in a block
  • The model size has no limitations except those offered by the computer system

Some of the libraries that come with Extend are generic, discrete-event, animation, and plotter library. Custom libraries for special simulations such as electronics, manufacturing and so on are also available. For our purposes, it is sufficient to have only the first four libraries. A brief description of each of these libraries is given below.

Continuous simulations are analogous to a stream of fluid passing through a pipe. The volume of flow may increase or decrease but the flow is always continuous. The Generic library allows simulation of almost any continuous modeling problem with the use of a few blocks. The Generic library contains blocks that perform such basic functions as math, decision handling and input/output. It is possible to use these blocks to build a model in the areas of finance, economics, electronics, demographics, biology, chemistry, physical systems, physiology, and many more.

Using the same analogy of fluid flow in a pipe, a discrete-event simulation would be able to simulate the scenario of either the pipe being empty or has something traveling down it. Whether something will come out of the pipe depends on an event occurring at the
other end. In a discrete-event simulation the state of the model changes only when events occur and the simulated time advances from one event to the next. Mere passing of time has no direct effect on the simulation. The Discrete-Event library contains all the basic tools for creating models that use queues, servers, item-specific attributes, and priorities. These blocks are useful for simulating queuing theory, paper flow, computer networks, service industry waiting lines, and so on.

The Plotter library holds blocks for plotting the results in any type of model. In besides the simple plotters, these plotters can show scatter plots, moving strip charts, and so on.

Animation library holds blocks that may be used for animating hierarchical blocks. These blocks serve as an important debugging tool.

**Model Construction Process**

Model construction is done by simply selecting blocks from appropriate libraries on to a model worksheet and then the blocks are connected in proper order by connecting the input of a block to the output of next block and vice-versa. It is possible to achieve the same results by using different blocks in different order and, it is up to the modeler to select the optimum combination that will make the model structure simple and reduce the run time. Extend allows the use of blocks from any library in any type of model however, the blocks from discrete-event library can only be used in a discrete-event simulation because these blocks tend to change the timing of the model thereby making it discrete-event and not continuous.
4. MODEL DESCRIPTION

Introduction

This chapter deals with the description of the model ALPS (Airport Landside Planning and Simulation) developed as a result of this research. The description includes the boundaries of the model, assumptions, input procedure, Extend building blocks, hierarchical blocks, and the running of the model.

Model Environment

ALPS is a simulation model based on a highly graphical interface. It visually reflects the real-world airport landside environment through its block arrangements. The model is constructed in two parts based on the Enplaning process and the Deplaning process. All the blocks arranged in the model represent the actual flow pattern of the landside environment. The amount of detail included in the model is based on the purpose of the model. Elements of the real world environment which cause a significant difference in the decision making process only are considered for the simulation.

The elements of the airport landside and the interface between the airside and landside included in the model and which define the model boundaries are:

1. On the Landside
   a) Terminal Curb Frontage used for loading and unloading passengers, visitors and baggage. This element serves as the boundary for the landside, ground access interface.
   b) Parking for vehicles. Private and rental vehicles need parking facilities at the airport for loading/unloading or pickups/drop-ins.
2. On the Airside interface

a) The **Boarding Gates** which allow passengers to board the airplane define the boundary of the model on the airside. This does not include the gate assignment for arriving airplanes.

**Model Assumptions**

Assumptions are made to simplify the model construction and yet not lose the essence of the real-world system.

1) Simulation time is specified by the user for a particular running of the model.

2) The enplaning and deplaning passenger flows are never mixed and are treated separately.

3) The passenger and aircraft arrival patterns are generated independent of each other by a user defined distribution.

4) Passenger arrivals are generated in the enplaning model at the curb frontage and in the transfer passenger generator whereas, aircraft arrivals are generated in the deplaning passenger model at the individual arrival gates.

5) Each facility may have a server-queue system. The user defines the number of servers, service time for each server, total area of the facility and length of the queue limitations. Total area of a facility does not include the area occupied by equipment and office space. Total area includes only the area provided for passenger circulation and waiting areas near service facilities.

6) If the arrivals are more than the processing capacity of a facility then, arriving entities are put into a queue and wait times for each entity are noted.

7) To get the best results, the demand pattern is specified for the worst conditions or according to the horizon year busiest day - busiest hour forecasts.
8) Number of gates, their sizes (Heavy, Medium, and Small) are defined by the user for both the enplaning and deplaning models separately. The range of capacity of a gate is arbitrarily fixed for simplicity purposes. For example:

- Small Aircraft Gate = 1 - 100
- Medium Aircraft Gate = 101 - 250
- Heavy Aircraft Gate = 251 and above

**ALPS Model Features**

Figure 2 shows the Enplaning Passenger model. For the sake of simplicity a basic model consisting of all the facilities required for an international airport is modeled. The model can be opened in Extend by selecting open file command from the file menu and then selecting the name of the file from the list of files displayed. Extend then opens all the libraries (Generic, Discrete-event, Animation, & ALPSLIB) containing the blocks used in the model. If a new model command is selected from the file menu then, the libraries necessary for the simulation need to be opened by selecting the open library command from the library menu. ALPSLIB is a library containing the hierarchical blocks and some standard library blocks specially modified to suit the needs of the landside simulation.

Each block in the model performs a specific function. Although a block may be used more than once in a model each block can have different input and a corresponding output. For example, Figure 2 shows that the circulation block is repeated. In the first block the delay may be 'x' units while the second block has a delay of 'y' units. The executive block controls discrete-event simulations and should always be placed on the left of all the other blocks. The ReadMe icon when double clicked opens up to show help text which briefly explains the flow of the model and also the functions of each block. Most of the blocks are hierarchical blocks i.e., each block is actually a sub model performing a specific function. The structure of a hierarchical block (Figures 5-21) or the dialog box of a
Figure 2. Enplaning Passenger Model
Airport Landside Planning and Simulation Model (ALPS)
standard library block is revealed by double clicking on the box. Any changes that the user may want to make can be entered in the dialog box or the structure window. A Block can be added to a model by simply selecting the library menu from the main menu then clicking a library will open the library window. The required block can be simply dragged on to the model worksheet. To remove a block from the model simply select the block which will then be darkened to show that it has been selected then, hit the delete button on the keyboard to remove it from the model worksheet. Deleting a block will cause all the connections to the block to disappear therefore, caution should be observed before deleting a block.

**Enplaning Passenger Model Flow Description**

The model flow progresses from left to right. The first block Curb Delay block simulates the activities of a terminal curb frontage. Vehicles arrive at the curb to load or unload passengers, visitors, and baggage. The delay for each vehicle is different and depends on the type of vehicle (taxi, bus, limousine, or a private car), the number of passengers or occupants, the age of occupants, the number and size of their baggage, etc.,. After unloading the passengers, the vehicles either go to the parking facilities or out of the airport area.

The Parking / Exit block simulates the parking lot scenario. Vehicles (entities) enter the block as the simulation progresses and occupy the allotted parking spaces. Each vehicle is delayed for a different time and after the assigned delay, the vehicle exits the system. This block takes the entities which do not enter the parking lot out of the airport system.

The entities coming out of the curb delay block are vehicles and need to be converted to their respective passengers or occupants. The Passenger Generator block performs this conversion. Each entity that enters the block is given a value equal to the
number of occupants of the vehicle. The passenger group then proceeds to the entrance of the terminal building.

Meanwhile the Transfer Passenger block generates the transferring passengers and sends them to the entrance of the terminal. The arrival pattern and the passenger group size of the transfer passengers can be specified by the user.

Upon entering the terminal the passengers are delayed for a certain time which corresponds to the activities performed in the waiting lobby. The Circulation Lobby block simulates the activities performed by the passengers in the waiting/circulation lobby. These activities may be making a decision, shopping, beverages / coffee shop, books / newspaper reading, or simply waiting. Therefore, each passenger has a different delay time which the user specifies. For the sake of simplicity, the delay caused by each of the activities inside a circulation lobby are combined and not taken separately.

The Ticket Counter block simulates the ticketing facility. The user can specify the number of ticket counters and the area for passengers waiting in lines. The passengers entering this block enter a queue and then wait for a counter to free up where they will be processed. The delay for each passenger varies and depends on the amount of baggage carried, whether he already has a ticket or needs to purchase a new one, whether it is a domestic flight or an international flight, and also the efficiency of the ticketing clerk. The user should take into account all of these factors before setting the delay caused to each passenger. It should be noted here that the visitors entering this block are not delayed and can go on to the next facility without experiencing any delay.

Passengers and visitors proceed to the security counters after completing the ticketing and baggage check-in process. Each person enters a queue and waits for a security counter to free up. The delay for each passenger depends on the efficiency of the security guard.
The entities exiting from the security area then proceed to the concourse which again is simulated in the circulation block. The People entering this facility are delayed for the time they take to traverse to the departure gates and in other miscellaneous activities such as shopping, having beverages or simply waiting for the flight boarding call. Again the user should set the delay by taking these factors into consideration.

Once the boarding call is given, the passengers proceed to the holding rooms for boarding pass check-in and boarding the aircraft whereas the visitors leave the system and exit through the exit doors.

The activity inside the holding room or waiting areas before passengers enter the aircraft is simulated in the Departure Gate block. As soon as the boarding call is given the passengers begin to enter the holding area. For the simplicity of model it is assumed that after experiencing a proper delay in the concourse, the passengers directly enter the holding area and no specific action is taken as to making a boarding call. The passenger entering first in the holding area experiences less delay.

**Deplaning Passenger Model Flow Description**

This model simulates the deplaning passenger process and the activities associated with it. Figure 3 illustrates the model setup.

The Gate Assignment block assigns the arriving flights to specific gates as per a predetermined order specified by the user. The Deplaning Passenger block receives the output of the gate assignment block and simulates the deplaning process at a gate. The gate size corresponds to the three aircraft groups. This block introduces arriving passengers in one single group and allows them to wait in a queue before they disembark the aircraft. The passengers have to walk through the boarding bridge towards the concourse. Attribute values that help the passengers locate their baggage in the baggage claim area is assigned. The attribute value varies depending on the number of baggage carousels available.
Figure 3. Deplaning Passenger Model
Airport Landside Planning and Simulation Model (ALPS)
The passengers have to pass through the immigration and naturalization check after deplaning from the aircraft. The Immigration Block simulates this process. This block should only be used for simulating international flight arrivals. The passengers entering this facility have to wait in a circulation. This gives them a chance to produce the required documents and to make sure everything is in order. Passengers then enter the line for an immigration counter and wait for their turn. The waiting time for each passenger depends on the efficiency of the checking clerk and the length of the queue.

The next step for the passengers is to claim their baggage. The Baggage Claim Block accomplishes the simulation of this process. Passengers entering this block have to pass through a circulation area before they arrive at their designated baggage carousel. Once they reach the respective baggage carousels, they have to wait for their baggage to arrive at the carousel. The delay experienced in walking in the circulation area, waiting for the baggage to arrive at the carousels, and collecting the baggage is all put together and specified by the user.

After collecting the baggage, passengers have to get their baggage checked through the customs check. The Customs Block simulates this process. This block should only be used for international flights arrivals. This block has two types of counters, Red and Green. The passengers who have custom taxable items in their luggage pass through the red channel and those who have no such items to declare use the green channel. The passengers using the red channel experience more delay than the ones using the green channel. This facility is provided to decrease the processing time and consequently the delay. The user has to specify the number of passengers using either of the two channels.

At this point, the transferring passengers are separated from the terminating passengers. The transferring passengers are then simply taken out of the system through an exit and each passenger is accounted for. The terminating passengers proceed towards the waiting area provided for circulation and to hold visitors / greeters. This activity is
simulated by the Circulation Block. The delay experienced by the passengers in this block includes the delay for walking, for using facilities such as rest-rooms, telephones, renting a car, getting information, and waiting for a mode of transport. After these activities passengers are allowed to go out of the system through the waiting area exits. This completes the deplaning process.

Figure 4. shows a list of the blocks used in the construction of both the enplaning and the deplaning passenger models. These blocks were custom created specifically for the purpose of constructing a airport landside simulation model. Figure 5 through 21 show the structure of all the hierarchical blocks used in the models and are listed in a alphabetical order.
Figure 4 Alphabetical List of ALPSLIB Library
Alphabetical Block Reference for ALPSLIB Library

**Actimple (LOS) Block (ALPSLIB)**

This block holds many items and passes them out based on the delay and arrival time for each item. The item with the minimum total delay and arrival time is passed out first. The delay time for each item is set in the dialog box or through the delay connector(D). This block can be used as a processing facility such as a ticket counter. The number of counters is set in the dialog box (maximum number in activity). The block can output information such as average delay for each item and average length of the queue.

**Area of Facility Block (ALPSLIB)**

This block generates a constant value at each step. This value represents the area of a facility and can be specified in the dialog box. This block is useful for calculating the level of service at a facility.

**Baggage Claim Block (ALPSLIB, Fig. 5)**

This is a hierarchical block and represents a baggage claim facility. Passengers are directed towards their respective baggage carousels and are delayed for the time required to collect their baggage. Delay is set in the activity block in the structure window and capacity of the facility is defined by the user. The block outputs average delay & LOS (area/pass.).
Circulation Lobby Block (ALPSLIB, Fig. 6)

This block represents a circulation lobby / waiting area. The capacity and delay for each passenger is set in the activity block in the structure window. The area of the facility and simulation time are specified by the user in the corresponding blocks in the structure window. This block outputs the average delay for each passenger and also the LOS in terms of the area / passenger.

Curb Delay Block (ALPSLIB, Fig. 7)

This block represents curb activity. Vehicles are generated in the generator block in the structure window and are put in a queue. The delay for each vehicle is specified by the user in the activity block. The output is the average delay for vehicles and also the LOS in terms of the minimum number of curb spaces required to satisfy the demand.

Customs Counters Block (ALPSLIB, Fig. 8)

This block represents a customs checking facility. Passengers arriving in this facility use one of the two green or red channels. Passengers using red channel experience more delay than the ones using the green channel. The delay for passengers using either of these channels is set by the user in the activity block in the structure window. The block outputs average delay for passengers and also the level of service in terms of the area per passenger.
Deplaning Gate (Heavy) (ALPSLIB, Fig. 9)

This block represents the heavy aircraft arriving at a gate. Deplaning passengers pass through the loading bridge towards the terminal. Delays are experienced inside the aircraft because all passengers cannot disembark from the airplane at the same time. Passengers are further delayed for the time they take to walk towards the terminal. Arrival time between two consecutive flights and the capacity of each flight can be set in the program block in the structure window. Delay inside the airplane and in the loading bridge can be specified separately. Output is the average delay for each passenger.

Deplaning Gate (Medium) (ALPSLIB, Fig. 10)

This block functions exactly as the Deplaning Gate (heavy) block and differs only in the capacity of the airplane and the time of arrival of two consecutive flights.

Deplaning Gate (Small) (ALPSLIB, Fig. 11)

This block functions exactly as the Deplaning Gate (heavy) block and differs only in the capacity of the airplane and the time of arrival of two consecutive flights.
Gate Assignment Block (ALPSLIB, Fig. 12)

This block generates the incoming flights and assigns them to the gates as per a predetermined order selected by the user. The flight schedule (arrival time of flights, the capacity of each flight, assigned gate number) is input in the program block (under the columns time, item value, and attribute value respectively).

Holding Room/Gate (Heavy) (ALPSLIB, Fig. 13)

This block represents the holding room/gate waiting area facility. Passengers arrive at random time intervals and have to wait in a queue for getting their boarding passes checked and complete any required checks before boarding the airplane. The passenger arriving first is assumed to be the first to board the airplane hence experiences the least delay. The delay for each passenger can be set through the activity block in the structure window. The delay pattern may be set confirming to the arrival pattern of the passengers into the waiting area. Capacity of each gate is set in the U-Gate block and is usually equal to the capacity of the airplane. Output is the average length of the queue, average waiting time, and LOS in terms the area per passenger.

Holding Room/Gate (Medium) (ALPSLIB, Fig. 14)

This block functions exactly as the holding room/gate (heavy) block and differs only in terms of the capacity of the gate. This block is used to hold passengers traveling in a medium sized aircraft.
Holding Room/Gate (Small) (ALPSLIB, Fig. 15)

This block functions exactly as the holding room/gate (heavy) block and differs only in terms of the capacity of the gate. This block is used to hold passengers traveling in a small aircraft.

Immigration Block (ALPSLIB, Fig. 16)

This block simulates the immigration and naturalization facility. The structure of this block and its function is similar to the ticket counter block. Passengers entering this block have to wait in the queue for their turn at the checking facility. The delay for each passenger varies and is set in the activity block in the structure window. The number of counters to be provided is the user's decision. The output of this block includes the average delay for each passenger and the LOS at the facility in terms of the area per passenger.

NewCombine Block (ALPSLIB)

This block is similar to the combine block of the discrete-event library and differs only in the number of input connectors available. This block simply joins outputs from different blocks in a single stream and passes them to the output. It acts like an entrance door. This block does not alter any properties of the input items and is useful to combine similar or different outputs of other blocks.
Passenger Generator Block (ALPSLIB, Fig. 17)

This block takes the output of the curb delay block that is vehicles and converts them into their occupants i.e., passengers and visitors. The set value block in the structure is used to set the group size. The size of each group is set through the input random number block. Passenger groups are converted to individuals in the 'Q' block.

'Q' (LOS) FIFO Block (ALPSLIB)

This block is identical to the Queue, FIFO block of the discrete-event library except that it outputs maximum length of the queue and also the average waiting time through its connector outputs. This block provides a first-in-first-out (FIFO) queue. The maximum length of the queue can be set in the dialog box. The block outputs average length, average wait time, and utilization (time of simulation over time the block was in use). This block was modified to suit the output requirements for determining the LOS of a facility.

Security Block (ALPSLIB, Fig. 18)

This block simulates the security facility at a landside terminal. Passengers entering this block have to wait in a queue for their turn at the security checking counter. The structure window shows five checking counters the number can be changed as required. These counters are represented by the activity blocks. The delay can be set in the activity blocks and may be constant for a counter or vary as the user defines. The output is average waiting time, average length of the queue and the LOS of the facility in terms of the area per passenger.
Simulation Time (LOS) Block (ALPSLIB)

This block outputs a constant number at every step in the simulation. The user specifies the constant value which should be equal to the duration of the simulation. This block is used for determining the LOS of a facility.

Ticket Counters Block (ALPSLIB, Fig. 19)

This block simulates the activities of a ticketing facility. Passengers entering this block have to use a circulation/waiting area to get to their ticket counter. At the ticket counter they wait in a queue for their turn to check their baggage in and also to check their tickets to get a boarding pass. The number of counters is set by the user in the activity block and the delay is set in the input random number block. The structure window shows two such activity and random number blocks. The first one represents the circulation area and the second one represents the ticketing facility. The output is average delay, average length of the queue, and the LOS of the facility in terms of the area per passenger.

Transfer Passenger Block (ALPSLIB, Fig. 20)

This block generates the transferring passengers from other flights. The number of transfer passengers and their arrival pattern is set in the dialog of the program block in the structure window. The output of this block is passengers transferring from other flights.
Vehicle Parking Block (ALPSLIB, Fig. 21)

This block simulates the parking environment and the airport ground access exit. This block accepts input from the output of the curb delay block (vehicles), and makes a decision (specified by the user) as to send the vehicles out of the airport environment or to send them to the parking lot. The select DE output block is used to make this decision. The activity block can be told to hold a user specified number of parking spaces and the input random number block can be set to delay the vehicles according to a user specified delay distribution. Vehicles opting to exit the airport environment are passed out of the simulation and are accounted for in the exit block. The vehicles in the parking lot go out of the simulation after they are delayed for their parking time and are accounted in the exit block. Performance measures such as the average wait time for a vehicle to get a parking space and the length of the queue of vehicles waiting to get a parking space can be obtained from the dialog box of the activity block.
Figure 6. structure of Circulation Lobby Block
ALPS Library
Figure 7. structure of Curb Delay Block
ALPS Library
Figure 8. structure of Customs Counters Block
ALPS Library
Figure 9. structure of Deplaning Gate (Heavy) Block
ALPS Library
Figure 10. Structure of Deplaning Gate (Medium) Block
ALPS Library
Figure 13. Structure of Holding Room/Gate (Heavy) Block
ALPS Library
Figure 14. Structure of Holding Room/Gate (Medium) Block
ALPS Library
Figure 15. Structure of Holding Room/Gate (Small) Block
ALPS Library
Figure 16. Structure of Immigration Block
ALPS Library
Figure 17. Structure of Passenger Generator Block
ALPS Library
Figure 18. Structure of Security Block
ALPS Library
Figure 19. Structure of Ticket Counters Block
ALPS Library
Figure 20. Structure of Transfer Passenger Block
ALPS Library

Animation block

Generates transfer passengers

Transfer delay block, accounts for delay due to physical transferring of passengers from one point to another.
Figure 21. Structure of Vehicle Parking Block
ALPS Library
5. MODEL RESULTS AND ANALYSIS

Introduction

To demonstrate the versatility and capabilities of the model, the Roanoke Regional airport (Virginia) was simulated. The enplaning and deplaning models were tested for a base year (1994) and horizon year (2010) demand levels.

Base Year Scenario

Roanoke airport has four gates with loading bridge facilities. The traffic mainly consists of commuter airlines with a few short range transports arriving during the busy hour. Apron positions available for commuters presently number five. Five airlines American, Delta, North-West, United, and US Air operate flights from the airport. The typical week day busy hour arrivals are shown in table 2 [Official Aircraft Guide, OAG, Jan. 15, 1994]. The busy hour occurs from 1 PM to 2 PM with a total of eight arrival's, seven of which are commuter flights and one (B-737) is a medium sized aircraft. As shown in table 2 all flights arrive in a span of thirty-five minutes.

Horizon Year Scenario

The demand for horizon year is assumed to be a 100% increase over the base year demand and service facilities are assumed to remain the same as in the base year scenario.

Assumptions

1. A turn around time of 30 minutes is assumed for all commuter aircraft and 35 minutes for all medium aircraft.
Table 2
Base Year Busiest Hour Flight Schedule
for Arriving Flights
(Official Airline Guide, Jan. 15 1994)

<table>
<thead>
<tr>
<th>Arrival Time</th>
<th>Airline</th>
<th>Aircraft Type</th>
<th>Aircraft Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:15 PM</td>
<td>North-West</td>
<td>Sweringen Metro</td>
<td>14</td>
</tr>
<tr>
<td>1:20 PM</td>
<td>US Air</td>
<td>Saab SF3</td>
<td>35</td>
</tr>
<tr>
<td>1:21 PM</td>
<td>United Airline</td>
<td>Jetstream J41</td>
<td>36</td>
</tr>
<tr>
<td>1:29 PM</td>
<td>US Air</td>
<td>Boeing 737</td>
<td>120</td>
</tr>
<tr>
<td>1:35 PM</td>
<td>United Airline</td>
<td>Jetstream J41</td>
<td>36</td>
</tr>
<tr>
<td>1:40 PM</td>
<td>Delta Airline</td>
<td>Embraer EM2</td>
<td>26</td>
</tr>
<tr>
<td>1:50 PM</td>
<td>US Air</td>
<td>De-Havilland 8</td>
<td>46</td>
</tr>
<tr>
<td>1:50 PM</td>
<td>US Air</td>
<td>Sweringen Metro</td>
<td>14</td>
</tr>
</tbody>
</table>
2. The departure schedule for the busy hour is assumed to be same as the arrival schedule after adding appropriate turn around times.

3. A passenger to visitor/greeter ratio of 1:2 is assumed to simulate worst conditions.

4. Transferring passengers account for 50% of the arriving passengers.

5. Length of the concourse is assumed to be 400 ft. with a width of 15 ft.

6. Terminal circulation lobby which houses the ticket counters and other commercial passenger service facilities is assumed to be 300 ft. long and 20 ft. wide.

7. Waiting areas for each gate are assumed to be 75 ft. long and 40 ft. wide.

8. Average delay at a ticket counter is assumed to be 2 minutes with a 1 minute minimum and 3 minute maximum delay.

9. Two security counters process about 10 passengers each per minute.

10. Departing passengers start arriving at the terminal no more than 30 minutes before the departure time.

11. Since all gates are located close to the terminal, baggage claim time for each passenger is assumed to be no more than 5 minutes with a 3 minute minimum delay and an average of four minutes.

Results

The deplaning and enplaning models are shown in the Figures 22 and 23. The results of the Roanoke airport model are shown in figures 24 through 31. Figures 24 through 26 show the LOS (Area/Pass.) of some of the important facilities at roanoke airport for the base year and horizon year demand scenarios. Figures 27 through 31 show the queuing comparisons for the two scenarios. The LOS is measured in terms of the area of the facility (A), total simulation time (T), average waiting time for each passenger (t), and the number of passengers (n) served in the facility during the simulation time period.

\[
\text{LOS (a = area/passenger) sq.units/pass.} = \frac{(T \times A)}{(n \times t)}.
\]
Roanoke Airport
Base Year Scenario
Deplaning Model

Figure 22. Deplaning Model (Roanoke Airport)
Roanoke Airport
Base Year Scenario
Enplaning Model

Figure 23. Enplaning Model (Roanoke Airport)
Figure 24. Base/Horizon Year Scenario
Transient LOS for Baggage Claim Area

Figure 25. Base/Horizon Year Scenario
Transient LOS for Ticket Counters
Figure 26. Base/Horizon Year Scenario
Transient LOS for Security Counters

Figure 27. Base/Horizon Year Scenario
Passenger Queuing in Circulation Area
Figure 28. Base/Horizon Year Scenario
Passenger Queuing for Deplaning Gate

Figure 29. Base/Horizon Year Scenario
Passenger Queuing in Baggage Claim Area
Figure 30. Base/Horizon Year Scenario  
Passenger Queuing for Ticket Counters

Figure 31. Base/Horizon Year Scenario  
Passenger Queuing for Security Counters
6. CONCLUSIONS AND RECOMMENDATIONS

Conclusions

The results obtained for the simple scenario of Roanoke airport give LOS in terms of area/passenger. This is a very easy and convenient way to identify the functioning of the airport under a specific demand condition. The same method could be applied to the horizon year scenario to obtain the standards offered by the airport for an increased demand. This process may not take more than a few minutes and is very ideal for planning purposes where instant results are needed.

This model demonstrates that simulation is an ideal tool for the use in the modeling of a dynamic environment such as the airport landside. Also it demonstrates that the modeling process need not be extremely detailed and difficult to implement. The simulation software EXTEND is seen as a powerful tool for modeling purposes.

The flexibility of changing the basic model to suit any particular airport is a very important aspect of ALPS. For example the basic model had four baggage carousel but Roanoke had only two. The two carousels could simply be disconnected from the rest of the model or totally removed as was the case here. Also the basic model represented an international airport facility whereas Roanoke serves only domestic traffic. This change was easily accomplished simply by removing blocks such as immigration and customs which were not required in this case. Any additional blocks such as the enplaning gates or deplaning gates could be easily added just by picking them from the ALPSLIB.

The time required to input the data for the first run was about 20 minute for each model. The runtime varied depending on the length of the simulation and for a 60 minute run the actual time the model was simulating was 12 minutes. Subsequent data changes such changing the demand levels required less than 5 minutes real time although the run time remained the same. This demonstrates the quick assessment capabilities of the model.
Recommendations

The following recommendations are a result of the experience gained in developing ALPS, by studying and modeling landside operations closely, and the analysis done on the airport operations. The proposed recommendations are:

a) An extensive library of modeling blocks catering to the various requirements of the different airport scenarios may be developed so that the model will become more user friendly.

b) The model blocks at present are hierarchical i.e., they are like sub models. This might be confusing to some of the users. Efforts should be made through extensive programming to make these blocks simple to understand.

c) The model outputs at present are available only in the individual facilities. A single output may be designed to give a summary of the outputs of each facility.

d) ALPS simulates the airport environment in two different models one for enplanement and one for deplanement. The possibility of combining these two models to represent an actual scenario may be researched.

e) Passengers are always flowing in two opposite directions yet, ALPS does not simulate possibility of the mixing of passengers. This possibility should be investigated.

f) The Measures of Effectiveness for this model are the LOS and queue characteristics. The standards for LOS are prescribed by the FAA. These LOS standards may not reflect the actual LOS perceived by the passengers. Therefore better methods for assessing the MOE's should be adopted.
Appendix A

Generic Library Blocks Used In The Model

CONSTANT

This block generates a constant value at each step. The value of the constant can be specified in the dialog (the default constant is 1). If an input is provided, the constant is added to the input. This block is typically used for setting the value for the inputs of other blocks.

EQUATION

This block is used for basic math purposes. The block allows you to enter equations. The user can enter a free-form equation into the dialog and Extend will calculate the results. The equation can be modified as the simulation runs.

HELP

This block shows help text and the user may use this block to document his models. This block may also be used to include information about what impact specific blocks or parameters have on the model, authorship, cross-reference to other models, and so on. There are nine help text boxes provided to write the text. Each box can take up to 255 characters of text.
INPUT RANDOM NUMBER

This block generates random integers or real numbers. The range of the outputs can be set through the two inputs, 1 and 2, or the range can be set in the dialog. The type of distribution available are: uniform, binomial, Poisson, normal, LogNormal, exponential, Erlang, HyperExponential, Weibull, and general. A button is provided in the dialog for plotting a sample of the chosen distribution. The general distribution uses a table of up to 50 values and can be used for a discrete distribution, a stepped distribution, or an interpolated distribution.
Appendix B
Discrete-Event Library Blocks Used In The Model

ACTIVITY, DELAY

This block holds an item for a specified amount of time, then releases it. The delay time is set in the dialog or, if connected, the value at the D connector when the item is received (the connector overrides the dialog). This block can be used for any kind of service delay.

ACTIVITY, MULTIPLE

This block holds many items and passes them out based on the delay and arrival time for each item. The item with the minimum total delay and arrival time is passed out first. The delay time for each item is set through the D connector or, if nothing is connected there, can be specified in the dialog. An example of this block is a supermarket where customers arrive at different times and take a varying amount of time to shop. Customers who arrive first or only shop for a little time will leave first; customers who arrive later or shop a long time will leave last. This block is suitable for a service facility in an airport.

COMBINE

This block combines the items from two different sources into a single stream. The items are not batched together and retain their separate identities.
EXECUTIVE

This block controls the timing and passing of events in a discrete event model. This block is the heart of each discrete event model and must be placed to the left of all the other blocks in the model. Generally, the dialog settings need no modification or no other block needs to be connected to this block. It is very important to place this block on the left edge of the model.

EXIT

This block passes items out of the simulation. The total number of items absorbed by this block is reported in its dialog and at the # connector.

GATE

This block passes a new item into a section of the model only when an item leaves that section. This block is used to restrict the passing of an item into a system that can only have a specific number of items in the system at a time. The first items to arrive are allowed through, and then items are allowed to pass through the block when the sensor connector receives an input indicating that an item has left the section. The sensor connector does not accept items, but only views them.
GENERATOR

This block provides items for a discrete event simulation at specified arrival times. Items can be created with a random distribution or at a constant rate of arrival. The number of items output at each event can be specified in the dialog or at the value (V) connector. The parameters are set in the dialog for the distributions which include: uniform integer, uniform real, binomial, Poisson, normal, log normal, exponential, Erlang, HyperExponential, Weibull, and general also a constant can be output at each event. The general distribution may have up to 20 points and may be interpreted as a discrete, stepped, or interpolated distribution. The input connectors 1 and 2 allow you to modify the parameters of the random distribution as the simulation progresses. This block always pushes items and therefore, should always be followed by a queue or a resource.

GET ATTRIBUTE

This block displays and optionally removes attributes on items. The attribute is shown in the dialog and output at the A connector. Attributes are usually set with the set attribute block. As items are passed through the block, the block reads or removes an attribute, and that attribute can be specified as the first attribute in the list or a named attribute. If an attribute is found its value is reported in the dialog and sent through the A connector. If no attribute is found then the block can be specified to output a no value or a 0. There is also a choice in the dialog to change the value of the item to the value of the attribute. The Δ connector outputs 1 if the attribute value has changed since the last value was read in.
PROGRAM

This block schedules many items. These items with a given priority, start time, item value, attribute name, and attribute value may repeat on a regular basis. This block is useful for repetitive or timed needs. This block always pushes items so it should usually be followed by a queue block or a resource block. Up to 500 events can be generated before repeating. If the start connector is not connected, the times in the dialog are relative to the simulation time. If the start connector is connected, the block will start its program at the time that it views an item or a true value (greater than 0.5) at the connector. In this case, the times in the dialog are relative to the first time the start connector views an item or a true value. The V connector shows the value of the item that is output.

QUEUE, FIFO

This block provides a first-in-first-out (FIFO) queue. The maximum length, which determines how many items the queue can hold, can be set in the dialog. The user can specify that the simulation should stop when the queue is full. The average queue length, average wait time, and utilization of the queue can be viewed in the dialog.
SET VALUE

This block assigns value to the items passing through it. This block is used to change the initial value of 1 to a different value. The V connector allows the values set by the block to be changed & if it is not connected, then the value is as specified in the dialog.

SELECT DE OUTPUT

This block selects outputs based on a decision. The dialog has no options for changing the outputs after a given number of items have passed and selecting based on the select connector. If the select connector is used, the default is that the top output is selected with a 0 and the bottom output with a 1. The user can specify whether invalid selections cause Extend to use the top output or to wait until the connector has a valid input. The user can also specify that each true value (greater than 0.5) or item at the select connector toggles the output.

SET ATTRIBUTE

This block sets the attributes of items passing through it. Up to seven attribute names and values may be assigned to an item with each set attribute block. The attributes may add to or replace existing item attributes. The value of one of the attributes may be specified using the A connector. The value at the A connector overrides the value in the dialog.
Bibliography


Vita

The author was born on April 24, 1968, in Hyderabad, in the state of Andhra Pradesh, India. He received his undergraduate degree in Bachelor of Engineering in Civil Engineering from Osmania University, Hyderabad, India in June 1989. After working for almost two years in Construction industry, he started pursuing graduate studies at Virginia Tech from fall 1991. He completed Master of Science, in Civil Engineering, specializing in Transportation Engineering in February 1994. He plans to pursue a carrier in Airport Engineering.

[Signature]