User Manual
for
Aircraft Repair Model Simulation

Software version 1.0

CS 422 Software Engineering Principles

15 Nov. 2000

Team 9: Goldrush:

Nick Capron

Kyle Klicker

Carina Lansing

Jonathan McPherson

Andy Reynolds

Dave Stone

Russell Swannack

Miguel Vilchez

Kelly Yearout

Manual Version 1.3 (Draft)

This document describes the Aircraft Repair Model Simulation in partial fulfillment of the requirements for the WSU CS442 Software Engineering class. This document is subject to the restrictions listed within.

Abstract

The goal of ARMS is to compare various maintenance scheduling alternatives in order to optimize repair times and downtime costs. Seven types of commercial aircraft will be evaluated using three different scheduling techniques. The model will provide a graphical user interface, so the user can quickly and easily set parameters, run the model, compare scheduling alternatives, and view results.

This document is the User Manual for Aircraft Repair Model Simulation software version 1.0. It is designed to allow users to quickly access information on the various functions of ARMS by the function name. Each function has its own sub-section with information including; the purpose of the function, materials needed, preparations for executing the function, inputs to the function, cautions and warnings regarding the function, invocation of the function, how to suspend or quit the function, output from the function, error conditions associated with the function, and any information that is related to the function. The format of this document complies to the IEEE standard 1063-1987 for software user documentation.

Restrictions

By opening and installing the Aircraft Repair Model Simulation, ARMS, the customer is bound to the following conditions. The use of the ARMS and its documentation is restricted solely to said customer. ARMS may only be installed on a single computer unless additional licenses are granted by team Goldrush. ARMS may not be copied, except for archival purposes as allowed by U.S. copyright laws. The transfer of ARMS is strictly forbidden unless all legal claims to ARMS are forfeited the customer.

Warranties

Team Goldrush makes no warranties, either express or implied, regarding ARMS, the ARMS user manual, and their fitness for any specific purpose. Team Goldrush provides ARMS and user manual as-is. Team Goldrush is not responsible for any adverse effect ARMS may have on a computer's hardware/software, computer peripherals, data storage devices, property, and persons resulting from use of ARMS and its user manual. Some states disallow the exclusion of implied warranties. Certain exclusions may not apply to you. Specific legal rights are provided by this warranty. There may be additional rights you have which vary from state to state.

Table of Contents

1 Introduction ***6***
1.1 Audience 6
1.2 Applicability 6
1.3 Purpose 6
1.4 User Manual Usage 6
1.5 Related Documents 7
1.6 Conventions And Terms 7
1.7 If You Have A Problem 7

2 Installing ARMS ***7***
2.1 Before You Begin 8
2.2 Unpacking ARMS 8
2.3 Compiling ARMS 8
2.4 Running ARMS 8

3 Using ARMS ***8***
3.1 Add Event 8
3.1.1 Purpose *8*
3.1.2 Materials *9*
3.1.3 Preparations *9*
3.1.4 Inputs *9*
3.1.5 Cautions and Warnings *9*
3.1.6 Invocation *9*
3.1.7 Suspension of Operations *9*
3.1.8 Termination of Operation *9*
3.1.9 Output *9*
3.1.10 Error Conditions *10*
3.1.11 Related Information *10*
3.2 Delete Item 10
3.2.1 Purpose *10*
3.2.2 Materials *10*
3.2.3 Preparations *10*
3.2.4 Inputs *10*
3.2.5 Cautions And Warnings *10*
3.2.6 Invocation *11*
3.2.7 Suspension of Operations *11*
3.2.8 Termination of Operations *11*
3.2.9 Outputs *11*
3.2.10 Error Conditions *11*
3.3 Pause 11
3.3.1 Purpose *12*
3.3.2 Materials *12*
3.3.3 Preparations *12*
3.3.4 Inputs *12*
3.3.5 Cautions And Warnings *12*
3.3.6 Invocation *12*
3.3.7 Suspension of Operations *12*
3.3.8 Termination of Operations *12*
3.3.9 Outputs *12*
3.3.10 Error Conditions *12*
3.3.11 Related Information *12*
3.4 Create New Queue Discipline 13

3.5 Restart Application 13
3.5.1 Purpose *13*
3.5.2 Materials *13*
3.5.3 Preparations *13*
3.5.4 Inputs *13*
3.5.5 Cautions And Warnings *13*
3.5.6 Invocation *13*
3.5.7 Suspension of Operations *13*
3.5.8 Termination of Operations *13*
3.5.9 Outputs *14*
3.5.10 Error Conditions *14*
3.6 Resume 14
3.6.1 Purpose *14*
3.6.2 Materials *14*
3.6.3 Preparations *14*
3.6.4 Inputs *14*
3.6.5 Cautions And Warnings *14*
3.6.6 Invocation *14*
3.6.7 Suspension of Operations *14*
3.6.8 Termination of Operations *14*
3.6.9 Outputs *15*
3.6.10 Error Conditions *15*
3.7 Simulation Rate 15
3.7.1 Purpose *15*
3.7.2 Materials *15*
3.7.3 Preparations *15*
3.7.4 Inputs *15*
3.7.5 Cautions And Warnings *15*
3.7.6 Invocation *15*
3.7.7 Suspension of Operations *15*
3.7.8 Termination of Operations *15*
3.7.9 Outputs *16*
3.7.1 Error Conditions *16*
3.8 Start 16
3.8.1 Purpose *16*
3.8.2 Materials *16*
3.8.3 Preparations *16*
3.8.4 Inputs *16*
3.8.5 Cautions And Warnings *16*
3.8.6 Invocation *16*
3.8.7 Suspension of Operations *17*
3.8.8 Termination of Operations *17*
3.8.9 Outputs *17*
3.8.10 Error Conditions *17*

4 Error Messages, Known Problems, Error Recovery 17
4.1 Input Screen Error Messages and Error Recovery 17
4.2 Progress Screen Error Messages and Error Recovery 17
4.3 Report Screen Error Messages and Error Recovery 17
4.4 Known Problems 17

5 References 18

6 Glossary 18

Table of Figures

Figure A-1. Input Screen. 20
Figure A-2. Progress Screen. 20
Figure A-3. Report Screen. 21
Figure A-4. Insert Event. 21

List of Tables

Table B-1. The Seven Aircraft Types. 23

1 Introduction

Aircraft Repair Model Simulation, ARMS, is a tool to help users design aircraft maintenance facilities. This manual will help them become familiar with ARMS and its many features. The following section lists the requirements for running ARMS and explains the format of the document so as to allow easiest reference for all types of users.

1.1 Audience

This manual will help users become familiar with the many features of ARMS. Users should be somewhat familiar with the using the Linux operating system to execute programs. Users should also have an understanding of the aircraft repair process. This will allow users to take full advantage of ARMS.

1.2 Applicability

This manual, version 1.3, covers ARMS version 1.0 only. Previous and future versions of ARMS may have some similarity in functionality to ARMS v1.0, but to truly learn ARMS please get an ARMS manual for your version of ARMS. The ARMS product is intended operate on a PC using the Linux operating system, (kernel 2.2 or newer), with 10 megabytes of hard drive space and 32 megabytes of random access memory. Java 1.3 is also required for ARMS to function correctly.

1.3 Purpose

The Aircraft Repair Model Simulation, (ARMS), is a powerful tool designed to aid in the understanding of the effects of different arrangements of aircraft repair facilities. It allows users to simulate these repair facilities and see the effects on downtime cost, and effectiveness of the repair facility arrangement. ARMS is easy to use, so users will be able to sit down and get ARMS running and try it out right away. ARMS is functionally powerful and the ARMS user manual will give a detailed explanation of the many features it contains.

1.4 User Manual Usage

Sections two and three will first, help with the ARMS installation process, and second give a detailed explanation of the functionality of ARMS. Section two gives a brief overview on how to install ARMS. A detailed description of every feature of ARMS is in section three. For quick

reference on any function, section three lists functions alphabetically by name. Section four has information concerning error conditions. The glossary defines terms you may be unfamiliar with. It is located at the end of this document.

1.5 Related Documents

If you would like some background information related to ARMS read srs_gold_V3.6, and dnb_gold_V1.5 found at www.tricity.wsu.edu/~goldrush. These documents will give insight to the production and history of ARMS.

1.6 Conventions And Terms

The ARMS user manual follows IEEE standard 1063-1987 for user manuals. It uses the reference mode style to explain ARMS functionality.

ARMS uses three different GUIs, an input screen, a progress screen, and a report screen. Users can access different functions from each of these screens. This document will refer to these screens by the aforementioned names. Screen shots are located in appendix A; figures A-1, A-2, and A-3.

1.7 If You Have A Problem

If you should have any problem related to ARMS contact team Goldrush at goldrush@tricity.wsu.edu, or visit our website, www.tricity.wsu.edu/~goldrush for online help. If you have any suggestions on how ARMS could be improved please email us at goldrush@tricity.wsu.edu and include “KyleFixIt” in the subject field. For errors and improvements to the ARMS user manual please email goldrush@tricity.wsu.edu and include “AndyTypeIt” in the subject line.

2 Installing ARMS

This Section will help in getting started with ARMS, starting with how to get ARMS installed and running ARMS. Following sub-sections will introduce you to all the functions of ARMS. Reading this manual will give you an in-depth overview of ARMS, but for the impatient each function has its own section for quick reference.

2.1 Before You Begin

Before you install ARMS certain requirements will need to be met:

- A computer running Linux with kernel version 2.2 and higher
- A copy of ARMS, (ARMS.tar.gz)
- Java 1.3 installed on your computer.

2.2 Unpacking ARMS

Before you can install ARMS it must be unpacked. Open a shell window and change to the directory where the ARMS.tar.gz resides. Type the command “tar xzvf ARMS.tar.gz” and press enter. This will create a new directory named ARMS. ARMS is now ready to be compiled.

2.3 Compiling ARMS

To compile ARMS, first change to the ARMS directory. Type the command “make”. ARMS will now compile. If you have any problems please email goldrush@tricity.wsu.edu. When ARMS is finished compiling it is ready to use.

2.4 Running ARMS

To run ARMS change to the ARMS directory and type “java ARMS”. ARMS will start and you can begin using it.

3 Using ARMS

This section covers the different functions a user may interact with when using ARMS. Functions are listed alphabetically by the functions name. Since there are three different screens the screen from which the function is accessed will be mentioned.

3.1 Add Event

The “Add Event” function allows users to enter a new event into the event list. It can be accessed from the progress screen.

3.1.1 Purpose

The “Add Event” function adds an event to the event list. This allows users to customize ARMS to simulate certain conditions that might arise in an aircraft repair facility.

3.1.2 Materials

Appendix B lists the seven aircraft types. Figure A-4 in appendix A shows the Insert Event dialog box.

3.1.3 Preparations

Before an event can be added the “Pause” button must be pressed.

3.1.4 Inputs

When adding an event, there are three pieces of data that must be entered, the event type, the aircraft type, and the day that the event occurs. There are three different event types: arrival, departure, and end simulation. There are seven different aircraft types, which are represented by the numbers 1-7. The day on which an event occurs is a number greater than 0.00 and less than the planned duration of the simulation.

3.1.5 Cautions and Warnings

Each field must be completed for the “Add Event” function to complete.

3.1.6 Invocation

The “Add Event” function is accessed from the progress screen. There is a button labeled “Add Event”. Simply press this button to invoke the function. This causes a dialog box to appear on the screen. The dialog box has two drop down menus and a text box for entering the inputs. Select the desired input and then click “Ok”.

3.1.7 Suspension of Operations

The “Add Item” function cannot be suspended.

3.1.8 Termination of Operation

The insert event dialog box has a button labeled ”Cancel”. Clicking this button will close the box without adding an event.

3.1.9 Output

After the dialog box closes the new event will appear in the “Event List” window.

3.1.10 Error Conditions

If the number entered for the time is greater than 364.99 it will not be included in the statistics generated by ARMS.

3.1.11 Related Information

If a user wants the simulation to end before the planned duration is up, add an “endsim” event at the current time.

The information shown in the “Event List” is formatted as follows:

- Items of information are separated by colons
- The first field is the type of event ‘A’ for departure and ‘D’ for departure
- The second field is the time at which the event will occur
- The third field is a unique identification number assigned to each aircraft by ARMS
- The last field is the aircraft type.

3.2 Delete Item

This function is used to delete events from the “Event List”, “Queue List”, and from the “Service Stations”. It is accessed from the Progress screen.

3.2.1 Purpose

The “Delete Event” function adds an event to the event list. This allows users to customize ARMS to simulate certain conditions that might arise in an aircraft repair facility.

3.2.2 Materials

There are no separate materials needed for this function.

3.2.3 Preparations

The user must first press the “Pause” button. Then an event from, the “Event List”, the “Queue List”, or the “Service Stations” should be selected.

3.2.4 Inputs

There is no input required for this function.

3.2.5 Cautions And Warnings

When deleting events the user should note the following:

- If the user deletes the “endsim” event the simulation will not terminate.

- If a departure event is deleted, the aircraft remains where it is until a new departure is scheduled
- Deletes cannot be undone.

3.2.6 Invocation

The “Delete Item” function is invoked by first highlighting an event by clicking on it and then clicking the “delete Item” button above the corresponding list.

3.2.7 Suspension of Operations

This function cannot be suspended.

3.2.8 Termination of Operations

Once an event is highlighted and the “Delete Item” button is clicked, the delete function executes and cannot be terminated.

3.2.9 Outputs

The highlighted event will be removed from its list when the ”Delete Item” button is pressed.

3.2.10 Error Conditions

There are no error conditions associated with this event.

3.2.11 Related Information

The information shown in the “Event List” and “Queue List” is formatted as follows:

- Items of information are separated by colons
- The first field is the type of event ‘A’ for departure and ‘D’ for departure
- The second field is the time at which the event will occur
- The third field is a unique identification number assigned to each aircraft by ARMS
- The last field is the aircraft type.

The “Service Stations” list has a slightly different format:

- The first field is a number corresponding to a repair bay
- The second field shows the type of aircraft accepted to the bay, either wide-body or regular
- The remaining fields are the same as the “Event List” and the “Queue List”.

3.3 Pause

The “pause” function temporarily suspends ARMS. It is accessed from the progress screen.

3.3.1 Purpose

The “Pause” function allows users to temporarily stop the simulation. This allows users to scan through the “Event list”, “Queue List”, and “Service Stations” to get an idea of what’s happening with the simulation, and also add and delete events if they so chose.

3.3.2 Materials

There are no separate materials needed for this function.

3.3.3 Preparations

There are no preparations to make before invoking the “Pause” function.

3.3.4 Inputs

There is no input required for this function.

3.3.5 Cautions And Warnings

The “Pause” function does not do anything that could invalidate the simulation.

3.3.6 Invocation

The “Pause” function is invoked by clicking the “Pause” button that appears on the progress screen.

3.3.7 Suspension of Operations

This function is designed to suspend the operation of the simulation.

3.3.8 Termination of Operations

The “Resume” function is the inverse of the “Pause” function.

3.3.9 Outputs

There is no output from the “Pause” function.

3.3.10 Error Conditions

There are no error conditions associated with this event.

3.3.11 Related Information

To add or delete event the user must first press the “Pause” button. ARMS starts in paused mode.

3.4 Create New Queue Discipline

The “Create New Queue Discipline” button will allow users to add new aircraft repair facilities setups. It is accessed from the input screen. It is not implemented in this version of ARMS.

3.5 Restart Application

Once the simulation ends users are allowed to start from the beginning.

3.5.1 Purpose

After the simulation has run the results are shown on the report screen. Users can then use the “Restart Application” button to start ARMS over.

3.5.2 Materials

There are no separate materials needed for this function.

3.5.3 Preparations

There are no preparations to make before invoking the “Restart Application” function.

3.5.4 Inputs

There is no input required for this function.

3.5.5 Cautions And Warnings

ARMS does not save the results shown on the report screen. Once the “Restart Application” button is pressed the report screen will close. The user is responsible for any data they wished to be saved.

3.5.6 Invocation

The “Restart Application” function is invoked by clicking the “Restart Application” button that appears on the report screen.

3.5.7 Suspension of Operations

This function cannot be suspended.

3.5.8 Termination of Operations

This function cannot be terminated.

3.5.9 Outputs

When the “Restart Application” button is pressed the report screen will close and the input screen will open.

3.5.10 Error Conditions

There are no error conditions associated with this event.

3.6 Resume

ARMS requires the simulation to pause in order for adds and deletes to be made. A function is needed to un-pause the simulation.

3.6.1 Purpose

The ”Resume” button lets users un-pause the simulation.

3.6.2 Materials

There are no separate materials needed for this function.

3.6.3 Preparations

There are no preparations to make before invoking the “Resume” function.

3.6.4 Inputs

There is no input required for this function.

3.6.5 Cautions And Warnings

The “Resume” function does nothing that could invalidate the results of the simulation.

3.6.6 Invocation

The “Resume” function is invoked by clicking the “Resume” button that appears on the progress screen.

3.6.7 Suspension of Operations

This function cannot be suspended.

3.6.8 Termination of Operations

The “Pause” is the inverse of the “Resume” function.

3.6.9 Outputs

When the “Resume” button is pressed the text on the “Add Item”, and all the “Delete Item” buttons will be grayed out.

3.6.10 Error Conditions

There are no error conditions associated with this event.

3.7 Simulation Rate

It is sometime desirable to increase or decrease the rate the simulation executes.

3.7.1 Purpose

The “Simulation Rate” slider allows users to speed up or slow down the rate at which ARMS is execution. This allows users to get to the end fast or watch every little detail.

3.7.2 Materials

There are no separate materials needed for this function.

3.7.3 Preparations

There are no preparations to make before invoking the “Simulation Rate” function.

3.7.4 Inputs

There is no input required for this function.

3.7.5 Cautions And Warnings

The “Simulation Rate” function does nothing that could invalidate the results of the simulation.

3.7.6 Invocation

The “Simulation Rate” function is invoked by clicking the slider button labeled ”Simulation Rate” that appears on the progress screen. Moving the slider to the left will decrease the rate and movement of the slider to the right will increase the rate of the simulation

3.7.7 Suspension of Operations

This function cannot be suspended.

3.7.8 Termination of Operations

The “Pause” is the inverse of the “Resume” function.

3.7.9 Outputs

Events will appear to take place slower or faster in the “Event List, “Queue List”, and the “Service Stations”.

3.7.10 Error Conditions

There are no error conditions associated with this event.

3.8 Start

ARMS needs some information before it can start. The “Start” function is accessed from the input screen.

3.8.1 Purpose

The “Start” function allows the user to start ARMS running once the information ARMS needs has been entered.

3.8.2 Materials

If you are unfamiliar with the queuing disciplines see the glossary for more information.

3.8.3 Preparations

ARMS requires several items to be set before the “Start” button can be pressed.

3.8.4 Inputs

There are three different items to select before invoking the “Start” function:

- Set the queue Discipline using the drop-down menu
- Enter the number of wide-body and regular service stations
- Enter a random seed, (ARMS provides a random seed, but it can be overridden).

3.8.5 Cautions And Warnings

The user should make sure that:

- The total number of service stations does not exceed 10
- The random seed is between 1 and 99999

3.8.6 Invocation

The “Start” function is invoked by pressing the button labeled ”Start” that appears on the input screen.

3.8.7 Suspension of Operations

This function cannot be suspended.

3.8.8 Termination of Operations

The “Start” function cannot be terminated.

3.8.9 Outputs

Once the “Start” button is pressed the input screen will close and the progress screen will open.

3.8.10 Error Conditions

An error will occur if the user enters more than 10 total stations or if an invalid seed is inserted.

4 Error Messages, Known Problems, Error Recovery

This section will cover errors and problems with ARMS. Each of the three screens will be separated into sub-sections.

4.1 Input Screen Error Messages and Error Recovery

The input screen will produce an error message under the following circumstances:

- More than 10 total service stations has been entered
- An invalid random seed has been entered.

These errors can be recovered from by entering no more than 10 service stations, and a random seed between 1 and 99999.

4.2 Progress Screen Error Messages and Error Recovery

There are no error messages associated with the progress screen.

4.3 Report Screen Error Messages and Error Recovery

There are no error messages associated with the report screen.

4.4 Known Problems

There are no known problems with ARMS at this time. If one is discovered please email problem to goldrush@tricity.wsu.edu and type “kylefixit” in the subject line.

5 References

1. Sheldon, F.T., "Project Requirements, CS422 Software Engineering Principles" http://www.eecs.wsu.edu/~sheldon/cs422.html. Fall 2000.
2. IEEE-SA Standards Board, IEEE Standard for Software User Documentation. IEEE Std 1063-1987. USA, December 10, 1987.
3. Team Goldrush, "Design Notebook for Aircraft Repair Modeling Simulation", version 1.5 November 8, 2000.
4. Team Goldrush, "Software Requirement Specifications for Aircraft Repair Modeling Simulation", version 3.6, October 16, 2000.
5. Team Goldrush, "Critical Design Review for Aircraft Repair Modeling Simulation." October 25, 2000.
6. Sheldon, F.T., "Cpts 422 Project Requirements, Project 4 "Jet Aircraft", Table 1" http://www.eecs.wsu.edu/~sheldon/cs422/prjreq.html. Fall 2000.

6 Glossary

This glossary defines acronyms used in the ARM User Manual. It also lists the seven aircraft types and explains the different queuing disciplines.

ARMS – Aircraft Repair Model Simulation

GUI – Graphical User Interface

MB – megabyte, as in 1 million bytes of data storage

PC – Personal Computer

RAM – Random Access Memory

Dialog Box – A small window that appears on the screen usually invoked by pressing a button, allowing a user to enter information required for the program to continue.

APPENDIX A: Screen Shots

This appendix contains screen shots of various screens encountered while using ARMS.

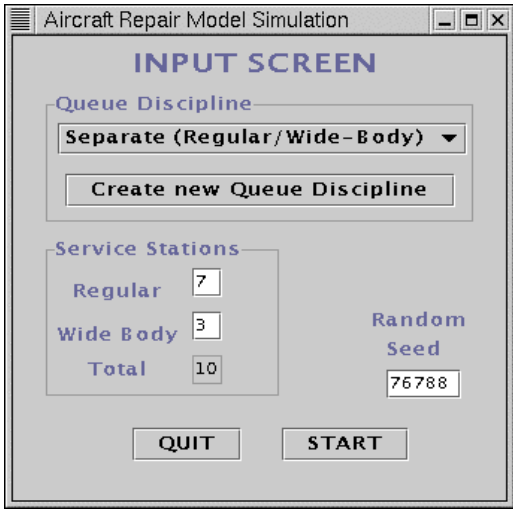

Figure A-1. Input Screen.

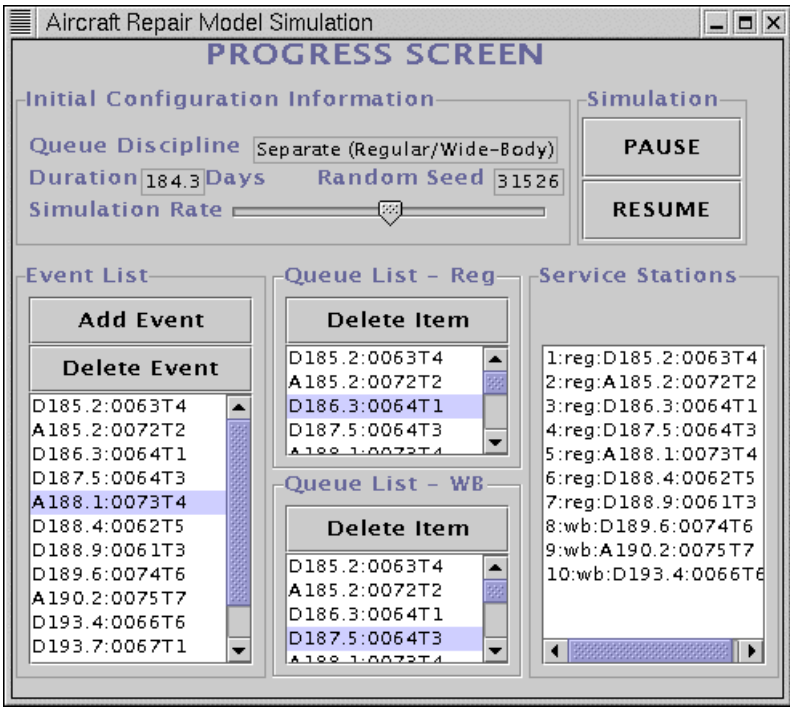

Figure A-2. Progress Screen.

Figure A-3. Report Screen.

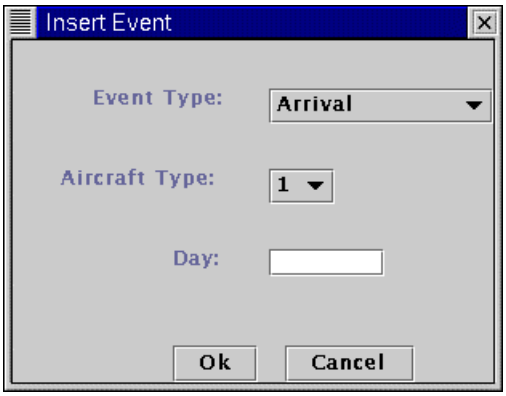

Figure A-4. Insert Event.

APPENDIX B: Aircraft Types

This appendix is a table of the seven aircraft type, a * denotes a wide-body aircraft.

Table B-1. The Seven Aircraft Types.

plane type *(i)*	number of engines	*a(i)*	*A(i)*	*B(i)*	*P(i)*	*r(i)*	*c(i)*
1	4	8.1	0.7	2.1	0.30	2.1	2.1
2	3	2.9	0.9	1.8	0.26	1.8	1.7
3	2	3.6	0.8	1.6	0.18	1.6	1.0
4*	4	8.4	1.9	2.8	0.12	3.1	3.9
5	4	10.9	0.7	2.2	0.36	2.2	1.4
6	2	6.7	0.9	1.7	0.14	1.7	1.1
7*	3	3.0	1.6	2.0	0.21	2.8	3.7

Queuing Discipline 1- All waiting aircraft form a single, first-in/first-out (FIFO) queue. All stations (the number specified by the user) service all aircraft.

Queuing Discipline 2 – Wide-body and regular body aircraft wait in separate FIFO queues. The wide-body queue is given non-preemptive priority over the regular body queue. All stations (the number specified by the user) service all aircraft.

Queuing Discipline 3 – Wide-body and regular body aircraft wait in separate FIFO queues.