

ETMS/ATMS Software Requirements

Name: R. Sharick

Date: ~~05/21/06~~04

Feature Described: GAAP Functionality

Document Version: ~~1.10-5~~

Remarks:

Revision History			
Version	Date	Authors Initials	Description of Change
1.0	4/16/04	RS	Initial version for FAA review.
1.1	5/21/04	RS	<u>Changes to FADT file format.</u> <u>Remove GAAP parameter message types.</u> <u>Add changes to START_PARAMETERS block.</u>

1. Background

When the arrival demand at an airport exceeds its capacity, the FAA may impose a Ground Delay Program (GDP). Flights scheduled to arrive during the program are assigned arrival slots. Flights that are not known about at the time the program is issued receive a FA delay that varies depending on the arrival time. These flights are known as pop-ups. Some airports do not fit this model. The known demand may not exceed the capacity but it is believed that the unknown demand will. This can occur at airports that serve large numbers of General Aviation (GA) flights. The General Aviation Airport Program (GAAP) was designed for this situation. As usual, scheduled flights will receive a slot, but unassigned slots will also be generated for the pop-up flights. As pop-up flights become known, they will be assigned to one of the open slots or they will receive an FA delay. The FA delay time will also differ in that it will be one set value for all flights.

This document describes the software requirements to implement the GAAP feature. Requirements below are numbered and in boldface. Unbolded text is included to clarify but is not, strictly speaking, part of the requirement.

2. Functional Decomposition

Overview

The processing of GAAP controls will introduce new functionality into several ETMS programs. No new processes or data flows will be introduced but many will be modified.

FSM is used to specify the GDP and it creates a file that is sent to Autosend. Autosend breaks this up into multiple files and forwards them to router and then on to IFCN and finally to EDCT. EDCT uses the slot list file to create entries in its Control Data Base (CDB) for the controlled flights. It uses the FA delay table file to assign delays to the pop-up flights. The FA delay table is in turn forwarded to FDB so that it knows which airports are controlled and so it can detect the pop-up flights.

FSM also sends the GDP parameters to ADL_DD via ADL_FE. ADL_DD creates a control airport entry but will not write the parameters into the ADL files until it receives a message from EDCT on the same program.

As flights enter ETMS, the FDB checks whether they will arrive at a controlled airport within the time frame of the program. If so, it sends a message to EDCT to inform it of the pop-up flight. EDCT compares the arrival time with the FA delay table and assigns the appropriate delay to the flight. A message is sent back to FDB with the control times but these flights do not have a slot. Once the flight has control information in the FDB, it will appear in subsequent ADLs with control data.

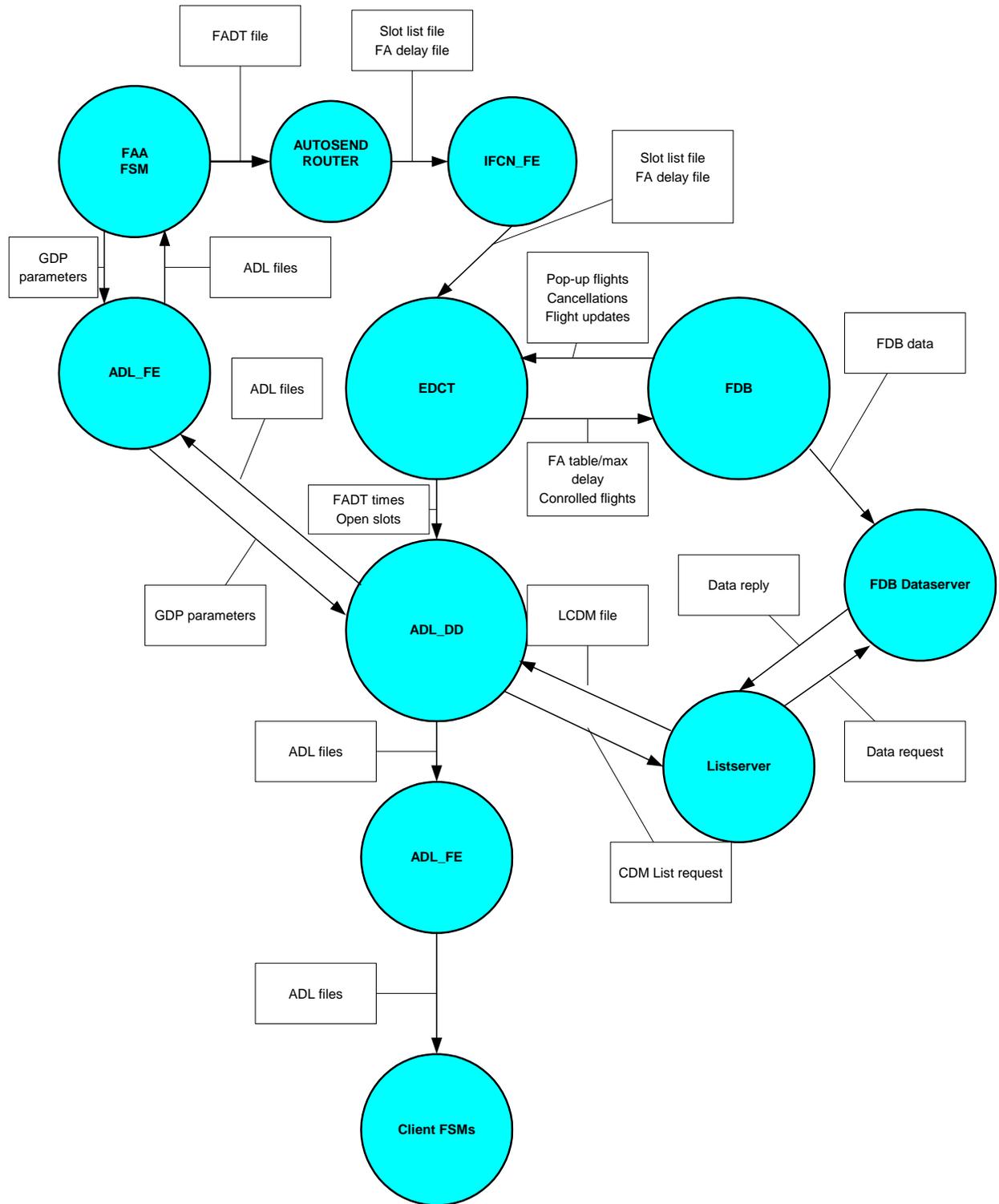
The GAAP program FA delay table file will contain a list of unassigned slots. EDCT will store these and assign them to pop-up flights. A pop-up flight will receive a slot whose control time is as close to the flight's arrival time as possible. If the difference between the arrival time and the control time is greater than the maximum delay, then the flight will not receive a slot but will be given an FA delay equal to the maximum delay. No flight at a GAAP airport will be delayed longer than the maximum delay.

A GAAP program will differ in that no FA delay table will exist. Rather one value, the maximum delay, will apply to all pop-up flights that are not assigned an open slot. FDB will still need to know about the time frame of the program so that it can detect pop-ups. The FA delay file will contain exemption and exclusion data as it does now but will include a maximum delay and no FA delay table.

FSM must be able to display the open slots for a GAAP controlled airport. In order for the list of open slots to appear in an ADL file, EDCT will need to send the initial list to ADL_DD and EDCT will need to update ADL_DD as slots are used and if slots become open again.

Flights that receive an open slot will be treated just like any other controlled flight with a slot; in particular, they can be substituted.

Following is a diagram of the processes and data flows involved in a GAAP program.



2.1. EDCT

EDCT will need to store the unassigned slots that FSM has created for a GAAP controlled airport. It is critical that the list of slots be preserved even if the program restarts; they must be written to disk and not maintained in memory only. The current design of EDCT has both master and slave programs processing controls in parallel. It is also critical that they receive the same data so that they achieve the same results. If a master/slave switch occurs, they must have the same information.

2.1.1. EDCT shall store the data related to the GAAP program:

2.1.1.1 ~~The type of program~~ **method of handling pop-up flights, GAAP.**

2.1.1.2 **The list of slots.**

2.1.1.3 **The maximum delay time.**

2.1.2. **The data shall be stored in such a way that they are preserved if EDCT restarts.**

2.1.3. **The same data shall be stored whether the string the program is running on is master or slave.**

The next three requirements describe what EDCT must do when it first receives data for a GAAP controlled airport.

2.1.4. **EDCT shall accept a list of unassigned slots and the maximum delay from FSM.** The list of unassigned slots and the maximum delay will be contained in the FADT file received from FSM, via autosend and IFCN. See ICD for the format of new sections.

2.1.4.1 **The control type for the file shall be GAAP.**

2.1.5. **EDCT shall add the list of unassigned slots to its database.**

2.1.6. **EDCT shall send the list of unassigned slots to ADL_DD.** The message type will be NWA_CDM_TO_DD_ADD_SLOT and the contents will be a space delimited, null-terminated string with the airport first followed by the slot names.

2.1.7. **EDCT shall send the FA delay data to the FDB.** The data sent to the FDB are the airport name and the start and end times of the program. This will be the same as for a regular GDP. FDB still needs this information to determine whether a flight is a pop-up or not.

There are two scenarios for detecting pop-up flights. The first occurs only at the beginning of a program and the second occurs continuously throughout a program.

In the first scenario, FSM will assign slots to all the flights it knows about based on the data in the ADL. ADLs are generated every five minutes. A flight entry can be created between the time that the ADL was generated and the time the program data is in place. For a regular GDP these flights are detected by comparing a list of flights arriving at the airport when this FA table is received to the flights already assigned slots. In general, flights on the FA table list are treated as pop-ups and assigned an FA delay. For a GAAP controlled airport they are now eligible for a slot.

2.1.8. **EDCT shall accept a list of flights that are scheduled to arrive at the controlled airport and if these are not already controlled shall treat them as pop-ups eligible for an open slot.** This list comes to EDCT via the TDB and FDB data server.

Once a GDP of any kind is in place, FDB can detect pop-ups that will arrive at the airport. Pop-ups occur when a flight plan is filed or a flight is created that will arrive within the start and end times of

the program. FDB sends a transaction to EDCT to inform it of such flights. For regular pop-ups, EDCT will assign a delay based on the FA delay table. For a GAAP control, EDCT must first try to assign a slot and if that fails, it will assign the max delay.

2.1.9. EDCT shall accept messages from FDB when it detects pop-up flights. The message type is MSG_PROPOSED_FLIGHT.

2.1.10. For any pop-up flight, EDCT shall first determine whether the flight is eligible for a slot. A pop-up flight is eligible for a slot if its arrival time is between the start and end time of the program and it is not exempt or excluded according to the criteria defined for the program.

2.1.11. For an eligible flight EDCT shall search through the available unassigned slots and attempt to assign that flight to the first slot that is at or after its requested arrival time up to the maximum delay time.

2.1.12. If a flight is assigned a slot, EDCT shall perform the following actions.

2.1.12.1 EDCT shall create an entry in its database for the flight with the following data:

2.1.12.1.1 The control type - GAAP.

2.1.12.1.2 The assigned slot name.

2.1.12.1.3 The control time of arrival for the assigned slot.

2.1.12.1.4 The control time of departure which shall be the control time of arrival minus the ETE of the flight.

2.1.12.2 EDCT shall remove the assigned slot from the list of open slots for the airport.

2.1.12.3 EDCT shall send a message to ADL_DD to remove the assigned slot from its list of unassigned slots. The message type will be NET_CDM_TO_DD_REMOVE_SLOT and the contents will be a space delimited, null-terminated string with the airport first followed by the slot name.

2.1.12.4 EDCT shall send a message to FDB to indicate that the flight now has a slot. The message type will be MSG_EDCT with a control type set to 'gaap'.

2.1.13. If EDCT finds no unassigned slot between the requested arrival time and the maximum delay time, it shall assign an FA delay equal to the maximum delay time.

Except for the assignment of the maximum delay, these pop-up flights will be treated the same as flights that receive an FA delay in a regular GDP.

A flight that receives a GAAP slot will be treated just like any other controlled flight. In particular, the operator of the flight will be able to substitute it with other flights.

2.1.14. EDCT shall allow substitutions for flights with a control type of GAAP. This includes simple substitutions and slot credit substitutions.

2.1.15. When EDCT receives a message from FDB for a pop-up flight and that flight is arriving at a non-GAAP controlled airport, it shall assign an FA delay as it currently does.

When a controlled flight that is operated by a CDM participant is cancelled, the operator can choose to retain the slot, possibly so that it can substitute another flight into it. Since many of the flights arriving at a GAAP controlled airport will not be CDM participants, there will be no way for them to retain and use the slots of cancelled flights. It would be better to return these slots to the pool of

open slots, rather than wait for a program compression. The next requirements describe how these flights will be identified and how their slots will be handled.

2.1.16. EDCT shall identify an unaffiliated flight as one that one that does not have a three-letter carrier name and that does not exist in any of the N-number flight id files. The N-number flight id files are associated with the airline definitions file. The list all of the GA flights that will be treated as CDM members. So, if a flight has a three-letter code or exists in the hash table, then it is affiliated.

2.1.17. When EDCT receives a cancellation of an unaffiliated flight, it shall return the slot to the pool of unassigned slots.

2.1.17.1 EDCT shall remove the flight from its control database.

2.1.17.2 EDCT shall add the slot back into the list of unassigned slots.

2.1.17.3 EDCT shall send a message to ADL_DD to add the slot to its list of unassigned slots. The message type shall be NWA_CDM_TO_DD_ADD_SLOT. The contents of the message shall be the name of the airport and the name of the slot.

2.1.17.4 EDCT shall send a message to FDB to decontrol the flight. The message type shall be MSG_CONTROL_CANCEL. The contents of the message shall be the standard edct to fdb transaction packet.

2.1.18. If an affiliated flight with a GAAP control type is cancelled, then EDCT shall treat the flight as it now treats CDM participant flights that are cancelled.

While substitutions are turned off, the current assignments of slots to flights must remain frozen. This is how a regular GDP works. For a GAAP program, additionally, it means that any pop-ups that appear while subs are off should not be assigned to a slot because the list of open slots may change. EDCT will need to keep a list of such flights and when subs are turned back on, EDCT will have to go through the list and assign slots or maximum delays to the flights.

2.1.19. If substitution processing is OFF at a GAAP controlled airport, then EDCT shall not assign a slot to a pop-up flight.

2.1.19.1 EDCT shall maintain a queue of such flights.

2.1.20. When substitution processing is turned back ON at the GAAP controlled airport, then EDCT shall assign the queued pop-up flights to slots in the order that they were queued or assign the maximum delay if no slot can be assigned.

The list of slots at a GAAP controlled airport will exist only while the program is active. When the program is purged, the list should be disposed of. When the program is revised or compressed, but remains a GAAP program, then a new list will be sent. This new list should replace the old list.

2.1.21. When EDCT receives a PURGE request for a GAAP controlled airport, it shall remove all unassigned slots for that airport.

2.1.22. When EDCT receives a revision or a compression of a GAAP program, it shall remove all unassigned slots for that airport and replace them with the list of unassigned slots contained in the new FADT file, if any.

2.1.22.1 EDCT shall send the new list of slots to ADL_DD. The message type will be NWA_CDM_TO_DD_ADD_SLOTS and the contents will be a space delimited, null-terminated string with the airport first followed by the slot names.

If the arrival demand at a GAAP controlled airport starts to exceed the capacity, the FAA might change a GAAP program to a regular GDP.

2.1.23. EDCT shall be capable of switching from a GAAP program to a non-GAAP program at an airport. The value of the POPUP_MODE parameter determines how the pop-up will be handled.

2.1.23.1 If a GAAP program is replaced by a non-GAAP program, then EDCT shall assign FA delays based on the FA delay table.

2.1.23.2 If a non-GAAP program is replaced by a GAAP program, then EDCT shall assign FA delays based on the maximum delay associated with the GAAP program.

2.1.24. EDCT shall provide a report that lists the unassigned slots at a specified airport.

2.1.24.1 The name of the report shall be EDCT OPENSLOTS.

2.1.24.2 The request shall be accepted from TSD, FSM, and through Netmail.

2.1.25. EDCT shall be able to provide all existing reports for a GAAP program.

2.2. FDB

The first requirement simply defines the new control type and states where it must appear. Basically, it must appear in the three main user outputs, the ADL files, the FTM, and list reports. It also needs to appear in the CDM transaction files that are used as input to other programs.

2.2.1. FDB shall define a new control type for pop-up flights that receive slots.

2.2.1.1 The name of the type shall be GAAP.

2.2.1.2 FDB shall supply this control type when requested for ADL files.

2.2.1.3 FDB shall write this control type into the CDM transaction files when the flight is initially assigned the slot.

2.2.1.4 FDB shall send this control type to FTM.

When a pop-up flight is assigned a slot, EDCT will send a message to FDB. This message will be the normal control message used in a regular GDP. As far as the FDB is concerned nothing else is different about the flight except for the control type.

2.2.2. FDB shall accept messages from EDCT when a pop-up flight is assigned an unassigned slot. These flights will be treated just like a regularly controlled flight except for the control type. That is, EDCT assigns the slot and control times and then sends a slot message to FDB. FDB updates its entry for the flight and then sends a message back to EDCT with other updated data such as the ERTA. EDCT sends a transaction under the message type MSG_EDCT with a control type set to 'gaap'.

2.2.3. FDB shall match such GAAP flights to entries in its database by flight ID, IGTD and airport pair.

2.2.3.1 If a matching flight is found, then FDB shall update the flight entry.

2.2.3.1.1 The control time fields and the control slot name shall be set to the data sent by EDCT.

2.2.3.1.2 The control type shall be set to GAAP.

2.2.3.1.3 The OCTD shall be set to the CTD and the OCTA shall be set to the CTA.

2.2.3.1.4 The EDCT delay flag shall be set to True.

2.2.3.2 FDB shall send a control transaction to FTM ~~with the control type set to GAAP.~~

2.2.3.3 If a matching flight is not found, then FDB shall create an entry for the flight but also display an error message in its trace pad.

The FDB and EDCT exchange data to ensure that their respective databases have the same data on controlled flights. EDCT sends the initial control information and then any subsequent changes due to updates or substitutions. FDB sends any updated information on cancellations, flight plans or airline submitted data.

2.2.4. Whenever FDB receives updated data on a GAAP controlled flight, it shall send a message to EDCT with the updated information. However, if several items are updated, it shall only send one message.

2.2.5. In all other ways, FDB shall treat a GAAP controlled flight just like a regular GDP controlled flight.

As described in the EDCT section above, an unaffiliated flight that is cancelled will lose its slot. EDCT must send a transaction to FDB to decontrol the flight. In contrast to an affiliated, cancelled flight, the unaffiliated flight will be removed from the ADL files. This means that the flight must be removed from the TDB. The FDB starts this procedure by sending a transaction to EDCT on the cancellation of every controlled flight. If the flight is unaffiliated, EDCT will send a message back to decontrol the flight; if it is affiliated, no message will come back.

2.2.6. FDB shall accept a message from EDCT when a GAAP flight is to lose its slot. The name of this transaction shall be MSG_CONTROL_CANCEL and the contents of the message shall be a normal edct to fdb transaction packet.

2.2.7. FDB shall match such flights to entries in its database by flight ID, IGTD, and airport pair and a cancelled status.

2.2.7.1 If a matching flight is found that is cancelled and has a control type of GAAP then FDB shall:

2.2.7.1.1 update the entry by removing the control times and arrival slot, and setting the control type to none.

2.2.7.1.2 remove such a flight from the TDB.

2.2.7.1.3 send a message to FTM to remove the control information.

2.2.7.2 If a matching flight is not found then shall display an error message.

2.3. ADL_DD

ADL_DD will maintain a list of open slots for each GAAP controlled airport. As with EDCT, the data must be persevered even if the program restarts and the programs running on the master and slave strings must have the same data in case a master/slave switch occurs.

2.3.1. ADL_DD shall store the names of unassigned slots for a GAAP controlled airport.

2.3.1.1 The data shall be stored in such a way as to be preserved if ADL_DD restarts.

2.3.1.2 The same data shall be stored whether the string the program is running on is master or slave.

ADL_DD receives messages from both FSM and from EDCT when a GDP of any type is issued. It receives the parameters of the program from FSM and it receives a variety of control data from EDCT. The program data received from FSM and from EDCT must match. The match is based on the airport and on the program times. If the data do not match, the parameters will not appear in the ADLs.

2.3.2. ADL_DD shall accept GAAP control information from FSM. The message type will be M_ADD_ADL_GAAPGDP_PARAM and the contents will be all of the parameters for the program. Since the GAAP program is treated like a regular GDP, no new message type is required.

2.3.2.1 ADL_DD shall incorporate the data into its internal structures.

2.3.2.2 ~~ADL_DD shall set the control type to GAAP.~~

Any time a new program is sent out, ADL_DD must remove any existing open slots.

2.3.3. When ADL_DD receives a message from EDCT that a new GDP is in place for an airport, it shall delete any unassigned slots associated with that airport. This takes care of removing unused slots prior to a revision or compression. It assumes that adl_dd matches the program from edct; if they don't match, no parameters will appear in the ADL, and the open slots will stay.

The next three requirements describe the actions that ADL_DD will take as new slots are added, slots are used, and slots are returned.

2.3.4. ADL_DD shall accept messages from EDCT on new unassigned slots. The message type will be NWA_CDM_TO_DD_ADD_SLOT and the contents of the message will be the airport and a list of all of the unassigned slots.

2.3.4.1 ADL_DD shall verify that the airport has a control type of GAAP.

2.3.4.2 ADL_DD shall add the slots to its list of unassigned slots for the airport.

2.3.5. ADL_DD shall accept messages from EDCT on assigned slots. The message type will be NWA_CDM_TO_DD_REMOVE_SLOT and the contents of the message will be the airport and the name of the slot.

2.3.5.1 ADL_DD shall remove the slot from the list of unassigned slots for that airport.

2.3.6. ADL_DD shall accept messages from EDCT on slots that have been returned to the pool of unassigned slots. The message type will be NWA_CDM_TO_DD_ADD_SLOT and the contents of the message will be the airport and the name of the slot.

2.3.6.1 ADL_DD shall add the slot to the list of slots for that airport.

2.3.7. When ADL_DD receives a message from EDCT to purge a ground delay program, ADL_DD shall remove all of the unassigned slots associated with that program.

The ADLs for a GAAP controlled airport will contain a block of data listing the open slots. If the airport has no open slots, then no list will appear.

2.3.8. ADL_DD shall display the list of unassigned slots in the ADL for that airport. This data block shall be displayed only if at least one unassigned slots exists.

2.3.8.1 START_UNASSIGNED_SLOTS.

2.3.8.2 The list of the slots by name, ~~each on a separate~~ fifty per line.

2.3.8.3 END_UNASSIGNED_SLOTS.

2.3.8.32.3.8.4 If there are no unassigned slots for the airport, then the word NONE shall be written in the block.

2.3.9. ADL_DD shall request data for thirty-six hours into the future. This is accomplished by altering the Listserver request in the adl_dd configuration file.

The next two sections deal with extending the time range of data in the flight database. Currently, scheduled flights based on the OAG are fed into FDB by the Schedule Data Base (SDB) process fifteen hours before the departure time. Airline data are also fed in fifteen hours in advance of flights' departure time from the future flights data files. This fifteen hour value will have to be increased, which will require modifying SDB and CDM_PARSER. However, there may be serious implications in FDB since many data structures and processes assume a fixed limit. The changes to SDB and CDM_PARSER are described here but they may not be implemented in the 7.9 release.

2.4. SDB

2.4.1. The SDB will send scheduled data to the FDB for flights departing 24 hours in advance of the current time. Currently, it sends them 15 hours in advance.

2.5. CDM_PARSER.

2.5.1. CDM_PARSER will send to the FDB airline data, FC and FM messages, that have an IGTD 24 hours in advance of the current time. CDM_PARSER buckets these injections in fifteen minute intervals. Every fifteen minutes it scans the appropriate future flights files, finds the flights that have an IGTD less than 15 hours in advance of the current time, and sends all of them off to the FDB.

2.5.2. CDM_PARSER will write into the appropriate future flights files airline data that have an IGTD beyond 24 hours of the current time.

3. External Systems

3.1. Listserver

3.1.1. Listserver shall display a control type of GAAP for pop-up flights assigned a slot in any list request.

3.1.2. The listserver shall allow list requests of up to 36 hours of data.

3.2. FSM

3.2.1. FSM shall send the list of open slots and the maximum delay time to Autosend.

3.3. Autosend

3.3.1. Autosend shall send the list of open slots and the maximum delay to EDCT.

4. Databases

4.1. Control Database

4.1.1. The CDB shall store the names of the unassigned slots for all GAAP controlled airports.

5. Interfaces

5.1. EDCT/ADL_DD

5.1.1. The following new message types shall be defined in the network addressing communication files.

5.1.1.1 **NWA_CDM_TO_DD_ADD_SLOT.** This message, from EDCT to ADL_DD, contains a new list of unassigned slots for an airport. The contents of the message are the name of the airport and a list of the slots.

5.1.1.2 **NWA_CDM_TO_DD_REMOVE_SLOT.** This message, from EDCT to ADL_DD, contains the name of a slot that has been assigned to a pop-up flight and, thus, should be removed from the list of unassigned slots. The contents of the message are the name of the airport and the name of the slot.

5.2. FADT File

These may not be the correct names of the data but are included here as reminders that they will be present in some form.

~~5.2.1. The file may have a new control type of GAAP.~~

5.2.1. The file shall have a section for program parameters.

5.2.1.1 The new data block shall start with START_PARAMETERS.

5.2.1.2 The new data block shall end with END_PARAMETERS.

5.2.1.3 Between the start and end shall be the following lines:

5.2.1.3.1 TYPE, for the type of program. This shall be GDP, GS, BLKT, or COMP.

5.2.1.3.2 POPUP_MODE, for the method of handling pop-up flights. The value shall be GAAP or FA.

5.2.1.3.3 START, for the start time of the program.

5.2.1.3.4 END, for the end time of the program.

5.2.1.3.5 DELAY_LIMIT, for a GAAP program only, the maximum delay that a flight may be assigned.

5.2.2. If the file has a ~~control type~~**pop-up mode** of GAAP, then it shall also have a new data block to indicate the unassigned slots.

5.2.2.1 The new data block shall start with START_UNASSIGNED_SLOTS.

5.2.2.2 The new data block shall end with END_UNASSIGNED_SLOTS.

5.2.2.3 Between the start and the end shall be a list of the slots, ~~one~~ **fifty** per line.

~~5.2.3. If the file has a control type of GAAP, then it shall also have a new data block to indicate the maximum delay time.~~

~~5.2.3.1 The new data block shall start with START_MAX_DELAY.~~

~~5.2.3.2 The new data block shall end with END_MAX_DELAY.~~

~~5.2.3.3~~**5.2.2.4** Between the start and the end shall be the value of the maximum delay.

5.3. FSM/ADL_DD

~~5.3.1. The following new message type shall be used between FSM and ADL_DD.~~

~~5.3.1.1 M_ADD_ADL_GAAP_PARAM with a value of 236. This message, from FSM to ADL_FE and ADL_DD, indicates that the included parameters are for a GAAP program and will become the operative program at the airport.~~

~~5.3.1.2 M_DEL_ADL_GAAP_PARAM with a value of 237. This message, from FSM to ADL_FE and ADL_DD, indicates that the parameters for the airport should be removed.~~

6. User Interfaces

6.1. TSD

6.1.1. The TSD shall display a new EDCT command that will allow a user to request a list of unassigned slots at an airport.

6.1.2. The name of the request shall be EDCT OPENSLOTS.

6.1.3. The dialog box shall contain a prompt for the airport name.

6.1.3.1 TSD shall format the request as it does for all other EDCT requests and send the request to the EDCT Request Server. The message type will be NWA_ASD_TO_EDCT and the contents of the message will be the request and the name of the airport as typed by the user followed by the name of a directory into which the report will be copied.

6.1.4. TSD shall display the report that comes back from this request in a text window.

6.2. Netmail

6.2.1. Netmail shall allow a user to send the EDCT OPENSLOTS request to the EDCT Request Server.

6.2.2. The request shall contain the name of the airport and a directory into which the report will be copied.

6.3. FSM

6.3.1. FSM shall allow a user to request the EDCT OPENSLOTS report.

6.3.2. FSM shall add the request to the EDCT reports menu.

6.3.3. FSM shall prompt the user for the name of the airport.

6.3.4. **FSM shall send a message to ADL_FE.** The message type will be M_EDCT_COMMAND and the contents of the message will be the request and the name of the airport.

ADL_FE will forward this request to the EDCT Request Server, which will send a report back to ADL_FE. ADL_FE will in turn send that report back to FSM.

6.3.5. **FSM shall display the report that comes back from this request in a text window.**

7. Configuration Changes

7.1.1. ADL_DD

7.1.1.1 The configuration file for ADL_DD shall be modified to send a request for thirty-six hours of data to Listserver.

8. Logging and Debugging

8.1. FDB

8.1.1. The FDB shall write the control type of GAAP to the CDM transaction files when it is notified by EDCT through the M_GAAP_FLIGHT message that a pop-up flight has been assigned a slot.

8.1.2. If FDB cannot match a GAAP controlled flight to an existing flight, it shall display an error message in its trace file. The error message shall state: "GAAP ERROR: Flight ID not matched."

8.1.3. Any programs used to display data in the FDB shall be modified to show the GAAP control type.

8.2. EDCT

8.2.1. Any programs used to display data in the CDB shall be modified to show the GAAP control type.

8.2.2. Such programs shall also be modified to display any unassigned slots associated with a GAAP controlled airport.

~~8.2.2.~~

9. Version Compatibility

9.1. *Not applicable.*

9.1.1.

10. Issues

1. Requirement 2.1.10 states that a pop-up flight in a GAAP program may be exempted. Is this correct? Or should the exemption and exclusion criteria be ignored in a GAAP program and a pop-up will always receive either a slot or a maximum delay.
2. Should a GAAP flight be treated like a GDP flight within FDB or should there be any differences that should be implemented?

11. Design Issues and Suggestions

1. None.

12. References

1. None.

13. Allocation of Software Requirements to System Requirements

The following table shows how the software requirements in this document are allocated to the system requirements.

System Requirement	Software Requirements
2.1.1	2.3.9, 2.4.1, 2.5.1, 2.5.2
2.3.1	2.1.14, 2.1.15, 2.2.5
2.3.2	2.1.1, 2.1.2, 2.1.3, 2.1.4, 2.1.5, 2.1.7, 2.1.22, 2.1.23, 2.3.1, 2.3.2, 2.3.3, 2.3.4
2.3.3	2.1.8, 2.1.9, 2.1.10, 2.1.11, 2.3.5
2.3.4	2.1.12, 2.2.1, 2.2.2, 2.2.3, 2.2.4
2.3.5	2.1.13
2.3.6	2.1.16, 2.1.17, 2.1.18, 2.2.6, 2.2.7, 2.3.6
2.3.7	2.1.6
2.3.8	2.1.19, 2.1.20
2.4.1	2.1.21, 2.3.7
2.4.2	2.1.24, 2.3.8