CONTRACT N° : G4RD-2000-00228

PROJECT N° : GRD1-1999-10516

ACRONYM : MA-AFAS

THE MORE AUTONOMOUS - AIRCRAFT IN THE FUTURE AIR TRAFFIC MANAGEMENT SYSTEM

D66 – AGP Architecture Document

AUTHOR: SKYSOFT PORTUGAL

PROJECT CO-ORDINATOR: BAE SYSTEMS

PRINCIPAL CONTRACTORS:

Airtel ATN Ltd (Ireland)
ETG (Germany)
NLR (Netherlands)
Qinetiq (UK)
EUROCONTROL (France)

ASSISTANT CONTRACTORS:

Airsys ATM (France)
AMS (Italy)
FRQ (Austria)
NATS (UK)
SC-TT (Sweden)
SOFREAVIA (France)
Alenia Difesa (Italy)
DLR (Germany)
Indra Sistemas (Spain)
SCAA (Sweden)
Skysoft (Portugal)
Stasys Limited (UK)

Report Number:
Project Reference number : **MA-AFAS – WP2.4 – D66**
Date of issue of this report : **20-June-03**
Issue No: **1.0**
PROJECT START DATE : **1/3/2000**
DURATION: **36 Months**

Project funded by the European Community under the 'Competitive and Sustainable Growth' Programme (1998-2002)

All enquiries related to this publication should be referred to:

SKYSOFT PORTUGAL

Av. Conselheiro Fernando de Sousa, Nº19-12º,
1070-072 Lisbon
Lisbon, Portugal
Tel. 351-21-382 93 66 Fax. 351-21-386 64 93

D66 – AGP Architecture Document

Document No: 66

Issue 1.0

Contains 48 pages total
Comprising:
5 pages front matter
43 pages text and figures

Compiled by:		Title:	Skysoft Technical Manager
	Tiago Sepúlveda	Date:	
Approved by:		Title:	Skysoft Project Manager
	José Neves	Date:	
Authorised by:		Title:	MA-AFAS Project Manager
	Alfie Hanna	Date:	

LIST OF EFFECTIVE PAGES AND CHANGE HISTORY

Insert latest changed pages. Destroy superseded pages

TOTAL NUMBER OF PAGES IN THIS PUBLICATION IS 48 CONSISTING OF THE FOLLOWING							
Page No.	Date	Issue	DCR	Page No.	Date	Issue	DCR
All							

DISTRIBUTION LIST

This Document is distributed as below.

Additional copies held by unnamed recipients will not be updated.

Paper Copy No.	Name	Address
1	MA-AFAS Library	Avionic Systems
MASTER	Library	BAE SYSTEMS, Rochester

Electronic Copy	Name	Address
1	European Commission	EC, Brussels
2-20	MA-AFAS Consortium	maafas@bluecoat.org

1 SCOPE

1.1 Identification

This architecture document applies to the MA-AFAS Project AOC Ground Platform (AGP). It has been produced by Skysoft Portugal and AMS-UK for the MA-AFAS programme on behalf of BAE Systems.

The objective of this document is to provide a description of the AGP software architecture.

1.2 System Overview

The “More Autonomous Aircraft in the Future ATM System” (MA-AFAS) program is focused on developing CNS-based avionics components that will provide aircraft greater flexibility within the ATM system. The program includes development of an operational concept; specification and implementation of avionics packages, ground systems and infrastructure to demonstrate the operational concept; and trials and further work towards implementation of the concept.

One of the themes of the programme is the Airline Operational Centre (AOC). MA-AFAS functional requirements for the Airborne AOC system are addressed in the following project documentation

- D13, Definition of ATM MA-AFAS Airborne and Ground Functionalities.
- D18, Airborne Systems Requirement Specification.
- D19, Airborne Equipment System Design and Architecture Document

To support trials of the MA-AFAS avionics package AOC functions, an AOC Ground Platform (AGP) was specified and developed by the AOC Theme. The AGP is a trials system which simulates the functions of an AOC end system on the ground. The aim is to validate the correct functioning of the avionics and crew interactions as well as the air-ground interoperability under real flight conditions. The platform is responsible for the support of four AOC functional areas:

- AOC Flight Plan
- AOC and A/C Maintenance
- A/C AOC CDM
- Asset Management

To enable a correct operational use of the all the system, additional functional aspects were included as part of the AGP platform environment:

- System Communications
- HMI

A description of the AGP high-level requirements and main functionality is included in [MA-AFAS D38].

1.3 Document Overview

The present document is structured as follows:

Chapter 1: The present chapter, containing a brief introduction to the document, together with the sketch of its structure.

Chapter 2: Provides the list of referenced documents.

Chapter 3: It contains an overview of the AGP software.

Chapter 4: Describes the AGP software architecture.

Chapter 5: Presents the list of abbreviations used within this document.

2 REFERENCE DOCUMENTS

[RTCA 178B]	RTCA Software Considerations in Airborne Systems and Equipment Certification, December 1, 1992.
[MAAFAS D38]	MA-AFAS Ground System Requirements, Project Delivery D38, Issue 1
[MAAFAS ICD]	MA-AFAS AOC Air-Ground Interface Control Document – Issue 2

3 SOFTWARE OVERVIEW

The following diagram provides the AGP context diagram.

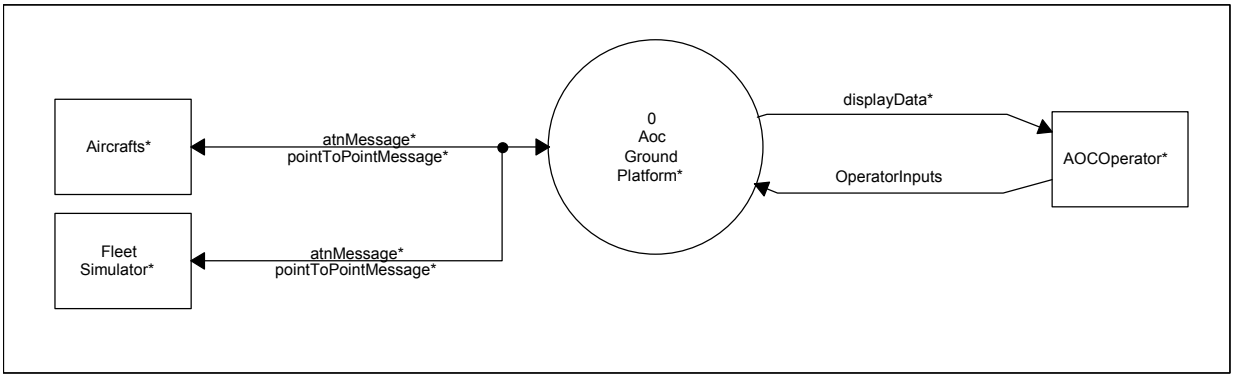

Figure 1 – AGP Context Diagram.

The AGP context diagram (**Figure 1**) is an abstraction of the AGP and all entities that interact with it. These entities are described below:

- The AOC Ground Platform is the application that supports the monitoring of all aircraft from an airline perspective.
- The AOC Operator will be a user that provides inputs to the AGP, and uses the visual data presented on the AGP displays.
- The Aircraft are the entities that will exchange messages/data with AGP.
- The Fleet Simulator is a simulator of a certain number of aircraft and respective flights.

4 SOFTWARE ARCHITECTURE

4.1 AocGroundPlatform

The diagram below (Figure 2) provides the functional breakdown of the AGP process into its three main modules: Communications, AGP Functions and HMI.

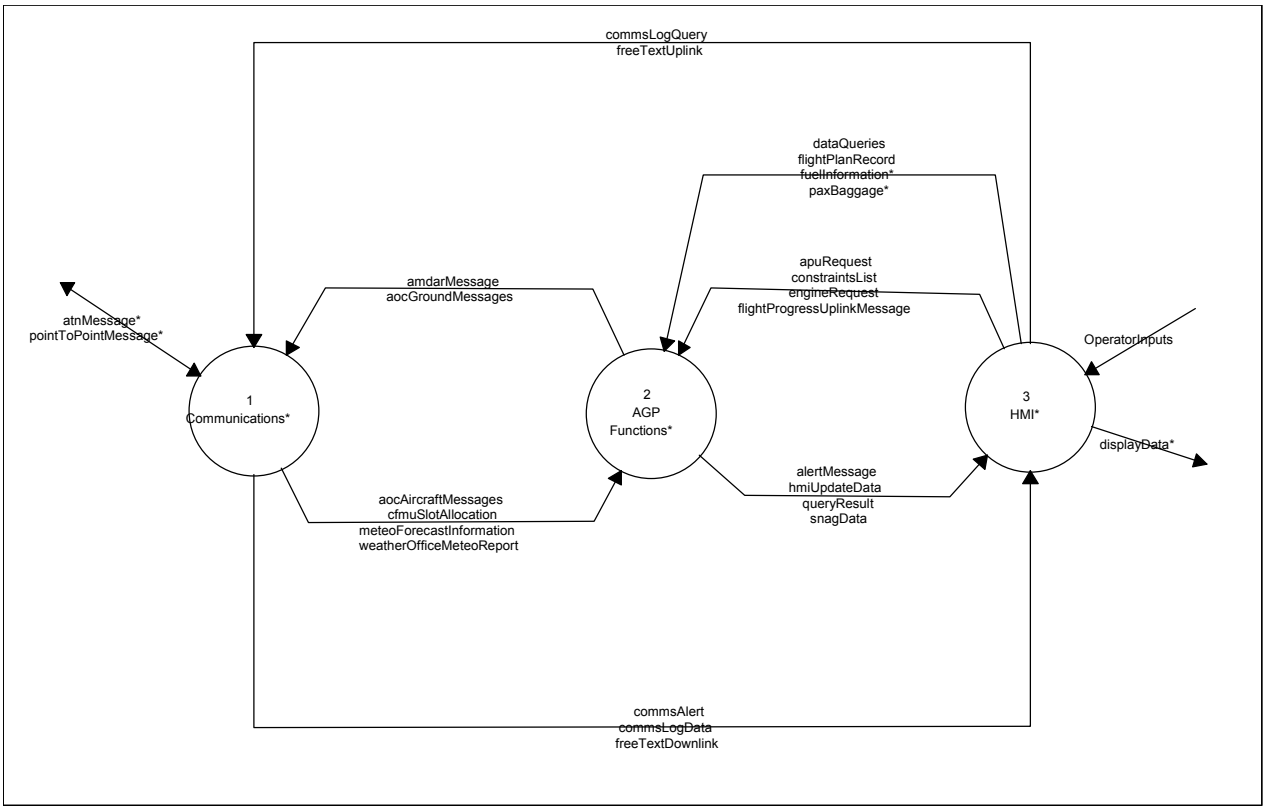

Figure 2 – AocGroundPlatform Process DFD.

Process Inputs:
- atnMessage
- pointToPointMessage
- OperatorInputs

Process Outputs:
- atnMessage
- pointToPointMessage
- displayData

4.1.1 Communications

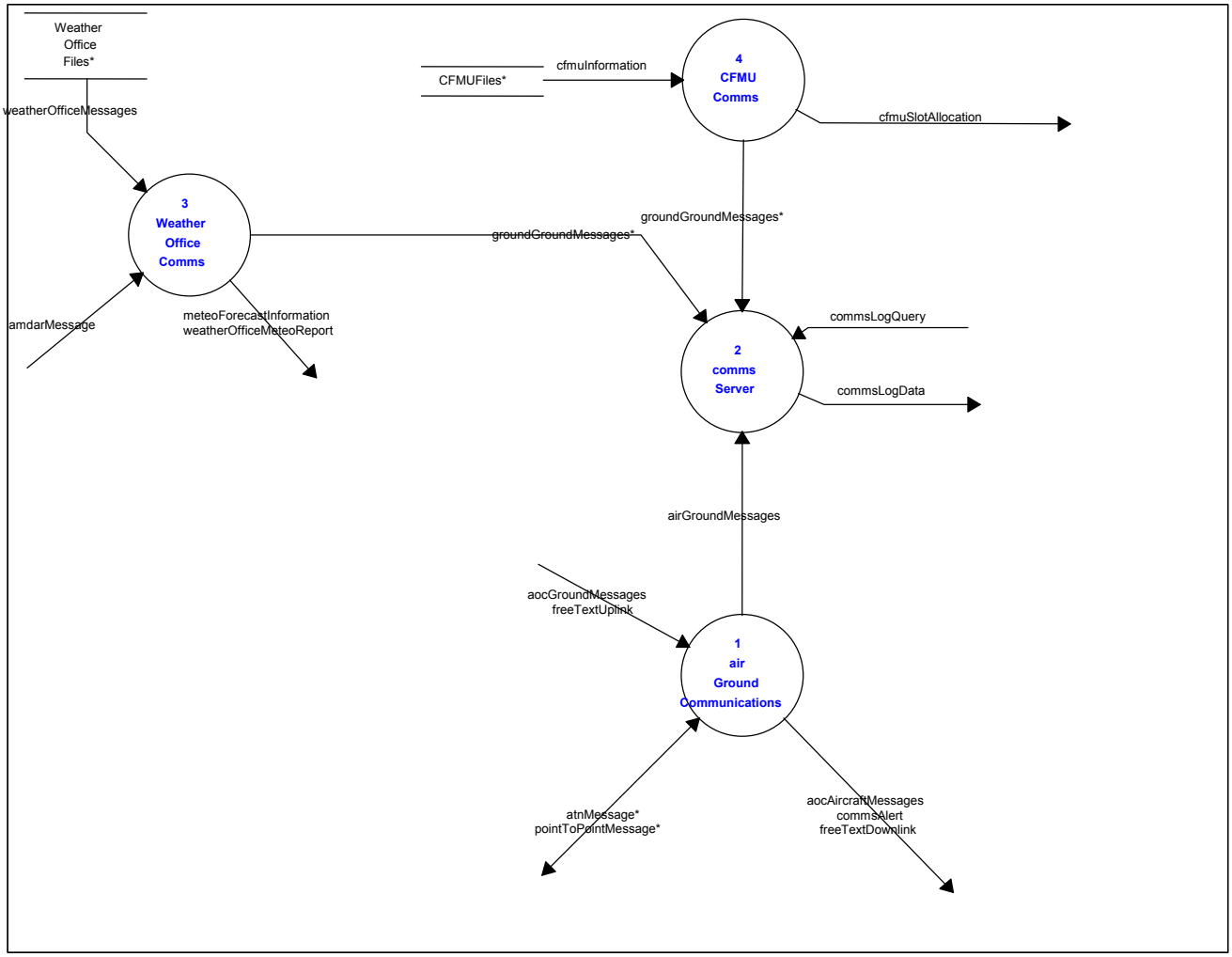

Figure 3 – Communications Process DFD.

Process Inputs:
- atnMessage
- pointToPointMessage
- commsLogQuery
- freeTextUplink
- amdarMessage
- aocGroundMessages

Process Outputs:
- atnMessage
- pointToPointMessage
- aocAircraftMessages
- cfmuSlotAllocation

- meteoForecastInformation
- weatherOfficeMeteoReport
- commsAlert
- commsLogData
- freeTextDownlink

4.1.1.1 airGroundCommunications

This process is responsible to handle the exchange of messages between the AGP and the company aircrafts. The Air-Ground Communications process uses GACS (Generic ATN Communications service) for communicating between air and ground stations. To communicate with GACS, this process uses GAPI (GACS Application Programming Interface). More details about GAPI can be found on GACS User Guide v1.3.

Process Inputs:
- aocGroundMessages from AGPFunctions (2)
- freeTextUplink from FreeTextTelex (3.4)
- atnMessage from Aircraft
- pointToPointMessage from Aircraft

Process Outputs:
- atnMessage to Aircraft
- pointToPointMessage to Aircraft
- aocAircraftMessages to AGPFunctions (2)
- commsAlert to FlightProgressDisplay(3.2)
- freeTextDownlink to FlightProgressDisplay(3.2)

Processing:
Receives a message from the AGP Functions process (2). The received messages are converted into a human-readable form. Afterwards the messages are converted to a format suitable for transmission and both the ready-to-transmit message and the human-readable message are sent to the comms server (1.2) to be stored. Finally, the message is sent to the aircraft using GAPI. The underlying communication structure will either construct a point-to-point message or an ATN message, depending on the network attached.

Receives a free text message from the free Text Telex process (3.4). The free text message is converted into a format suitable for transmission. Both the ready-to-transmit message and the original message are sent to comms server (1.2) to be stored. Sends the message to aircraft using GAPI. The underlying communication structure will construct either a point-to-point message or an ATN message, depending on the network attached.

Receives a message from an aircraft. Because some received messages are transmitted in a non-readable format, it may be required the parsing of the message to a human-readable form. After parsing the received message, the parsed message and the unparsed message are sent to the comms server (1.2) to be stored. Sends the message to AGP Functions (2).

Receives a free text message from the specified aircraft. Sends the message to comms server (1.2) to be stored. Sends the message to the flight progress display (3.2).

If the confirmation of a message sent using the GACS confirmed service does not arrives within a pre-defined time, the air-ground communications process sends an alert to the flight progress display (3.2).

Requirements Satisfied:
AGP_COM_001, AGP_COM_002, AGP_COM_003, AGP_COM_004, AGP_COM_005

4.1.1.2 commsServer

This process is responsible for providing a facility to store and retrieve all messages exchanged between the AOC Ground Platform and external entities (aircraft and other ground systems). Messages are timestamped on logging, and can be retrieved according to the flights to which they relate. The messages are stored in two forms, first the external form (i.e. exactly as the message was received, or transmitted), and secondly in a human-readable form for display.

This functionality is to be implemented in the phase 1 deliverable.

Process Inputs:
- groundGroundMessages from CFMUComms (1.4) and from WeatherOfficeComms (1.3)
- airGroundMessages from airGroundCommunications (1.1)
- commsLogQuery from HMI (3)

Process Outputs:
- commsLogData to HMI (3)

Processing:
Receives copies of all messages that are received by or sent from the meteo and CFMU interface (1.3 and 1.4) and air-ground communication (1.1) processes. When the communication processes send messages to the server, both the external (unparsed) and human-readable (parsed) forms of the original message are sent. All messages received by the comms server are stored in a database. Human-readable forms of the messages are stored so that when a log query is made, the stored message can be sent directly to the HMI to be displayed without having to parse the messages again.

Receives a comms log query from the HMI (3), containing details about the stored messages that are required. The human-readable forms of the messages are retrieved from the log and sent to the HMI (3) for display to the operator.

Requirements Satisfied:
AGP_HMI_020

4.1.1.3 WeatherOfficeComms

This process is responsible for simulating communications between the AGP and the weather office.

Process Inputs:
- weatherOfficeMessages from WeatherOfficeFiles
- amdarMessage from AircraftMeteoReports (2.2.1)

Process Outputs:
- groundGroundMessages to CommsServer (1.2)
- meteoForecastInformation to WeatherInformation (2.2.2)
- weatherOfficeMeteoReport to WeatherInformation (2.2.2)

Processing:
This process receives weather messages from the weather office files store. Weather messages are either meteo reports or meteo forecast files. Both kinds of messages are provided in ASCII format, therefore they do not imply conversion to human-readable form. The weather messages are sent to the comms server (1.2) to be logged. Finally, the received meteo data is sent to the weather information process (2.2.2).

On receipt of an AMDAR message from aircraft meteo reports process (2.2.1), if needed (TBD), the process will convert the AMDAR message into a format suitable for transmission. Afterwards, the ready-to-transmit message will be converted into a human-readable form. The message will then be transmitted and sent to the comms server (1.2) to be logged.

In the current implementation, the AMDAR message is not transmitted to an external entity; instead a message simulating the transmission is sent to a HMI component.

Requirements Satisfied:
AGP_COM_011, AGP_COM_070, AGP_COM_080, AGP_COM_090

4.1.1.4 CFMUComms

This process is responsible for simulating the communications protocols between the AOC Ground Platform and CFMU. The process simulates the transmission to CFMU of outbound messages by sending them to the HMI, where they are displayed.

This functionality is to be implemented in the phase 1 deliverable.

Process Inputs:
- cfmuInformation from CFMUFiles

Process Outputs:
- cfmuSlotAllocation to CompanyFlightPlanningDataServer (2.3)
- groundGroundMessages to CommsServer (1.2)

Processing:
Reads slot allocations and sends them to the company flight planning data server (2.3). Sends a copy of the message to the comms server (1.2), to be stored.

Requirements Satisfied:
AGP_COM_050 (Post MA-AFAS), AGP_COM_060 (Post MA-AFAS)

4.1.2 AGPFunctions

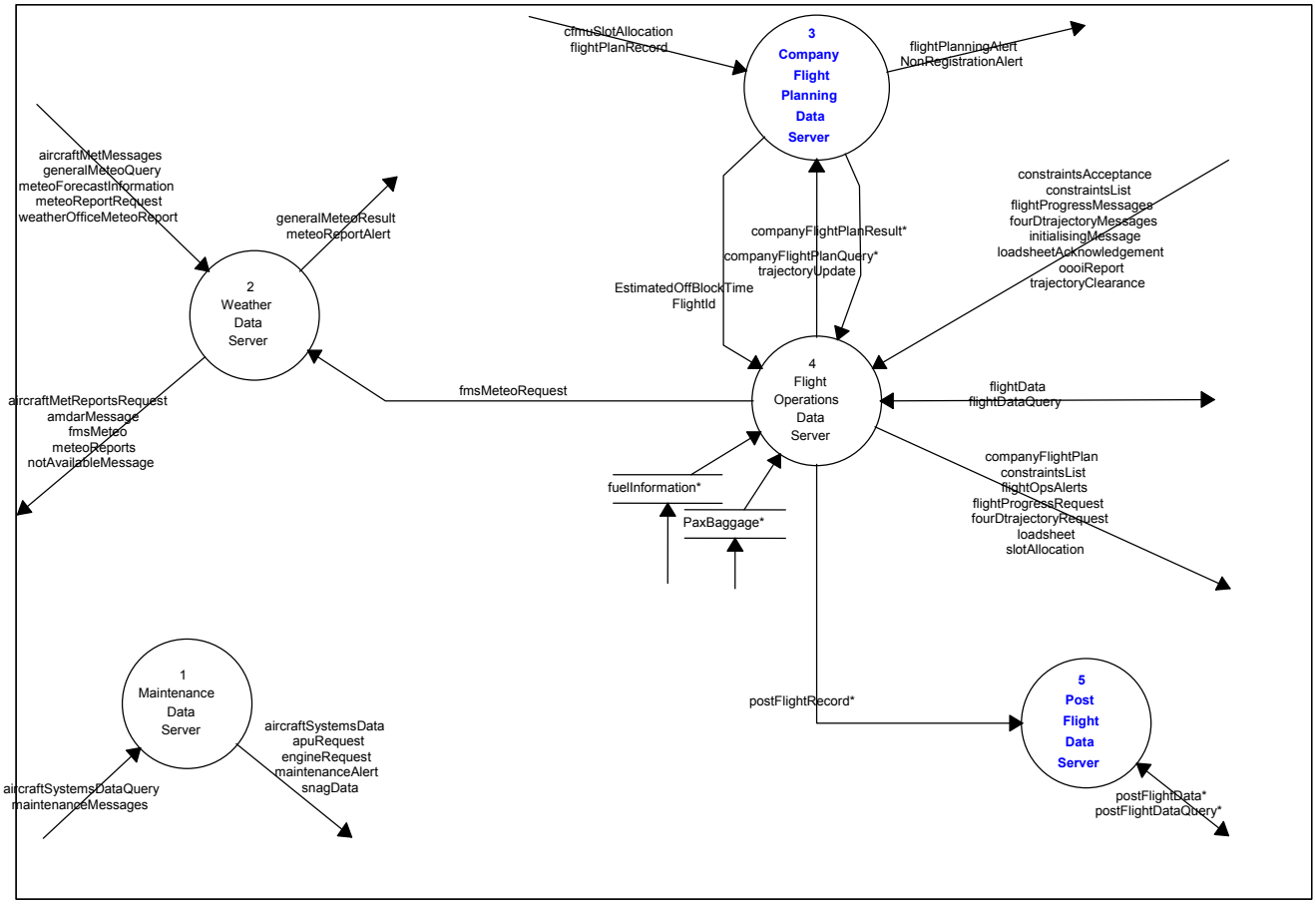

Figure 4 – AGPFunctions Process DFD.

Process Inputs:
- aocAircraftMessages
- cfmuSlotAllocation
- meteoForecastInformation
- weatherOfficeMeteoReport
- apuRequest
- constraintsList
- engineRequest
- flightProgressUplinkMessage
- dataQueries
- flightPlanRecord
- fuelInformation
- paxBaggage

Process Outputs:
- alertMessage

- hmiUpdateData
- queryResult
- snagData
- amdarMessage
- aocGroundMessages

4.1.2.1 MaintenanceDataServer

This process is responsible for the storage of maintenance reports (SNAG, APU, and Engine Status) received from company fleet aircrafts, and for the provision of data to AOC maintenance console.

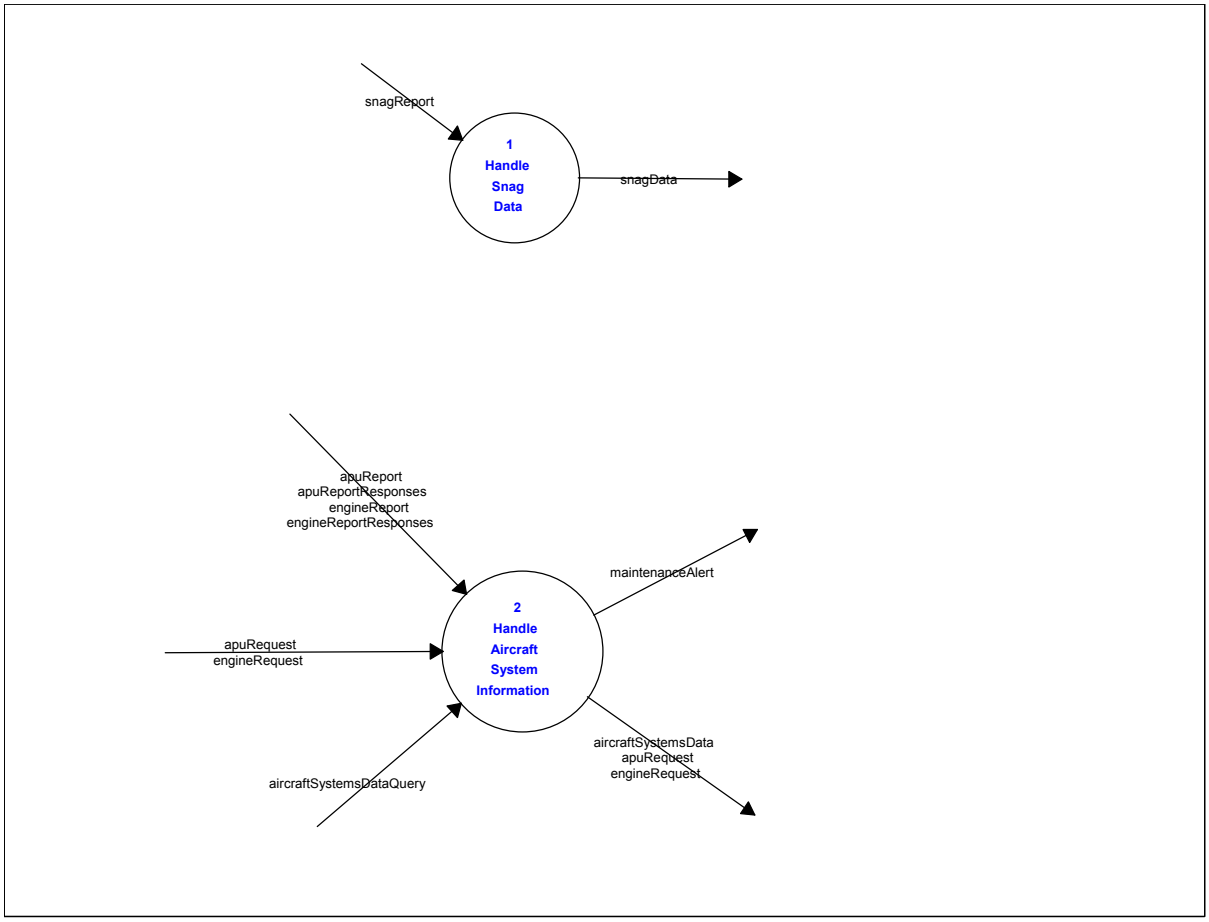

Figure 5 – AGP Functional Breakdown.

Process Inputs:
- aircraftSystemsDataQuery
- maintenanceMessages

Process Outputs:
- aircraftSystemsData
- apuRequest
- engineRequest
- maintenanceAlert
- snagData

4.1.2.1.1HandleSnagData

This process is responsible for the storage and retrieval of SNAG report data (information about failures and technical malfunctions of aircraft systems).

Process Inputs:
- snagReport from AirGroundCommunication (1.1)

Process Outputs:
- snagData to AOCMaintenanceConsole (3.3)

Processing:
Receives Snag report data. Stores the data. Sends it to the HMI maintenance console (3.3).

4.1.2.1.2 HandleAircraftSystemInformation

The second sub process included in maintenance data server is responsible for dealing with information about aircraft's internal systems (APU and Engine), and providing this information to maintenance console.

Process Inputs:
- apuReport from AirGroundCommunications (1.1)
- apuReportResponses afrom AirGroundCommunications (1.1)
- engineReport from AirGroundCommunications (1.1)
- engineReportResponses from AirGroundCommunications (1.1)
- aircraftSystemsDataQuery from AOCMaintenanceConsole (3.3)
- apuRequest from AOCMaintenanceConsole (3.3)
- engineRequest from AOCMaintenanceConsole (3.3)

Process Outputs:
- aircraftSystemsData to AOCMaintenanceConsole (3.3)
- apuRequest to AirGroundCommunications (1.1)
- engineRequest to AirGroundCommunications (1.1)
- maintenanceAlert to AOCMaintenanceConsole (3.3)

Processing:
Receives from the HMI maintenance console (3.3) a query about aircraft systems data. Sends the requested data back to the HMI maintenance console (3.3).

Receives from the HMI maintenance console (3.3) requests to set transmission contracts regarding aircraft's APU reports or engine status reports. These requests are stored and then sent to the air-ground communications process (1.1).

Receives from air-ground communications (1.1) a report conveying the data that was set by a former request, or receives a response from air-ground communications (1.1) stating that the A/C is not capable of meeting the contract.

Sends a maintenance alert to the maintenance console process (3.3), whenever an expected maintenance report is not received in a expected pre-defined time.

Requirements Satisfied:
AGP_MTC_001, AGP_MTC_011, AGP_MTC_015, AGP_MTC_016, AGP_MTC_027, AGP_MTC_028, AGP_MTC_029, AGP_MTC_030, AGP_MTC_031

4.1.2.2 WeatherDataServer

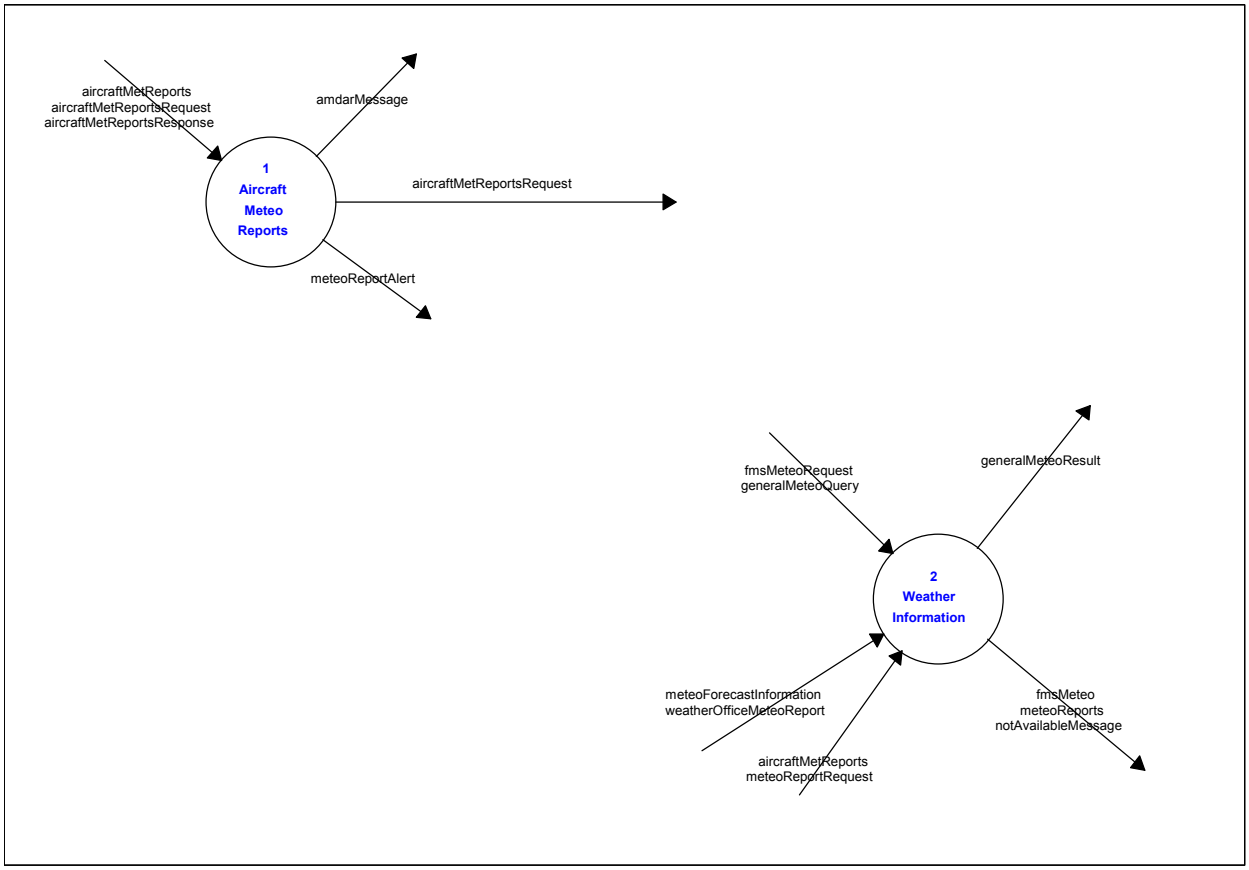

Figure 6 – WeatherDataServer Process DFD.

Process Inputs:
- aircraftMetMessages
- generalMeteoQuery
- meteoForecastInformation
- meteoReportRequest
- weatherOfficeMeteoReport
- fmsMeteoRequest

Process Outputs:
- aircraftMetReportsRequest
- amdarMessage
- fmsMeteo
- meteoReports
- notAvailableMessage
- generalMeteoResult
- meteoReportAlert

4.1.2.2.1 AircraftMeteoReports

This process is responsible for obtaining weather data from aircraft, and reformatting the data received for transmission to the Weather Office. The operator triggers requests for aircraft reports via the HMI; these requests can be either for data on demand or for data on a periodic basis.

This functionality is to be implemented in the phase 2 deliverable.

Process Inputs:
- aircraftMetReports from AirGroundCommunications (1.1)
- aircraftMetReportsRequest from FlightProgressDisplay (3.2)
- aircraftMetReportsResponse from AirGroundCommunications (1.1)

Process Outputs:
- amdarMessage to MeteoInterface (1.3)
- meteoReportAlert to FlightProgressDisplay (3.2)
- aircraftMetReportsRequest to AirGroundCommunications (1.1)

Processing:
Receives a request from the flight progress display (3.2) for an aircraft meteo report, containing:

- the Id number of the aircraft from which the meteo report is required,
- the data that is required in the report, and
- the contract details for the report.

Creates an aircraft meteo report request message, containing the above details. Sends the message to air-ground communications (1.1).

If a response message or a meteo report is not received within a pre-defined time, a meteo report alert is created and sent to the flight progress display (3.2).

Receives an aircraft meteo report response from air-ground communications (1.1). The response is either ACK, in which case a report should arrive shortly, or NACK, if the aircraft cannot supply the requested report. If the response is NACK, a meteo report alert is created and sent to the flight progress display (3.2). If the response is ACK, and a report is not received within a pre-defined time, a meteo report alert is created and sent to the flight progress display (3.2).

Receives the requested aircraft meteo report from air-ground communications (1.1).

Processes the data, to convert it into AMDAR format. Sends the AMDAR report to the meteo interface process (1.4). If the contract type of the message is for periodic reports, a timer is set up to wait for the next report. If the next report is not received within the specified time, a meteo report alert is created and sent to the flight progress display (3.2).

Requirements Satisfied:
AGP_AMG_210, AGP_AMG_220, AGP_AMG_230, AGP_AMG_240

4.1.2.2.2 WeatherInformation

The weather information process is responsible for the storage and management of weather data such as meteo forecast information, aircraft meteo reports and weather meteo reports. This process must also provide facilities to supply requested meteo data to other processes.

Process Inputs:
- meteoForecastInformation from WeatherOfficeComms (1.5)
- weatherOfficeMeteoReport from WeatherOfficeComms (1.5)
- fmsMeteoRequest from PreFlightSupport (2.4.2)
- generalMeteoQuery from FlightProgressDisplay (3.2)
- aircraftMetReports from AirGroundCommunications (1.1)
- meteoReportRequest from AirGroundCommunications (1.1)

Process Outputs:
- fmsMeteo to AirGroundCommunications (1.1)
- meteoReports to AirGroundCommunications (1.1)
- notAvailableMessage to AirGroundCommunications (1.1)
- generalMeteoResult to CompanyFlightPlanningDataServer (2.3)

Processing:
When this process receives meteo forecasts from weather office communications (1.5), it will store the received information.

Receives a request for FMS meteo data from pre flight support (2.4.2) and, based in the stored meteo forecast information, the process creates and sends the requested data to air-ground communications (1.1).

This process puts together aircraft meteo reports and meteo forecast data to make a meteorological database. Later, this process receives a general meteo query from the flight progress display (2.3). The requested data is retrieved and sent to the flight progress display (2.3).

Receives and stores weather office meteo reports from the weather office comms (1.5).

Receives meteo report requests from air-ground communications (1.1). If the requested report is available, sends it back to the air-ground communications (1.1), else, if the queried report is not available, sends a NOT AVAILABLE message to air-ground communications (1.1).

Requirements Satisfied:
AGP_FPL_240, AGP_FPL_250, AGP_FPL_260, AGP_FPL_270, AGP_FPL_300

4.1.2.3 CompanyFlightPlanningDataServer

This process is responsible for the management of flight plans. This includes the filing of flight plans with simulated ATC authorities, and interaction with the simulated CFMU flow management procedures (limited for MA-AFAS purposes to the reception of departure slot allocations).

This functionality is to be implemented in the phase 1 deliverable.

Process Inputs:
- cfmuSlotAllocation from CFMUComms (1.4)
- flightPlanRecord from FlightPlanningConsole (3.1)
- companyFlightPlanQuery from RegisterNewAircraft (2.4.1)
- trajectoryUpdate from FlightOperationsDataServer (2.4)

Process Outputs:
- flightPlanningAlert to FlightProgressDisplay (3.2)
- NonRegistrationAlert to FlightProgressDisplay (3.2)
- EstimatedOffBlockTime to PreFlightSupport (2.4.2)
- FlightId to FlightOperationsDataServer (2.4)
- companyFlightPlanResult to RegisterNewAircraft (2.4.1)

Processing:
Receives a previously generated flight plan from the flight planning console (3.1). Sends the flight plan to the flight plan data store. If an aircraft has not registered for this flight by a pre-defined time before the EOBT, creates a non-registration alert and sends it to the flight progress display (3.2).

Receives details of a slot allocation for a flight, containing details of a flight plan to be filed with ATC authorities. Stores the slot allocation data in the database. For each slot allocation, an entry is added to the timer lists for the specified time. When this time is reached, a slot allocation message is sent to the flight progress display (3.2). The stored flight plan and take-off time are updated. The estimated off-block time is recalculated and sent to pre-flight support (2.4.2), as well as the new slot allocation.

Receives a company flight plan query from register new aircraft (2.4.1), containing a request for a specific flight plan. Retrieves the flight plan from data store. Sends the company flight plan result to register new aircraft (2.4.1). Update the stored flight plan to indicate that the registration has been received.

A flight planning alert is generated if a specified condition occurs, for example a company flight plan not being sent to the CFMU simulation before a specified time. An alert is constructed and sent to the flight progress display (3.2).

Requirements Satisfied:
AGP_FPL_040 (Post MA-AFAS), AGP_FPL_050, AGP_FPL_060, AGP_FPL_140

4.1.2.4 FlightOperationsDataServer

This process is responsible for the air/ground communications applications relating to the aircraft operations.

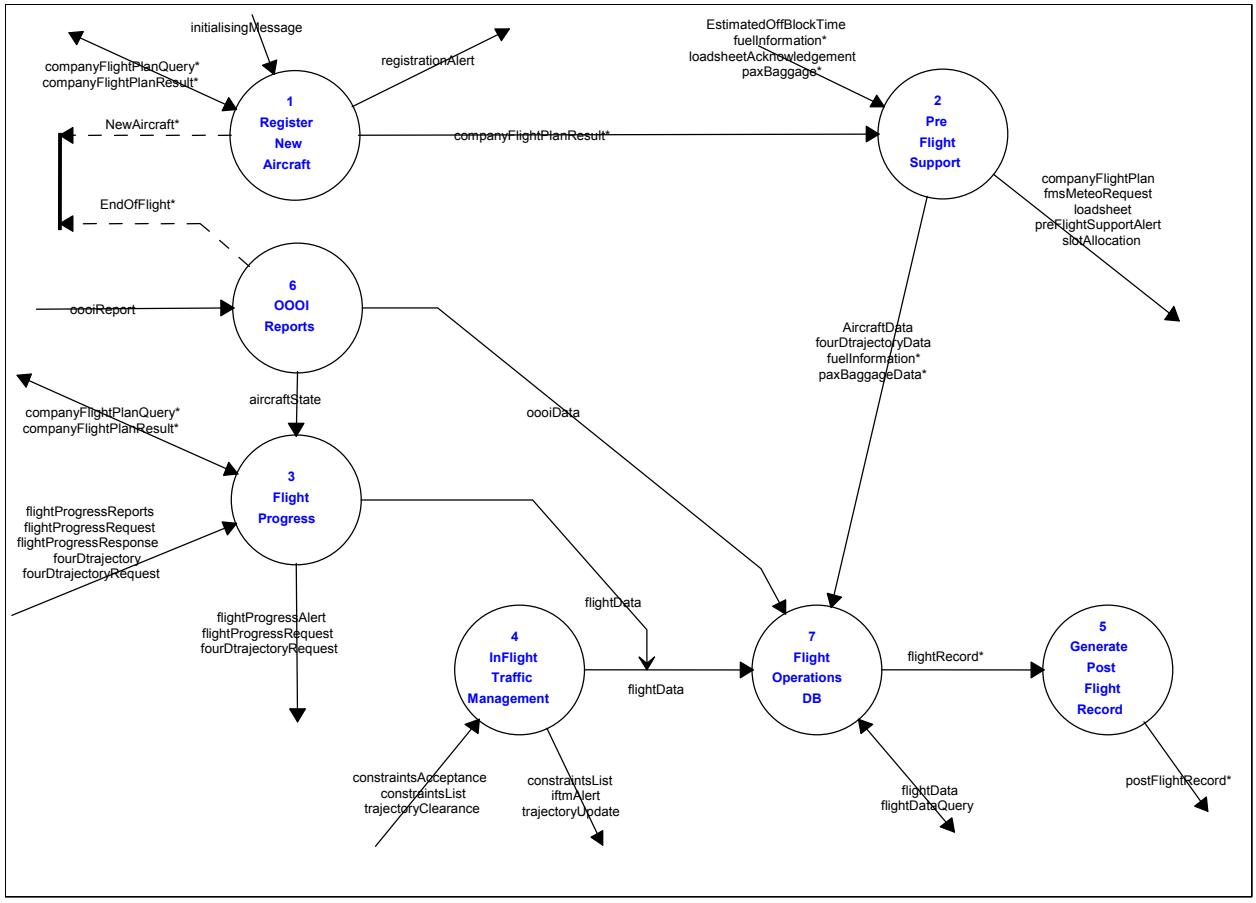

Figure 7 – FlightOperationsDataServer Process DFD.

Process Inputs:

- constraintsAcceptance
- constraintsList
- flightProgressMessages
- fourDtrajectoryMessages
- initialisingMessage
- loadsheetAcknowledgement
- oooiReport
- trajectoryClearance
- flightData
- flightDataQuery
- fuelInformation
- paxBaggage
- EstimatedOffBlockTime

- FlightId
- companyFlightPlanResult

Process Outputs:
- fmsMeteoRequest
- companyFlightPlanQuery
- trajectoryUpdate
- companyFlightPlan
- constraintsList
- flightOpsAlerts
- flightProgressRequest
- fourDtrajectoryRequest
- loadsheet
- slotAllocation
- flightData
- flightDataQuery
- postFlightRecord

4.1.2.4.1RegisterNewAircraft

This process is responsible for handling the registration of a flight as being ready to participate in AOC datalink communications. This registration is used primarily to associate a particular flight plan with an instance of communication. It also is used to trigger the pre-flight dialogue with the aircraft.

This functionality is to be implemented in the phase 1 deliverable.

Process Inputs:
- initialisingMessage from AirGroundCommunications (1.1)
- companyFlightPlanQuery from FlightProgressDisplay (3.2)
- companyFlightPlanResult from CompanyFlightPlanningDataServer (2.3)

Process Outputs:
- companyFlightPlanQuery to CompanyFlightPlanningDataServer (2.3)
- companyFlightPlanResult (NewAircraft trigger) to PreFlightSupport (2.4.2) and to FlightProgressDisplay (3.2)
- registrationAlert to FlightProgressDisplay (3.2)

Processing:
Receives an initialising message from air-ground communications (1.1), containing details of an aircraft registering for a flight. Creates a flight plan query containing the contents of the initialisation message (tail number, and optionally flight id, departure and destination), and sends it to the company flight planning data server (2.3). Receives a flight plan result from the company flight planning data server (2.3), containing the flight plan for the specified flight. Passes the flight plan result to pre-flight support (2.4.2), triggering the pre-flight support processing.

A registration unknown alert is generated if the registration cannot be associated with a company flight plan, and sent to the flight progress display (3.2).

Requirements Satisfied:
AGP_FPL_010, AGP_FPL_230

4.1.2.4.2 PreFlightSupport

This process is responsible for managing the AOC Ground Platform-aircraft dialogues which are specific to the period between the aircraft initial registration and departure (pushback). This includes triggering FMS meteo and company flight plan uploads, and providing the loadsheet.

This functionality is to be implemented partly in the phase 1 deliverable and partly in the phase 2 deliverable, as noted below.

Phase 1:
Process Inputs:
- EstimatedOffBlockTime from CompanyFlightPlanningDataServer (2.3)
- companyFlightPlanResult from RegisterNewAircraft (2.4.1)

Process Outputs:
- AircraftData to FlightOperationsDB (2.4.7)
- fourDtrajectoryData to FlightOperationsDB (2.4.7)
- companyFlightPlan to AirGroundCommunications (1.1)
- fmsMeteoRequest to WeatherInformation (2.2.2)
- preFlightSupportAlert to FlightProgressDisplay (3.2)
- slotAllocation to AirGroundCommunications (1.1)

Processing:
Receives the company flight plan for an aircraft which has just been registered with the AGP from the register new aircraft process (2.4.1). Sends all aircraft and flight details to the flight operations database (2.4.7). Creates a company flight plan message for the flight, and sends it to air-ground communications (1.1). Creates an FMS meteo request message for the flight, and sends it to the weather information process (2.2.2).

Receives the EOBT for a flight from the company flight planning data server (2.3). Sends the EOBT and slot allocation to the flight operations database (2.4.7). If an aircraft has registered for the flight, a slot allocation message is sent to air-ground communications (1.1). If a slot allocation acknowledgement has not been received from air-ground communications (1.1) by a pre-determined time before the EOBT, creates a pre-flight alert and sends it to the flight progress display (3.2).

Phase 2:
Process Inputs:
- EstimatedOffBlockTime from CompanyFlightPlanningDataServer (2.3)
- fuelInformation from AuxiliaryHMI (3.7)
- loadsheetAcknowledgement from AirGroundCommunications (1.1)
- paxBaggage from AuxiliaryHMI (3.7)
- companyFlightPlanResult from RegisterNewAircraft (2.4.1)

Process Outputs:
- AircraftData to FlightOperationsDB (2.4.7)
- fourDtrajectoryData to FlightOperationsDB (2.4.7)
- fuelInformation to FlightOperationsDB (2.4.7)
- paxBaggageData to FlightOperationsDB (2.4.7)
- companyFlightPlan to AirGroundCommunications (1.1)
- fmsMeteoRequest to WeatherInformation (2.2.2)

- loadsheet to AirGroundCommunications (1.1)
- preFlightSupportAlert to FlightProgressDisplay (3.2)
- slotAllocation to AirGroundCommunications (1.1)

Processing:
Receives details of how much fuel has been loaded onto the aircraft from the auxiliary HMI (3.7). In the context of the AGP trials, this information shall be operator-defined. Sends the fuel information to the flight operations database (2.4.7) to be stored.

Receives details of the passenger and baggage loads on the aircraft from the auxiliary HMI (3.7). In the context of the AGP trials, this information shall be operator-defined. Sends the information to the flight operations database (2.4.7) to be stored.

Receives a loadsheet trigger. This may either be on a timed basis (a specified number of minutes before estimated departure time) or when an aircraft is registered.

If the information required for the loadsheet is available:

- Creates a loadsheet containing the fuel, passenger and baggage information for the flight, and sends it to air-ground communications (1.1). If a valid loadsheet acknowledgement is not received from air-ground communications (1.1) within a pre-defined time, creates a pre-flight alert and sends it to the flight progress display (3.2).

If the information required for the loadsheet is not available:

- Wait until all information required is available, then create and send the loadsheet as above.

If the loadsheet has not been sent a pre-defined time before EOBT, creates a pre-flight alert and sends it to the flight progress display (3.2).

Requirements Satisfied:
AGP_FPL_110, AGP_FPL_130, AGP_FPL_170, AGP_FPL_180, AGP_FPL_190, AGP_FPL_200, AGP_FPL_210, AGP_COM_015

4.1.2.4.3 FlightProgress

This process is responsible for managing the applications by which the AOC Ground Platform can track the progress of the flight against its flight plan, while the aircraft is en-route. The applications includes request for 4D trajectory, and the flight progress report application itself. The operator triggers requests for flight progress reports via the HMI; these requests can be either for data on demand or for data on a periodic basis.

This functionality is to be implemented partly in the phase 1 deliverable and partly in the phase 2 deliverable, as noted below.

Phase 1:
Process Inputs:
- fourDtrajectory from AirGroundCommunications (1.1)
- fourDtrajectoryRequest from FlightProgressDisplay (3.2)

Process Outputs:
- flightData to FlightOperationsDB (2.4.7)
- flightProgressAlert to FlightProgressDisplay (3.2)
- fourDtrajectoryRequest to AirGroundCommunications (1.1)

Processing:
Receives a 4D trajectory request from the flight progress display (3.2). Sends the request to air-ground communications (1.1). If the 4D trajectory data is not received within a pre-defined time, a flight progress alert is created and sent to the flight progress display (3.2).

Receives the current 4D trajectory from air-ground communications (1.1). Checks the trajectory data for validity. Sends a flight data query to the flight operations database (2.4.7). Receives the stored flight data from the flight operations database (2.4.7). Checks the current trajectory data against the stored trajectory data. If significantly different, a flight progress alert is sent to the flight progress display (3.2). Sends the trajectory to the flight operations database (2.4.7) to be stored. If the trajectory report was requested by the operator, the trajectory is also sent to the flight progress display (3.2).

Phase 2:
Process Inputs:
- flightProgressReports from AirGroundCommunications (1.1)
- flightProgressRequest from FlightProgressDisplay (3.2)
- flightProgressResponse from AirGroundCommunications (1.1)
- fourDtrajectory from AirGroundCommunications (1.1)
- fourDtrajectoryRequest from FlightProgressDisplay (3.2)
- aircraftState from OOOIReports (2.4.6)
- companyFlightPlanQuery from FlightProgressDisplay (3.2)
- companyFlightPlanResult from CompanyFlightPlanningDataServer (2.3)

Process Outputs:
- flightData to FlightOperationsDB (2.4.7)
- flightProgressAlert to FlightProgressDisplay (3.2)
- flightProgressRequest to AirGroundCommunications (1.1)
- fourDtrajectoryRequest to AirGroundCommunications (1.1)
- companyFlightPlanQuery to CompanyFlightPlanningDataServer (2.3)

- companyFlightPlanResult to FlightProgressDisplay (3.2)

Processing:
Receives a flight progress request from the flight progress display (3.2). Sends the request to air-ground communications (1.1). If a flight progress response or flight progress report is not received within a pre-defined time of the request being made, a flight progress alert is created and sent to the flight progress display (3.2).

Receives a flight progress response from air-ground communications (1.1), containing details of whether the aircraft can satisfy the flight progress request. If the response is ACK, the response can be sent as part of the flight progress report. If not, this separate response message can be used. If the response is NACK, creates a flight progress alert containing the details of the rejection, and sends it to the flight progress display (3.2).

Receives a flight progress report from air-ground communications (1.1). Sends a flight plan query to the flight operations database (2.4.7). Receives a flight plan from the flight operations database (2.4.7), containing stored 4D position, next reporting point and fuel information. Checks the current 4D position, next reporting point and fuel information against the stored values. If significantly different, sends a flight progress alert to the flight progress display (3.2). Uses the position report to calculate the ETA at the next waypoint and compares it to the expected time. If significantly different, sends a flight progress alert to the flight progress display (3.2). Creates a flight data message containing the new data, and sends it to the flight operations database (2.4.7) and to the flight progress display (3.2).

Receives an aircraft state data from OOOI reports (2.4.6). If an "In" message is received, the flight is over and all flight progress contracts for that flight are cancelled by sending an appropriate flight progress request message to air-ground communications (1.1).

Requirements Satisfied:
AGP_FPL_090, AGP_CDM_040, AGP_CDM_017, AGP_CDM_050, AGP_CDM_060, AGP_CDM_070, AGP_CDM_080, AGP_CDM_090, AGP_AMG_010, AGP_AMG_021, AGP_AMG_040, AGP_AMG_050, AGP_AMG_060, AGP_AMG_090, AGP_AMG_095, AGP_AMG_100, AGP_AMG_110, AGP_AMG_120, AGP_AMG_130, AGP_AMG_140, AGP_AMG_150, AGP_AMG_160, AGP_AMG_170, AGP_AMG_180, AGP_AMG_190, AGP_AMG_200

4.1.2.4.4 InFlightTrafficManagement

This process is responsible for implementing the ground elements of the In-Flight Traffic Management application. This application allows the AOC operator to initiate a re-negotiation of the aircraft en-route trajectory, the re-negotiation itself being conducted between the pilot and the relevant ATC authority.

This functionality is to be implemented in the phase 2 deliverable.

Process Inputs:
- constraintsAcceptance from AirGroundCommunications (1.1)
- constraintsList from FlightProgressDisplay (3.2)
- trajectoryClearance from AirGroundCommunications (1.1)

Process Outputs:
- flightData to FlightOperationsDB (2.4.7)
- constraintsList to AirGroundCommunications (1.1)
- iftmAlert to FlightProgressDisplay (3.2)
- trajectoryUpdate to CompanyFlightPlanningServer (2.3)

Processing:
Receives a constraints list from the HMI (3). Reads the updated flight plan from the flight plan data store. Sends the constraints list to air-ground communications (1.1). If a constraints acceptance is not received within a pre-defined time after the constraints list was sent to the aircraft, an IFTM alert is sent to the flight progress display (3.2).

Receives a constraints acceptance message from air-ground communications (1.1), containing details of whether the pilot of the flight has accepted or rejected the constraints, and if rejected, possibly details of why the rejection has occurred. Sends an IFTM alert containing the details of the acceptance or rejection to the flight progress display (3.2).

Receives details of the cleared trajectory from air-ground communications (1.1). Sends the cleared trajectory to the flight operations database (2.4.7) and to the company flight planning server (2.4.7). If this message is not received within a pre-defined time of the constraints being accepted, an IFTM alert is sent to the flight progress display (3.2).

Requirements Satisfied:
AGP_CDM_100, AGP_CDM_110, AGP_CDM_120, AGP_CDM_130, AGP_CDM_140, AGP_CDM_150,

4.1.2.4.5 GeneratePostFlightRecord

This process is responsible to generate post flight data.

Process Inputs:
- flightRecord from FlightOperationsDB (2.4.7)

Process Outputs:
- postFlightRecord to PostFlightDataServer (2.5)

Processing:
When a flight is finished, a flight record is sent from the flight operations database (2.4.7) to this process. This data is processed and a corresponding post-flight record is generated and sent to post flight data server (2.5).

Requirements Satisfied:
AGP_AMG_001

4.1.2.4.6 OOOIReports

This process is responsible for dealing with information related to aircraft movements, specifically with OOOI events (Out - leaving gate, Off - taking off, On - touching down, In - arriving at gate).

Process Inputs:
- oooiReport from AirGroundCommunications (1.1)

Process Outputs:
- aircraftState to FlightProgress (2.4.3)
- oooiData to FlightOperationsDB (2.4.7)

Processing:
Receives OOOI report data from air-ground communications (1.1). Then, OOOI data is constructed and sent to flight operations database (2.4.7). If the event that is being reported is an Off or an In, the aircraft state is sent to the flight progress process (2.4.3).

Requirements Satisfied:
AGP_AMG_003

4.1.2.4.7 FlightOperationsDB

This process is responsible for providing a facility to store all aircraft data related to the flight operations. In addition, the process is responsible for providing a facility which can be used (by the HMI) to obtain a best estimate for current position of the aircraft.

This functionality is to be implemented partly in the phase 1 deliverable and partly in the phase 2 deliverable, as noted below.

Phase 1:
Process Inputs:
- AircraftData from PreFlightSupport (2.4.2)
- fourDtrajectoryData from PreFlightSupport (2.4.2)
- flightData from FlightProgress (2.4.3)
- flightDataQuery from HMI (3)
- oooiData from OOOIReports (2.4.6)

Process Outputs:
- flightData to HMI (3)

Processing:
Receives details of which aircraft are assigned to which flights from the pre-flight support process (2.4.2). The aircraft information is stored in the database, associated with the appropriate flight.

Receives trajectory information about specified flights from the pre-flight support process (2.4.2). The trajectory information is stored in the database, associated with the appropriate flight.

Receives an OOOI report from the OOOI reports process (2.4.6). The change in status for the appropriate flight is stored in the database. If the OOOI message is an In message, the flight has ended. All information stored for that flight is used to generate a post flight record by sending it to the appropriate process (2.4.5).

Receives flight data from flight progress (2.4.3). The flight data is stored in the database, associated with the correct flight.

Receives a query for flight data from the maintenance console (3.3). The required data is retrieved from the database and sent as flight data to the maintenance console (3.3).

Receives a query for flight data from flight progress (2.4.3). The required data is retrieved from the database and sent as flight data to flight progress (2.4.3).

Phase 2:
Process Inputs:
- AircraftData from PreFlightSupport (2.4.2)
- fourDtrajectoryData from PreFlightSupport (2.4.2)
- fuelInformation from PreFlightSupport (2.4.2)
- paxBaggageData from PreFlightSupport (2.4.2)
- flightData from FlightProgress (2.4.3) and from InFlightTrafficManagement (2.4.4)
- flightDataQuery from HMI (3)
- oooiData from OOOIReports (2.4.6)

Process Outputs:
- flightRecord to GeneratePostFlightRecord (2.4.5)
- flightData to HMI (3)

Processing:
Receives details about how much fuel has been loaded onto the aircraft, and the passenger and baggage load, from pre-flight support (2.4.2). The fuel, passenger and baggage load information is stored in the database, associated with the appropriate flight.

Receives updated flight data from in-flight traffic management (2.4.4). The flight data is stored in the database, associated with the appropriate flight.

Requirements Satisfied:
AGP_AMG_003, AGP_HMI_040, AGP_HMI_050, AGP_HMI_055

4.1.2.5 PostFlightDataServer

This process will store post flight data and supply it to the flight analysis console (3.5).

Process Inputs:
- postFlightRecord from FlightOperationsDataServer (2.4)
- postFlightDataQuery from FlightAnalysisConsole (3.5)

Process Outputs:
- postFlightData to FlightAnalysisConsole (3.5)

Processing:
Receives a post flight record from flight operations data server (2.4) and stores it.

When flight analysis console (3.5) sends a query regarding post flight data, the process replies with the requested data.

Requirements Satisfied:
AGP_AMG_001

4.1.3 HMI

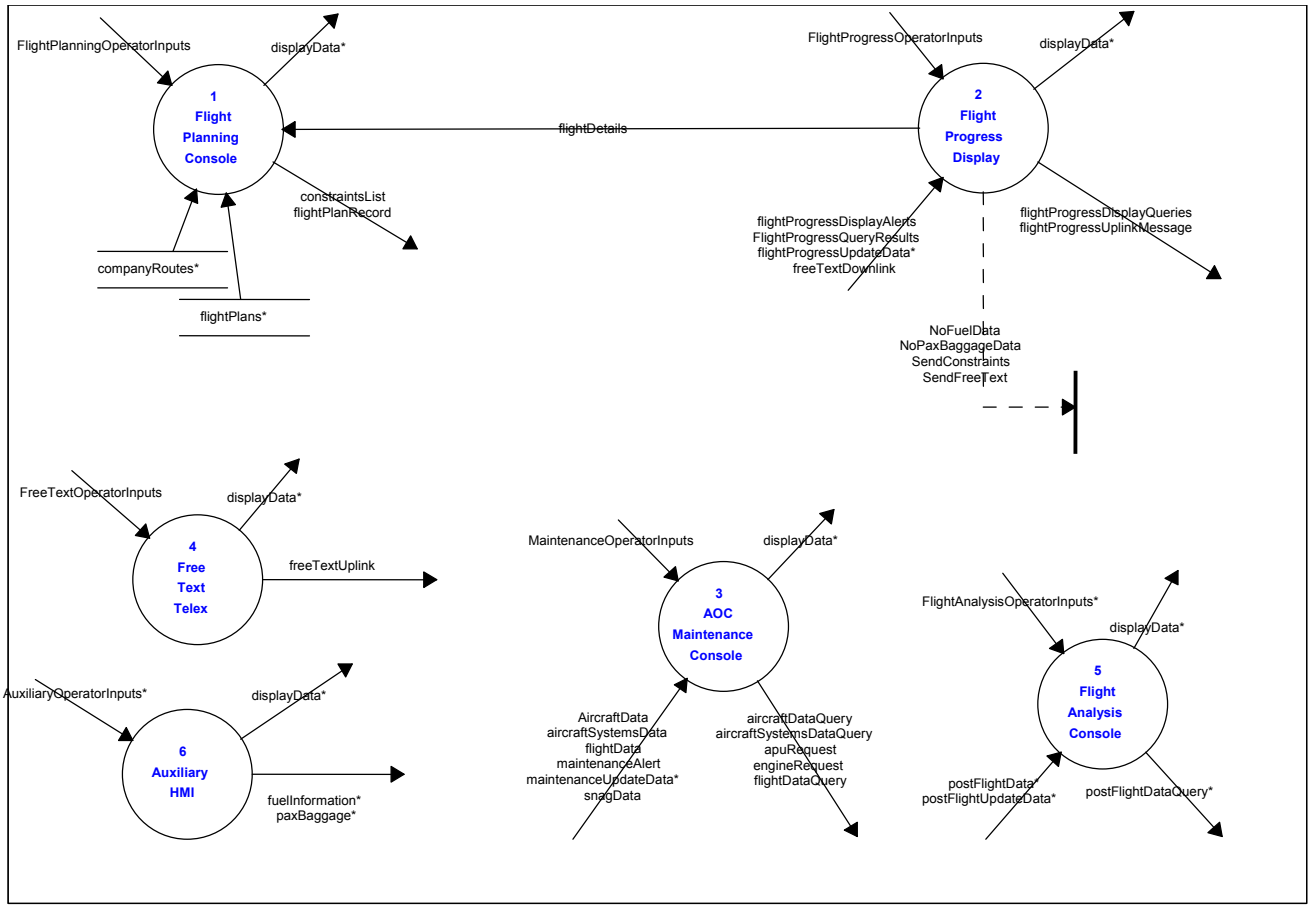

Figure 8 – HMI Process DFD.

Process Inputs:
- alertMessage
- hmiUpdateData
- queryResult
- snagData
- OperatorInputs
- commsAlert
- commsLogData
- freeTextDownlink

Process Outputs:
- displayData
- commsLogQuery
- freeTextUplink
- apuRequest
- constraintsList

- engineRequest
- flightProgressUplinkMessage
- dataQueries
- flightPlanRecord
- fuelInformation
- paxBaggage

4.1.3.1 FlightPlanningConsole

This process is in charge of making the interface between the AGP operator and the AGP.

Process Inputs:
- FlightPlanningOperatorInputs from AOCOperator
- flightDetails from FlightProgressDisplay (3.2)
- companyRoutes from CompanyRoutes
- flightPlans from FlightPlans

Process Outputs:
- constraintsList to FlightOperationsDataServer (2.4)
- flightPlanRecord to CompanyFlightPlanningDataServer (2.3)
- displayData to AOCOperator

Processing:
Receives a request to load a pre-defined route, or a pre-defined flight plan from the operator. The requested information is retrieved from the corresponding physical stores and displayed to the operator.

If requested by the operator, this process sends a flight plan record to be stored in the company flight planning data server (2.3).

Receives the current flight plan of a registered aircraft from the flight progress display (3.2), these details will be dynamically updated according to the flight situation and displayed to the operator.

If a route is being displayed, the operator can select it and request to send it to a registered aircraft as an IFTM constraints list. If he does, the constraints list is sent to the flight operations data server (2.4).

Requirements Satisfied:
AGP_HMI_002, AGP_HMI_003

4.1.3.2 FlightProgressDisplay

This process is intended to enable the monitoring of the company flights progress. The process also provides facilities for the exchange of messages with a selected company aircraft.

Process Inputs:
- FlightProgressOperatorInputs from AOCOperator
- flightProgressDisplayAlerts from AGPFunctions (2)
- FlightProgressQueryResults from FlightOperationsDataServer (2.4)
- flightProgressUpdateData from AGPFunctions (2)
- freeTextDownlink from AirGroundCommunications (1.1)

Process Outputs:
- displayData to AOCOperator
- flightProgressDisplayQueries to AOCOperator
- flightProgressUplinkMessage to AirGroundCommunications (1.1)
- flightDetails to FlightPlanningConsole (3.1)
- NoFuelData to HMI C-Spec (Control Table HMI)
- NoPaxBaggageData to HMI C-Spec (Control Table HMI)
- SendConstraints to HMI C-Spec (Control Table HMI)
- SendFreeText to HMI C-Spec (Control Table HMI)

Processing:
Sends a query to comms server (1.2) to get message log data. Receives the requested data from the comms server (1.2). This data is displayed to the operator.

When the process receives a an alert from air-ground communications (1.1) and/or from fleet simulator comms (1.4), a warning message is constructed to be displayed to the operator.

Receives flight progress update data from the flight operations database (2.4.7). The HMI display is updated accordingly.

Sends flight data queries to flight operations database (2.4.7). When the requested data is received from flight operations database (2.4.7), the HMI is updated.

If the operator requests to visualize weather data on the flight progress display, a query is created and sent to the weather information process (2.2.2). When a reply is received from weather information (2.2.2), the meteorological data is displayed to the operator.

Receives free text downlink message from air-ground communications (1.1) and updates the HMI accordingly.

When the operator chooses to uplink a constraints list to a registered A/C, the process sends flight plan data to the flight planning console (3.1).

Receives alerts from AGP functions (2) and updates the HMI accordingly.

Receives a flight planning alert from company flight planning data server (2.3), and shows an alert message to the AOC Operator.

Receives a meteo report alert from aircraft meteo reports (2.2.1) and updates the HMI accordingly.

Requirements Satisfied:
AGP_COM_015, AGP_FPL_210, AGP_FPL_230, AGP_AMG_010, AGP_AMG_180, AGP_AMG_200, AGP_AMG_240, AGP_HMI_002, AGP_HMI_015, AGP_HMI_020, AGP_HMI_025, AGP_HMI_030, AGP_HMI_040, AGP_HMI_045, AGP_HMI_035, AGP_HMI_050, AGP_HMI_055, AGP_HMI_003

4.1.3.3 AOCMaintenanceConsole

The purpose of this process is to inform the AGP operator about technical events. This task is done by using charts, and other relevant information.

Process Inputs:
- AircraftData from external file
- aircraftSystemsData from HandleAircraftSystemInformation (2.1.2)
- flightData from FlightOperationsDB (2.4.7)
- maintenanceAlert from HandleAircraftSystemInformation (2.1.2)
- maintenanceUpdateData from AGPFunctions
- snagData from HandleSnagData (2.1.1)
- MaintenanceOperatorInputs from AOCOperator

Process Outputs:
- displayData to AOCOperator
- aircraftSystemsDataQuery to HandleAircraftSystemInformation (2.1.2)
- apuRequest to HandleAircraftSystemInformation (2.1.2)
- engineRequest to HandleAircraftSystemInformation (2.1.2)
- flightDataQuery to FlightOperationsDB (2.4.7)

Processing:
On operator input this process queries the handle aircraft system information (2.1.2) about aircraft systems data. Receives the requested data from handle aircraft system information (2.1.2) and displays it in the HMI maintenance display.

The operator can also request the downlink of an APU or engine status report. These requests, after created, are sent to handle aircraft system information (2.1.2).

To complement the information that is being displayed, this process requests flight and aircraft data. Flight data is retrieved from the flight operations database (2.4.7) and aircraft data is retrieved from an external file and shown on the display.

Snag data is received from handle snag data (2.1.1) and the HMI maintenance display is updated accordingly.

Receives maintenance data from the AGPFunctions (2). The HMI maintenance display is updated accordingly.

Requirements Satisfied:
AGP_HMI_001, AGP_HMI_002

4.1.3.4 FreeTextTelex

The free text telex process enables the AOC operator to compose and send a free text message to a selected aircraft.

Process Inputs:
- FreeTextOperatorInputs from AOCOperator

Process Outputs:
- freeTextUplink to airGroundCommunications (1.1) and to FleetSimulatorComms (1.4)
- displayData to AOCOperator

Processing:
Receives text to be sent from the operator together with the choice of sending the message using the GACS confirmed service or the GACS unconfirmed service. Visual feedback is provided as the operator enters data.

Sends the message to air-ground communications (1.1) or fleet simulator comms (1.4).

Requirements Satisfied:
AGP_HMI_002, AGP_HMI_004

4.1.3.5 FlightAnalysisConsole

This tool is to be used when an AGP operator wishes to analyse post flight data. This process will display post flight data when requested by the operator.

Process Inputs:
- postFlightData from PostFlightDataServer (2.5)
- FlightAnalysisOperatorInputs from AOCOperator

Process Outputs:
- postFlightDataQuery to PostFlightDataServer (2.5)
- displayData to AOCOperator

Processing:
Queries post flight data server (2.5) about post flight data. Displays the queried data to the operator.

Requirements Satisfied:
AGP_HMI_005

4.1.3.6 AuxiliaryHMI

This process is used to house several complementary HMIs components like, for instance, input forms to define the amount of boarded fuel and pax/baggage data.

Process Inputs:
- AuxiliaryOperatorInputs from AOCOperator

Process Outputs:
- fuelInformation to FlightOperationsDataServer (2.4)
- paxBaggage to FlightOperationsDataServer (2.4)
- displayData to AOCOperator

Processing:
Receives data from the AOC operator about the amount of boarded fuel and pax/baggage information. Feedback is provided as the operator enters data. This data is sent to flight operations data server (2.4).

Requirements Satisfied:
AGP_FPL_170, AGP_FPL_180

5 ABBREVIATIONS

A/C	Aircraft
AGP	AOC Ground Platform
AOC	Airline Operational Centre
APU	Auxiliary Power Unit
DFD	Data Flow Diagram
ETA	Estimated Time of Arrival
FMS	Flight Management System
FOpsDB	Flight Operations Database
FPD	Flight Progress Display
FPDB	Flight Planning Database
GACS	Generic ATN Communications Services
GAPI	GACS Application Programming Interface
HMI	Human-Machine-Interface
MA-AFAS	More Autonomous Aircraft in the Future ATM System
OOOI	Out, Off, On, In