TITAN

Turnaround **I**ntegration in **T**rajectory **A**nd **N**etwork

Project Number: 233690

TITAN Model Software Design Document

CLASSIFICATION: PU **ISSUE:** 1.0 **DATE:** 13/07/2012

DOCUMENT REFERENCE				
Project	**Work Package**	**Partner**	**Nature**	**Number**
TITAN	02	CRI	DEL	05

Project co-funded by the European Commission and TITAN consortium.

DOCUMENT CONTROL			
Responsible	**Organisation**	**Name**	**Date**
Author	CRI	Andrea Villa/ Irene Navarro	16/04/2012
Partners involved	ISA	Tarja Kettunen	24/04/2012
Reviewer	CRI	Nicolás Suarez	24/04/2012
Approver	INE	Laura Serrano	29/05/2012

DOCUMENT CHANGE LOG			
Issue	**Date**	**Author**	**Affected Sections / Comments**
0.1	16/04/2012	Andrea Villa / Irene Navarro	All / New document
0.2	24/04/2012	Tarja Kettunen	All / New sections included
0.3	07/05/2012	Andrea Villa / Irene Navarro	All / Integrate ISA comments
0.4	29/05/2012	Andrea Villa / Irene Navarro/ Eva Puntero	All / Integrate WP2 partners comments
1.0	13/07/2012	Ana Sáez	Final version after EC Approval

DOCUMENT DISTRIBUTION		
To/Cc	**Organisation**	**Name**
To	EC	Remy Denos
To	INECO	Laura Serrano
To	BLUSKY SERVICES	Steve Zerkowitz
To	CRIDA	Nicolas Suarez
To	CRIDA	Eva Puntero
To	CRIDA	María Ruiz
To	CRIDA	Irene Navarro
To	INECO	Ana C. Sáez
To	INECO	Jose Luis Martin
To	ISA SW	Ian Crook
To	ISA SW	Matthew Bray
To	ISA SW	Zak Tibichte
To	ISA SW	Tarja Kettunen
To	JEPPENSEN	Alicia Grech
To	JEPPENSEN	John Butcher
To	RWTH Aachen University	Sebastian Kellner
To	RWTH Aachen University	Athanasios Katsaros
To	SLOT CONSULTING	Roland Gurály
To	SLOT CONSULTING	Noémi Král
Cc	SESAR JU	Paul Adamson
Cc	AENA/SESAR JU	Alejandro Egido
Cc	AENA/SESAR JU	Francisco Javier Fernández de Liger

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

The present document contains information about how the TITAN Model Software was designed based on the *Single Aircraft Turnaround Model.* It contains a description about how multiple flights are managed by the model and how they interact with each other. It also includes a description of the implementation of TITAN features into the model. Within this Software Design Document, narrative and graphical information of the global software design can be found comprising the class diagram and TITAN model interactions.

1. INTRODUCTION

1.1 Purpose

Purpose of this document is to provide a description of the TITAN model software design. It is based on and aligned with the Single Aircraft Turnaround Model.

For this particular Software Design Document the focus is placed on the interactions between model elements, TITAN interactions and information flow systems.

1.2 Document Structure

This document is divided into two sections:

- The first section is this introduction which presents the purpose, structure, audience, references and abbreviations and acronyms of the document.
- The second section describes the software design of the TITAN model.
- Annex 1 lists the model interaction points that allows the user to interact with the model.

1.3 Intended Audience

This document may be distributed freely within the TITAN consortium, both to those who are involved in the use of the model as well as stakeholders to check the consistency of the model.

1.4 Associated Documentation

[1] TITAN D2.2 Single Aircraft Turnaround Model SW Design Doc, v0.4, March 2012

[2] TITAN_WP2_Interaction points for the Model, v0.3, March 2012

[3] TITAN D1.4 TITAN Operational Concept Document – Issue 1, v1.0, October 2010

[4] TITAN D2.1 Technical Requirements Document, v1.0, July 2011

1.5 Abbreviations and Acronyms

ATC	Air Traffic Control/Controller
AS	Airside
BFIS	Baggage Flow Information Service
EIBT	Estimated In Block Time
EOBT	Estimated Off Block Time
LS	Landside
PFIS	Passenger Flow Information Service
SW	Software
TITAN	Turnaround Integration in Trajectory and Network

2. SOFTWARE DESIGN

2.1 Design Goals

The main objective of the TITAN Model is to provide a flexible software model which could be used to emulate the multiple turnarounds that occur every day at any airport. This software is an extension to the Single Aircraft Model software, where only one aircraft turnaround is simulated. This ability to replicate is one of the Single Aircraft Model software design objectives.

The second objective focuses on including into the Single Aircraft Model additional modelling features to reproduce TITAN interactions and information flow systems. This objective is achieved by including milestones and their monitoring as user decision points as well as information systems of passenger and baggage flows.

In short, the definition of a single aircraft turnaround is "*the sequence of operations that are required from the time point when an aircraft is stationary at the airport with the wheel chocks in place, to the time point when the chocks are removed and the aircraft receives ATC clearance to leave the stand*. All these operations provide a service to the aircraft. Each service is composed of several sub-processes as detailed in TITAN D2.2 Single Aircraft Turnaround Model SW Design Doc [1].

In reality, the turnaround process is made up of a complex set of often interdependent sub-processes, many of which require the use of a limited set of shared resources, and which can vary widely in the time and the way the processes are carried out. A turnaround starts when the first of the sub-processes previously defined for that turnaround starts and it ends when the last process is completed or closed.

The architecture logic of the TITAN Model software is exactly the same as the one of the Single Aircraft Model and data inputs and outputs work in a similar way.

2.2 Data structure

The data structure for the TITAN Model Software is basically the same as the one described for the Single Aircraft model. For a detailed description see [1]. In summary, the global data model structure for both the Single Aircraft model and the TITAN model is divided into two groups: data inputs and data outputs. The first data group is related to traffic data, processes description, turnaround modelling, airport resources and simulation parameters. They are used by the simulation to model a context of turnaround operations at an airport.

Output data are provided automatically by the model and they allow analyzing the model performance.

There are three types of input data which are used for both models.

- Data related to schedule: Schedule data are similar to those of the Single Aircraft model. Schedule, arrival and departure manifest files contain a complete traffic sample for the period to be simulated (e.g. one day). Additionally in the multiple aircraft model, connecting passengers can also be included by detailing arrival and departure flight codes and the number of passengers in connection.
- Turnaround data: As in the Single Aircraft model, services provided to the aircraft can be defined in a highly flexible way by the user via sub-processes definition (e.g. cleaning or baggage unloading) and interconnections between sub-processes can be implemented.

- Data related to the time consumption of entities[1]: Entities consume time by moving from one point to another at the airport (e.g. passengers walk from check-in desk to security control), by waiting in a queue (e.g. boarding gate queue) or by using resources (e.g. time at a passport desk).

  Resources include both fixed resources which are part of the airport infrastructure, such as baggage drops, check in desks, passport desks or gates and mobile resources that can be moved around the airport to support a turnaround, such as baggage load belts, catering trucks, buses etc. The latter also allows the user to define specific parameters for each resource such as quantity, capacity, time to aircraft, time at aircraft, etc.

Moreover, new simulation parameters and data inputs were introduced into the model as part of the multiple turnaround extension:

- The model includes a delay functionality which allows simulating flights arriving late. The user can choose between a fixed delay value and a delay distribution (e.g. normal distribution) as well as the percentage of delayed flights.
- For the TITAN model simulations, there are additional input data elements which aim at providing additional information to the user when the simulation is running (e.g. milestones) or defining special airport resources which are to be used under certain circumstances that are presented to the user in the course of the simulation (e.g. emergency Desk Pools for check-in, baggage drop, security control or passport control). If a process is delayed TITAN model allows the optional use of these special airport resources to help mitigate any related problems.
- Additional scenario settings also allow fixing parameters like gate availability period which is the time period needed for a new gate to be reallocated to an arriving flight if the initially allocated gate is being used by another delayed flight. Ground handling waiting period for a delayed process can also be defined.
- There is also an input data file to define which user interaction dialogs are presented to the user and how they are presented to him.

Output data are provided by means of text files where the different events are registered in chronological order. Data outputs provided for the single turnaround are now replicated for each aircraft turnaround.

2.3 Class diagram

A global class diagram about TITAN model is shown in Figure 1.

A global model class diagram was presented in full detail in [1]. It showed how turnaround sub-processes are defined by using nodes depending on the turnaround type. Different turnaround types may be defined according to flight and airline characteristics and each one can contain different sub-processes, through which entities flow during the simulation.

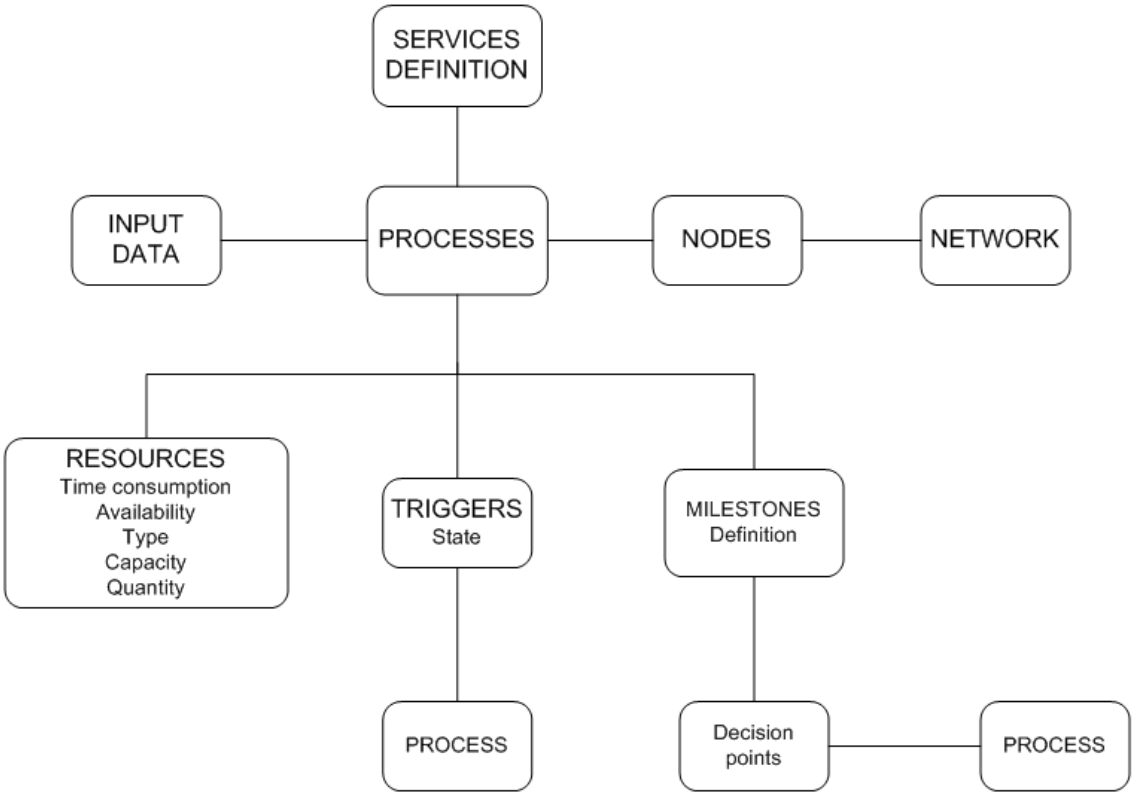
During the simulation, interactions between defined elements occur. This allows simulating real airport operations where multiple aircraft turnaround sub-processes are carried out in parallel.

Firstly, each airport has different resources defined. According to their properties and characteristics resources can be shared among all aircraft. This establishes a relationship between

[1] TITAN model considers following entities: passengers, baggage and operations.

different aircraft turnaround sub-processes (e.g. if all baggage load belts are occupied at a certain time point, the baggage unloading process of a new flight cannot start.)

Secondly, some sub-processes may be carried out in parallel but some may be dependent on the completion of other sub-processes. This situation is modelled by using triggers[2] after having established the relationships between the various sub-processes.



*In the class diagram the element 'process' refers to two TITAN operational concept terms [3]: process and sub-process. According to the Technical requirements document, the conceptual definition of the element 'process' *is the 'serie of actions' towards a particular result, as an example, to load mail/cargo [4]*. Data models of both TITAN models (single and multiple) are based on this definition and refers to process and sub-process operational concept terms using the same data model element: 'process'.

Figure 1: TITAN Global Class Diagram

Finally, TITAN services are implemented by using milestones[3]. The model uses a default set of milestones, based on the list and definitions given in the internal document [2]. This list includes TITAN operational concept milestones and some others that were considered needed for the proper modelling performance. The model's default milestones can be modified by the user to a certain degree. It has been proved necessary to define specific conditions for milestone accomplishment/completion in certain processes. When a milestone is missed or is going to be

[2] Triggers can be the start/end of other sub-processes that launch the start of a sub-process. They are used to define the turnaround sequence of sub-processes [1].

MilestoneTriggers can be the start/end of a sub-process that determines the milestone achievement.

[3] Milestones are notifications displayed when a process starts or has been completed [1]

missed a user interaction dialog appears where a set of possible decisions[4] defined to be applicable are presented to the user. The decision made by the user aims at minimising delay and can affect the process, the entity flow or a resource utilisation. The user can decide by using TITAN Model Information Services.

2.4 TITAN Model interactions

Both Single Aircraft Turnaround Model and TITAN model are designed to model the service provision to aircraft. According to different aircraft characteristics and airline operational concepts, different sub-processes can be defined for each turnaround.

As detailed before, a turnaround starts when the first of the sub-processes defined for that turnaround starts and it ends when the last sub-process is completed or closed. It is important to notice that the start of a turnaround process may not coincide with in block time but with some landside sub-processes such as check-in or security process.

2.4.1 Start of sub-processes

The start of a sub-process is related to one or more of the following features:

2.4.1.1 Triggers

TITAN model implements two types of triggers; process_start or process_complete. When a trigger is defined it is necessary to choose what type of trigger is being implemented and which the source sub-process is. For example, boarding could have a process_complete trigger related to refueling. In this case, refueling is the source process. That means that boarding is not allowed to start before refueling is completed.

2.4.1.2 Sub-Processes start time

The start time of a sub-process can be defined as a sub-process parameter. TITAN model allows defining the start time of a sub-process as an offset from an updated or initial event. For example, check-in process start time can be defined as an offset of two hours from the initial EOBT.

2.4.1.3 Resources availability

Each sub-process requires several resources for carrying out its actions. If the resource is available, the sub-process acquires it and entities can flow through the sub-process. If the resource is not available, the sub-process waits until the resource is made available. This results in an increase in the sub-process execution time. As detailed in [1] there are two different resource types: fixed resources and mobile resources. Fixed resources are related to a fixed location at the airport (check-in desks, baggage drops, etc) while mobile resources move from one location to another and time parameters such as time to aircraft or time at aircraft can be defined for them. The resource utilisation time impacts sub-process duration.

[4] Decision points or interaction points associated to milestones are control points that display a user interaction dialog when the milestone cannot be achieved at the planned time. They allow interacting with the model in order to decide which sub-processes will continue and how by choosing one action of a predefined list of actions that the model provides to the user [1].

Most of the resources are shared between all aircraft but some of them can be allocated to a specific airline such as baggage drops. Resources are gathered in a pool and the model chooses one randomly from the suitable pool each time it is required.
Sub-processes can use resources in different ways:

- *One time* resource strategy means that the sub-process uses and reserves the resource for the whole sub-process duration. This is, for example, used to model a boarding agent. The boarding agent stands at the boarding gate from the time the boarding gate is open up to its closing time even if there are occasionally no passengers boarding.
- *Shared* resource strategy. Resources have an associated utilisation time and once an entity has finished using a resource it is made available for use by another entity in the same or another sub-process. For example, a PRM passenger may use an ambulift for boarding but, once the passenger reaches the aircraft, the ambulift is free for use by another sub-process.
- *Acquire and Release* resource strategy. Resources can be acquired at one time point of a sub-process and be released later on in another sub-process. For example, a PRM passenger may need an assistant from check-in desk to boarding gate.

Interactions between aircrafts only involve the availability of resources, especially the availability of the allocated gate.

2.4.1.4 Interaction Points

The TITAN model has been designed to use a set of predefined interaction points to issue a user interaction dialog when the turnaround process of the flight is not proceeding according to initial planning. Interaction points are input data for the model where the user has certain flexibility as to which interaction points are active and which are the possible solutions presented to him for decision-making. When the interaction dialog appears the simulation is paused until the user chooses one solution to the particular critical situation. The user has access to background information to justify his/her decision-making, namely the Passenger Flow Information Service (PFIS) and the Baggage Flow Information Service (BFIS). The decision made can modify the course of the simulation but it may also have no impact.

Implemented interaction points into TITAN model can be classified as:

- Sub-process is late to start: This interaction point will be displayed if a given sub-process is not starting on time. It can appear due to lack of resources or because there is a trigger to a previous unfinished sub-process. Several options can be presented to the user in the interaction window:
    - Prioritize the use of resources.
    - Start the sub-process immediately.
    - Wait for a maximum of x minutes after which the dialog is shown again.
    - Do nothing (wait for sub-process to start)
- Ground handling is late to start: If ground handling cannot start as scheduled due to delay in the first ground handling sub-process attributed to lack of resources, the TITAN interaction window will be presented with the following choices:
    - Prioritize the use of resources
    - Start a different ground handling sub-process

- Cancel sub-process
- Wait for a maximum of x minutes after which the dialog is shown again
- Do nothing

A table showing all default interaction points of the model is provided in Annex 1.

2.4.2 End of sub-processes

The end of a sub-process can be determined through different features.

2.4.2.1 Defined closing time

Sub-process definition offers several options for closing the sub-process. Each sub-process may have a pre-defined duration after which the sub-process is closed. Closing time can also be defined as an offset from a specific event (e.g. check in process can be closed an offset of 30 minutes from EOBT). If the last entity passes successfully through the sub-process before the estimated closing time the user may also choose to close the sub-process in advance.

If completion time is close and not all entities have gone through the sub-process nodes, the sub-process can either be closed immediately or the user can choose to wait for a maximum of x minutes.

2.4.2.2 Time extension in a decision point

Decisions made at TITAN interaction points can affect sub-process closing time. An unfinished sub-process can be closed or extended by the user at a particular interaction point. Prioritizing resource allocation or modifying entity flow can result in an optimization of the global process duration.

2.4.2.3 Interaction Points

Implemented interaction points related to end of sub-processes can be classified as:

- Sub-process is late to finish: If a sub-process is not completed by its defined closing time an interaction dialog will appear showing this information. At this interaction point the user can:
  - Close the sub-process immediately
  - Wait for a maximum of x minutes
  - Wait for the sub-process to be completed
  - Do nothing (wait or not according to default sub-process settings)
  - Open a new desk (to remain open for x at most)
  - Search for entities (passengers) to move (e.g. to the front of the queue)
- EIBT is going to be missed: The gate allocated to an arriving aircraft may not always be available due to a delay of the previous aircraft turnaround process at the given gate. In this case an interaction dialog appears and it is possible to choose one of the following options:
  - Wait for the gate to be made available

- Choose a random gate *
- Choose an alternate gate manually *

(* from a subset of gates estimated to be available for the next x minutes)

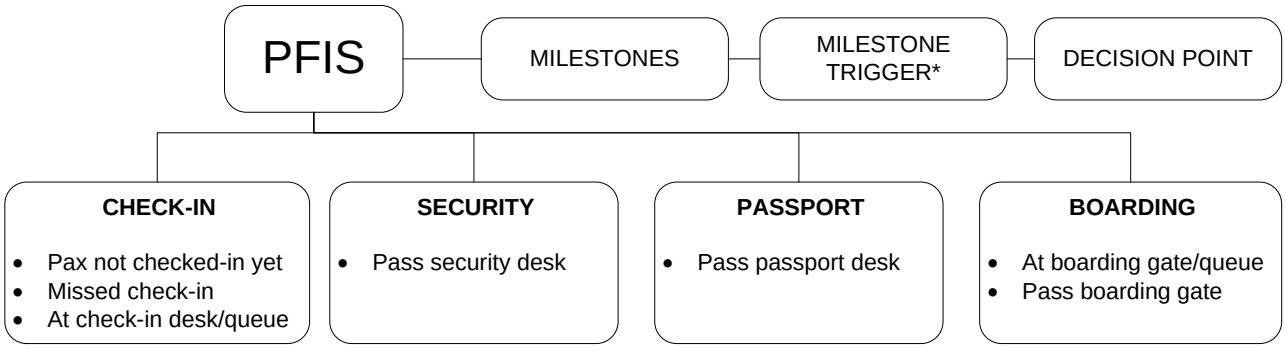
- EOBT is going to be missed: When the scheduled EOBT is going to be missed due to sub-processes still running, an alert is issued to the user in the form of an interaction dialog with the following options:
  - Do nothing (wait for the sub-processes to be completed)
  - Prioritize the use of resources
  - Leave immediately (all sub-processes are closed)
  - Close sub-process(es) from list

A table showing all default interaction points of the model is provided in Annex 1.

2.5 TITAN Model Information Services

2.5.1 Passenger Flow Information Service (PFIS)

Figure 2 shows a global structure of the PFIS.



*Milestone Trigger can be the start/end of a sub-process that determines the milestone achievement. TITAN model uses two different trigger methods: 'running' for the start of the sub-process and 'completed' for the end of sub-process (see section 2.5.4)

Figure 2: PFIS Diagram

PFIS has several monitoring points where the model obtain information about the turnaround performance at different sub-processes (check-in, security, passport and boarding) based on defined milestones according to TITAN Operational concept [3]. Each Milestone is associated to a milestone trigger that determines how it is achieved or missed. In case the milestone is missed, an interaction dialog may appear setting up the corresponding decision point. PFIS information is available at any moment during the simulation both during TITAN and Non-TITAN simulations[5] and also when an interaction dialog appears.

Figure 3 **¡Error! No se encuentra el origen de la referencia.**shows an example of the information provided by PFIS at an interaction point. In this example an interaction window appears as

[5] PFIS information is available but has not impact in the turnaround operation as user can not do anything with it

boarding process is going to finish late because at the estimated closing time 10 passengers have not checked-in yet. PFIS can be used to identify the location of all missing passengers. In this case, five passengers are located at the check-in desk or queue. PFIS also shows which check-in desks are being used.

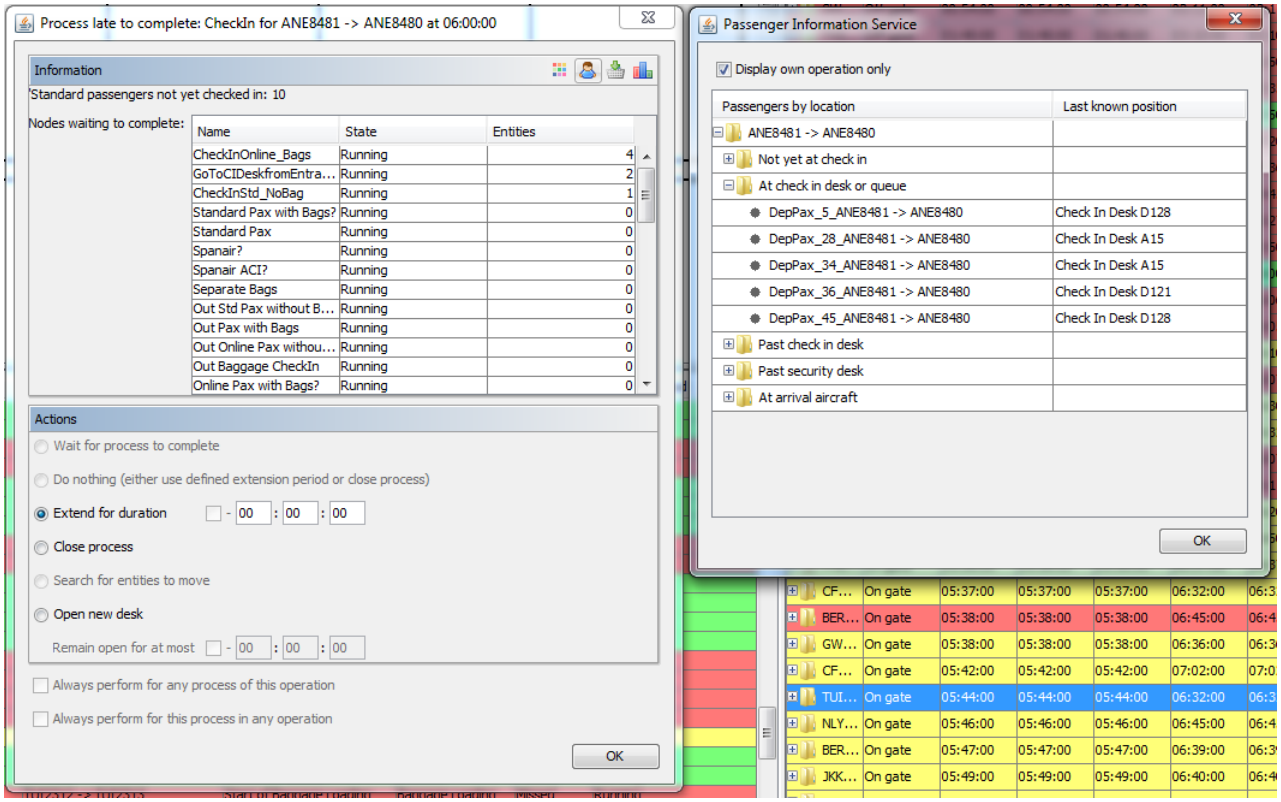


Figure 3: PFIS Interface

2.5.2 Baggage Flow Information Service (BFIS)

Figure 4 shows a global structure of the BFIS.

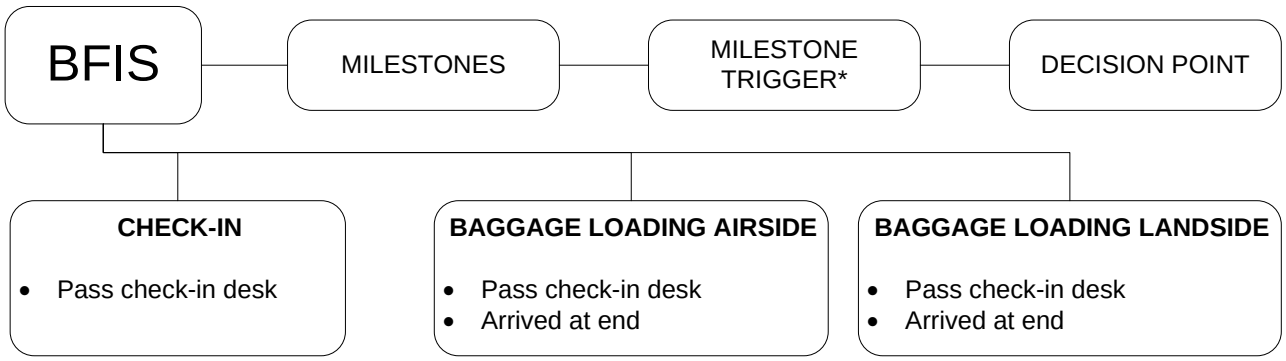


Figure 4: BFIS Diagram

BFIS is also available to the user as a separate window in the user interface and on interaction dialog boxes. BFIS definition is similar to PFIS and provides information on check-in process and baggage loading and unloading.

Figure 5 shows the information provided by BFIS when an interaction dialog during a loading process appears. In this case, baggage loading airside is going to start late. BFIS window allows the user to know where all baggage for that flight is.

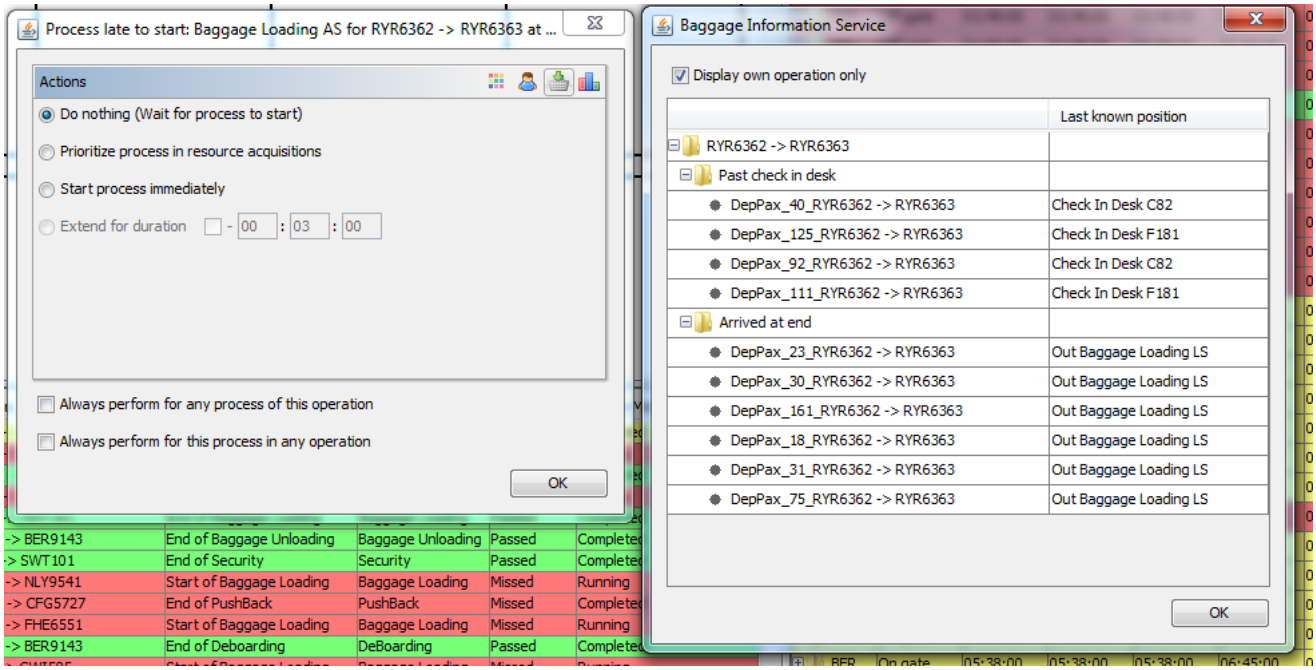


Figure 5: BFIS Interface

2.5.3 Resource utilisation information

Resource utilisation information into TITAN Model is always available in a separate window in the user interface and can also be shown on interaction dialog boxes. This way, the user can search for resource usage and status information before taking a decision to continue the simulation. The resource information window appears and the simulation is paused until the user closes it.

Information shown for each resource, both airport mobile and fix resources, includes the number of sub-processes or entities using it, the current queue size, the maximum queue size, minimum and maximum waiting time and average waiting time (average weight).

Figure 6 shows information related to check-in desks.

Name	Currently Used By	Current Queue Size	Maximum Queue Size	Minimum Wait	Maximum Wait	Average Weight
Check In Desk F183	0	0	0	00:00:00	00:00:00	00:00:00
Check In Desk F184	0	0	0	00:00:00	00:00:00	00:00:00
Check In Desk F185	0	0	57	00:00:00	01:03:13	00:29:08
Check In Desk F186	0	0	0	00:00:00	00:00:00	00:00:00
Check In Desk F187	0	0	0	00:00:00	00:00:00	00:00:00
Check In Desk F188	0	0	57	00:00:21	01:19:09	00:25:33
Check In Desk F189	0	0	3	00:01:14	00:09:50	00:05:26
Check In Desk F190	0	0	0	00:00:00	00:00:00	00:00:00
Check In Desk F191	1	23	31	00:02:55	00:56:41	00:29:39
Check In Desk F192	0	0	46	00:00:15	01:20:03	00:24:31
Check In Desk F193	0	0	4	00:01:29	00:11:24	00:04:49
Check In Desk F194	0	0	3	00:01:20	00:08:43	00:05:16
Check In Desk F195	1	13	36	00:01:27	01:14:26	00:31:41
Check In Desk F196	0	0	0	00:00:00	00:00:00	00:00:00
Check In Desk F197	0	0	10	00:00:38	00:27:20	00:12:12
Check In Desk F198	0	0	18	00:03:21	00:49:24	00:34:01
Check In Desk F199	1	4	51	00:01:21	01:18:00	00:31:01
Check In Desk F200	0	0	10	00:00:28	00:25:11	00:11:12

Figure 6: Resource information

2.5.4 Milestone status information

Milestone information is always available while the simulation is running in the model interface. The user can pre-define the milestones that will be shown as an input file by detailing the turnaround process, the sub-processes, the milestone names and the trigger method.

Figure 7Figure 7: Milestones information shows milestone information during a simulation. The first two columns show simulation time and operation. For each sub-process several milestones can be defined (e.g. one for the start of a sub-process and one for the end). Milestone Name, Target, and Trigger Method columns include relevant information about the milestone definition:

- The name of the milestone
- The turnaround sub-process linked to the milestone.
- The trigger method that shows how the milestone is defined. It may be ‘completed’ when the milestone refers to the end of a sub-process (it indicates that the milestone will be achieved when the target sub-process is completed) or ‘running’ when the milestone refers to the start of the target sub-process.

State column shows if the milestone was completed on time (in this case the row is shown in green), if there is a risk of missing the milestone (in yellow) or if the milestone was missed which means that the sub-process did not start or finish on time (red lines).

Time	Operation	Milestone Name	Target	State	Trigger Method
05:21:39	BER9872 -> BER9143	End of Security	Security	Passed	Completed
05:21:00	GMI6244 -> GMI6245	Start of Deboarding	DeBoarding	Passed	Running
05:20:00	BER3328 -> BER3330	End of CheckIn	CheckIn	Incomplete	Completed
05:20:00	CFG5398 -> CFG5011	Start of Baggage Loading	Baggage Loading	Missed	Running
05:20:00	TCX5016 -> TCX5017	Start of Baggage Loading	Baggage Loading	Missed	Running
05:20:00	SWT1273 -> SWT1274	Aircraft In Blocks	InBlocks	Passed	Running
05:20:00	RYR6362 -> RYR6363	Start of Deboarding	DeBoarding	Passed	Running
05:18:50	NLY9540 -> NLY9447	End of Baggage Unloading	Baggage Unloading	Passed	Completed
05:18:00	GMI6244 -> GMI6245	Aircraft In Blocks	InBlocks	Passed	Running
05:17:03	IBE2274 -> IBE3079	End of Baggage Unloading	Baggage Unloading	Passed	Completed
05:17:00	BER9872 -> BER9143	End of CheckIn	CheckIn	Missed	Completed
05:17:00	RYR6362 -> RYR6363	Aircraft In Blocks	InBlocks	Passed	Running
05:16:08	NLY9540 -> NLY9447	End of Deboarding	DeBoarding	Passed	Completed
05:16:00	TRA1293 -> TRA1294	Start of Deboarding	DeBoarding	Passed	Running
05:15:30	EZS1511 -> EZS1058	Start of Baggage Loading	Baggage Loading	Missed	Running
05:15:00	TUI2152 -> TUI2153	Start of Baggage Loading	Baggage Loading	Missed	Running
05:15:00	TUI2128 -> TUI2123	Start of Baggage Loading	Baggage Loading	Missed	Running
05:15:00	BER9738 -> NLY3403	Start of Deboarding	DeBoarding	Passed	Running
05:15:00	BER3328 -> BER3330	Start of Refueling	Refueling	Passed	Running

Figure 7: Milestones information

ANNEX 1: MODEL INTERACTION POINTS

Milestone ID	Sub-process	Type of Interaction Point	Description	Interaction Point Trigger
M7	In block	Process late to start	EIBT is going to be missed	Original gate not available
M8	Baggage Unloading Baggage Loading Catering Cleaning De-icing Power Supply Refuelling	Process late to start	Start of Ground Handling is going to be missed	Ground handling cannot start due to lack of resources for 1st process
M11A	Boarding	Process late to start	Start of Boarding is going to be missed	Process cannot start due to lack of resources
M11B	Boarding	Process late to complete	End of Boarding is going to be missed	Process not completed at scheduled time
M15	Off block	Process late to complete	EOBT is going to be missed	Processes not complete at scheduled time
M17	Check-In	Process late to complete	End of Check-In is going to be missed	Process not complete at scheduled time
M18	Security	Process late to complete	End of Security Control is going to be missed	Process not complete at scheduled time
M19	Passport	Process late to complete	End of Passport Control is going to be missed	Process not complete at scheduled time
M20	De-boarding	Process late to start	Start of De-boarding is going to be missed	Process cannot start due to lack of resources
M21	Baggage loading LS	Process late to complete	End of Baggage Loading is going to be missed	Process not complete at scheduled time
M22	Baggage Unloading	Process late to start	Start of Baggage Unloading is going to be missed	Process cannot start due to lack of resources
M23	Baggage loading AS	Process late to start	Start of Baggage Loading AS is going to be missed	Process cannot start due to lack of resources
M24	Refuelling	Process late to start	Start of Refuelling is going to be missed	Process cannot start due to lack of resources
M25	Pushback	Process late to start	Start of Pushback is going to be missed	Process cannot start due to lack of resources
M26	De-icing	Process late to start	Start of De-icing is going to be missed	Process cannot start due to lack of resources

***** TITAN Operational concept [3] considers the sub-process Baggage Loading; however the model has divided it into two sub-processes (Baggage Loading LS and Baggage Loading AS) for modelling purposes.

Table 1: TITAN Interaction Points