A discrete event modelling and simulation environment for applications in the TAM context

P. Förster
Institute of Flight Guidance, German Aerospace Center (DLR)
Lilienthalplatz 7, 38108 Braunschweig, Germany

Abstract

On basis of the Airport Collaborative Decision Making A-CDM the concept of the Total Airport Management TAM was developed by the DLR in the year 2006. As part of the business trajectory, the turnaround process, which connects the land- and airside, delivers the AOBT which is a substantial parameter of the A-CDM and TAM concept. In the view of A-CDM this parameter serves as a milestone and constitutes the time at which the aircraft is prepared and ready for the next flight.
Due to the discrete event character of a lot of systems at the airport a discrete event modelling and simulation environment was developed. The modelling and simulation environment discussed in this paper is intended to meet the demands of a decision support tool in the TAM context. One application of the environment is a turnaround model which allows a better estimation of the AOBT. It also permits a what-if probing which is a crucial element of TAM concept. A further purpose of the environment is to improve the understanding of the system. The model based design methodology that can be applied to the presented environment enables the user to perform structural changes of the simulation model at different levels of detail. The presented environment was integrated in the TAM test bed facility at the DLR Braunschweig. The characteristics of the environment, such as compositional and hierarchical modelling will be illustrated with the example turnaround. The examples deicing and taxi movement are intended to demonstrate the modular character of domain specific function blocks which can be used freely within the environment.

1. INTRODUCTION

The concept of A-CDM developed by Eurocontrol is founded mainly on information sharing between the different stakeholders at an airport. [1]
The information sharing is implemented by defined milestones. The goal is to achieve a common situation awareness to compensate disruptions in the operation sequence at the airport more efficiently and to benefit from better resource utilization.

The turnaround process or TA represents the link between the arrival and the departure part of a business trajectory, thus being a crucial part of the A-CDM concept. The most important measure of the TA is the Actual Off Block Time AOBT. At this time all sub-processes are fully completed and the aircraft is ready to move on. Another milestone of the concept is the corresponding target time, Target Off Block Time, TOBT. The TOBT has influence upon the stand management (belated arriving aircrafts are loosing their slot), or the collaborative predeparture sequence. It also affects the slot allocation by the CFMU. A better estimation of the AOBT enables the stakeholders to make a better use of their resources. For example the predeparture sequence and the slot allocation can be optimized by the controller or respectively the CFMU.
Airlines and ground handlers agree upon a TOBT at which the turnaround process has to be completed. This target time is passed on to the airport.
Because of the exact knowledge of the TOBT, all involved partners are able to coordinate the processes at the airport better.

The TAM concept developed by the DLR is based on A-CDM.This concept aims to improve the performance of an airport and is intended to accelerate the recovery time in case of disruptions. This is done mainly by extending the temporal scope of A-CDM and by integrating the landside [2].
The two key elements of TAM are a facility which depicts and analyses the actual situation at the airport (Airport Operations Centre APOC) and a planning tool which allows all participating stakeholders to discuss and decide the next steps depending on the results of a planning cycle (Airport Operations Plan AOP). The planning mechanism itself considers the flight plan and incorporates a complete simulation of the airport as well as tactical planning tools such as arrival or departure managers. It optimizes the plan in consideration of common agreed performance goals like punctuality or throughput. The result of a planning cycle serves as a foundation for the negotiation process of all partners. A crucial point of the TAM concept is the what-if ability. Every change in the resource allocation or target times initiates a new planning cycle. So, the what if probing illustrates the stakeholder the implications of his alteration to the airport system. It allows all partners to work out several different scenarios. This can be done in collaboration with others or by their own.

The modelling environment described in this paper is intended to simulate the complete turnaround process at a generic environment of the APOC. The results of the simulation like an AOBT or EOBT in case of executing a forecast are integrated in the system of the APOC. For example: “equipped” with a calculated TOBT, a departure manager starts a planning cycle to optimize the departure sequence.

The discussed modelling environment also supports the stakeholders in their decision making by calculating the effects in case of a what if probing. Furthermore it aims to help the user to gain a better understanding of the system and to raise his system-awareness.

2. ASPECTS OF THE ENVIRONMENT

Many activities at the airport show a discrete event character. As an example the TA can be viewed as a discrete system where processes start and end at distinct times. Between these timestamps, the system remains in the same state. A discrete event point of view seems reasonable when modelling complex systems in this domain.
The discussed modelling environment comprises three parts, the modelling part, the discrete event simulation and the what if probing.

The model based design uses executable models to test if the system requirements of the tackled subject can be met with the current design. This approach allows the designer to focus on the crucial aspects of a problem rather then being bothered with the implementations details. It is an efficient method of dealing with the increasing complexity of a system. To achieve a manageable conceptual design, the top down approach is a common and accepted method. It allows achieving the necessary level of detail by successive refinement of the different sub-processes or components. The bottom up approach permits the integration of knowledge of domain experts. This is done by developing sub components that are combined later into a whole system. The user of the presented modelling environment should be enabled to use both approaches.
To allow an easy handling and interaction when building a model the user can work with the drag&drop method of blocks that feature specific functionalities. All blocks can be combined freely within the model. To allow a user friendly developing surface und to take advantage of the high level language MATLAB®, the environment was created on a transaction oriented simulation language (SIMULINK®/SIMEVENTS®) [3] as part of the MATLAB suite. The basic approach was published first in 2009 [4].

The modelled design can now be tested with a simulation. In the TAM context the simulation of a TA model serves as an emulation of a part of the generic airport environment in the APOC. Here, all processes and sub-processes of the turnaround as well as dependencies of internal and external inputs, probabilities of failures and corresponding outage times or available resources have to be represented. The duration of sub-processes can be stochastic or resource-dependent. The simulation speed of the TA model is synchronised with the generic airport simulation which encompasses airborne- and ground traffic.

The third part of the environment represents the what if probing. A parallel fast time simulation calculates the effects made by the user, thus enabling him to evaluate the impact of his changes to the system. This is done by expanding the real-time simulation of the turnaround process with a second simulation mode which allows a forecast of the current situation.
The following chapter deals with some of the functional properties of the environment in case of the turnaround process.

2.1. Modelling functionalities of the environment

This chapter is intended to picture some of the modelling functionalities with the help of the TA.
The turnaround process can be described as a network model. The particular sub-processes like boarding or fuelling are connected by serial and parallel interdependencies. The duration of a sub-process can result in the aircraft type, the aircraft carrier or the destination, but there can be a lot of other dependencies which are capable of affecting the duration of a sub-process, like failures or resource shortages.
This chapter will illustrate how a TA could be modelled and how changes could be applied to a design by a user. In this case the user respectively the stakeholder at the APOC would be a ground handler.

The modelling part of the environment can be divided in three different user levels. The “first user level”, shown in figure 1 is to be used by the stakeholder. The turnaround process can be modelled and configurations of parameters can be made at this level.
Figure 1 shows a section of a generic net model of a turnaround process. The coloured blocks visible at this level are called *working blocks*. Within the environment exist several different working blocks. The blue blocks, like fuelling, represent processes or sub-processes. These blocks allow the configuration of the duration of the particular process. The beginning of a process or sub-process can depend on extrinsic factors. For example, the “boarding” sub-process (red outlined) is constraint by the Scheduled Off Block Time SOBT, which is given by the flight plan of the aircraft carrier. In case of a remote boarding, the beginning of the sub-process is not dependent on the predecessor (see the block “boarding_remote”).

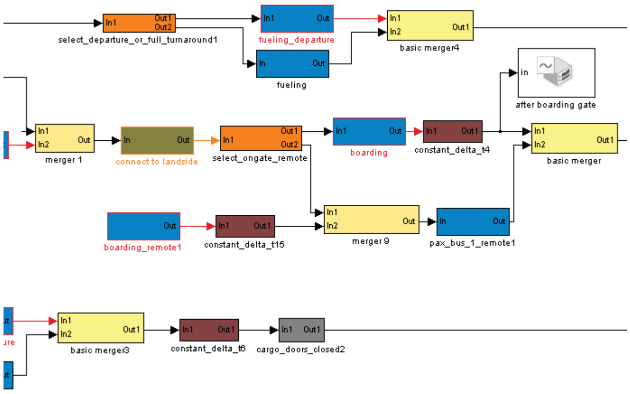

Figure 1 detail of a TA model

A block which selects the appropriate path of boarding at the gate or on a remote position has to be implemented and configured. Such block (orange block “select_ongate_remote” in figure 1) can also be used in case of a structural manipulation if the user decides to collate two similar nets. In case of increasing nets or complexity this is a reasonable design to keep a clear view of the model and to speed up simulation. For example: so called “night stopper” flights arrive at an airport late in the evening, yet their scheduled departure time is set for the next day. Some sub-processes however will be completed immediately after arrival, so that some of the sub-processes have to be carried out at the next day. One example is the “fuelling_departure” and the “fuelling” block. The “fuelling_departure” block belongs to the night stopper net and so this activity has to be delayed. The beginning of this sub-process is defined by the SOBT. Here again it shows a dependency on an extern measure. Later, the two paths are merged in a "basic merger" block (bright yellow). In contrast to this block the "merger" block (yellow) allows the progression of the aircraft only if all inputs of the block are enabled, meaning all previous activities have to be completed for this aircraft.

Because of several stakeholders acting in the APOC, the modelling environment should be integrated at a number of working positions. The integration in the data network of the APOC is an important property of the environment. For the user acceptance it is crucial to have an uncomplicated appliance. Due to the necessity of running on different workstations, the data processing of the modelling environment is based on a MYSQL® standard. The configuration of the parameters of the models is done via a graphical user interface or gui. On behalf of a user friendly interface aspects of usability as consistency, transparency, flexibility and fault tolerance were considered.
To configure the duration of a process or sub-process, an instruction has to be performed at the database. Figure 2 serves as an example of the gui of a working block that represents the stochastic modelling of the duration of a sub-process. The main menu and the submenus responsible for the configuration of the duration of the sub-process are displayed.

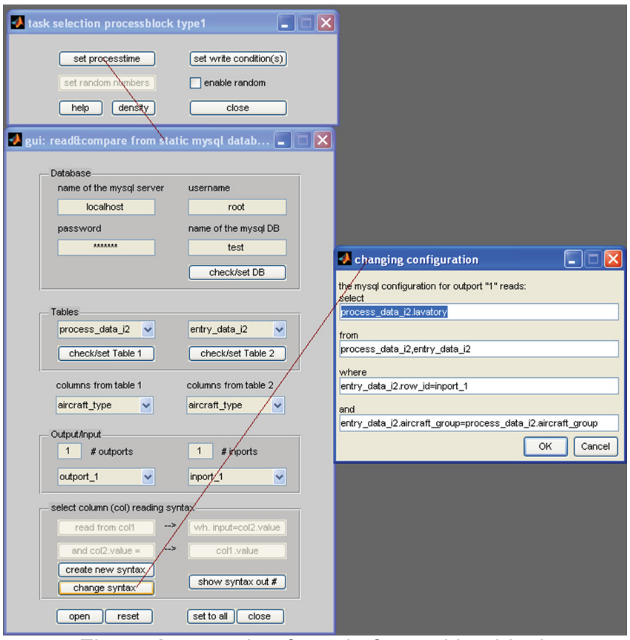

Figure 2 example of a gui of a working block

On the right hand side of figure 2 an example of a MYSQL query is pictured. In this case it is a simple matching between two tables to evaluate the duration of the sub-process "lavatory" which is (not visible) part of the net model shown in figure 1. The duration of the sub-process depends on the aircraft type. This value is stored in the table "process_data_i2". The aircraft type of the current aircraft is stored in the table "entry_data_i2". The user is able to apply changes to the MYSQL query in an easy way by using the edit options in the gui as well as to check the statement by the database. The upper part on the left hand side of the second picture of figure 2 allows the configuration of the database. The available tables in the current database are displayed in the popup menu in "Tables". Analogue to that the available columns of each table are displayed in the next popup menus.
To represent stochastic influences on the TA, the menu of the working block in figure 2 allows a configuration which covers of stochastic measures. For example the duration of a sub-process can be modelled stochastically. Furthermore the occurrence of a failure can be a stochastic variable, as well as the duration that is needed to recover from this exceptional event. Thus to cover disruptions, failure probabilities and downtimes can be configured by density functions. The duration of an abnormal situation or the possibility of the occurrence of such events can be described and configured with the help of several continuous and discrete density functions (for example normal, uniform, weibull, gamma, exponential, triangle). These density functions are provided by the SIMEVENTS environment. The combination of elements that are part of SIMEVENTS and functional blocks provided by the discussed environment will be illustrated in the next user level.
The duration of a sub-process can also be modelled depending on the availability of a resource. It is possible to define any resource, for instance a working unit like a fuel truck or a cleaning team. With a quantity of work defined for the particular sub-process and an amount of working units assigned to that sub-process, the working time to complete this task equals to the coefficient of quantity of work and working units. Thus the duration of the sub-process depends on the amount of resources. For every defined resource the duration for preparation and post processing can be configured. A priority of a process can also be assigned by the user. So in case of insufficient resources or a low priority, relative to other sub-process which are sharing the same resource, the process is delayed. Resources and for this reason working units can be modelled generically.

The following paragraph deals with some of the functionalities of the environment to cover different stochastic modelling requirements.
During the turnaround it is possible that disrupting events occur besides the critical path. These events do not affect the usual sub-processes but they can lead to massive delay. As shown by Wu [5] these disruptive events like damage of the aircraft or aircraft change can be modelled stochastically. To model the TA, a Monte Carlo simulation of a Markov chain of the turnaround model with stationary transient probabilities of the single states was implemented in [5]. To represent abnormal situations Wu defined an occurrence epoch of the disrupting events, counting at the start of the TA. Within this epoch a disruptive event occurs with a distinct probability at a distinct time. The disruption lasts for a certain time, defined also by a stochastic variable. In case of the TA model shown in figure 1 such an event could be modelled by adding a working block that is connected with a parallel path to the network. This will be explained afterwards.
The aircrafts in user level 1 pass through the model from the source to the sink as a scalar value (due to the fact of showing a part of the TA model in figure 1 neither source or sink are visible). An aircraft follows the serial and parallel structure of the created design. Inside each working block the aircraft works as a so called *entity* which possesses different attributes. These attributes could define the time span which the aircraft has to spend in this particular sub-process or the probability of a failure or an assigned resource. The above mentioned modelling of a disruptive event could be managed by positioning an

accordant working block in a parallel path. As soon as the aircraft enters the model a density function block placed inside this working block would create a random number for the occurrence time of the event. In case of an occurrence (modelled by a different block generating random numbers), the entity would be assigned with a duration that represents the downtime (modelled with a third block generating random numbers). Occurrence time, occurrence possibility and duration are then attributes which will be assigned to the entity and will be processed in the working block. As told before, these blocks are placed inside a working block which leads us to the “second user level”.
The second user level enables the user to access the underlying plane of level 1 and thereby allows him to design a working block. He is put in the position to apply changes to the structure or create a new type of a working block that meets the required demands. Nevertheless a basic knowledge of the concepts of SIMULINK and SIMEVENTS is necessary.
Figure 3 shows the structure of a working block which connects the turnaround to an extrinsic simulation of the landside in the configuration of a what-if probing (brown dyed block witch orange outlining in figure 1). The displayed configuration of this working block computes the time the simulation has to be prolonged if a delay at the landside occurs, that is passengers are late.

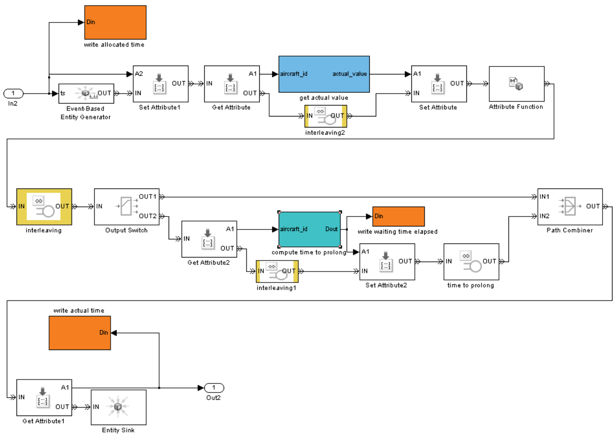

Figure 3 insight of a working block on user level 2

The progress of the simulation depends on the number of passengers appearing at the gate. Simply spoken if a specific number of passengers is not achieved at the gate at the beginning of the boarding, the boarding process is delayed by a given length. In this example the TA model computes a new, later AOBT which itself triggers the passengers simulation of the landside. Hence the forecast of the situation can illustrate arising problems to the stakeholder. The coloured blocks (except the semi yellow ones) in figure 3 are *function blocks*. They are part of the modelling environment and can be combined freely with elements provided by SIMULINK and SIMEVENTS (white and semi yellow blocks in figure 3). The latter comprises discrete event modelling elements like servers or queues as well as switches and gates. The blue function block in figure 3 is the same function block that’s gui is described in figure 2. Those blocks comprise of MATLAB s-functions that are developed to meet the requirements of different tasks. As an example they provide the functionality for time synchronisation or saving of data.

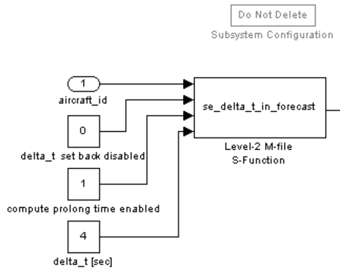

Figure 4 insight of a function block

The orange dyed data storing blocks in figure 3 can be placed at the beginning or the end of a working block. They’re storing free configurable messages and different data types like integers, doubles or characters in the database. These function blocks can be placed at any location within the sub-model. They also can be used in the first user level. SIMULINK/SIMEVENT data sinks can be used at any level as well, as shown in figure 2 (white block).
The blue function block in figure 3 matches two tables to evaluate the actual number of passengers waiting at the gate. The same block is shown in figure 2 but used in a different configuration.
When placed in a working block every function block can be accessed from any level above. For example the block from figure 2 is situated in the underlying level and the user gets access within the first level through the working block. Just as seen in figure 2 the number of inputs to configure a database request is generic and can be defined in the edit field “# inports”. Furthermore the number of outputs can be chosen freely. Every s-function of this “third user level” is designed to be generic, so that an easy and abstract modelling respectively parameter configuration is possible. The third level can be seen as a development level which calls for a profound knowledge of the MATLAB and SIMULINK domain. A third level s-function of a function block is shown in figure 4. This example computes the amount of time the boarding will be delayed in the simulation (cyan block in figure 3).
As an example for modelling within the second user level a stochastic effect should be described.
In [6], Fricke describes statistical distributions for process durations on the critical path of short and midrange flights. This can be modelled within the first user level by configuring the stochastic parameters of a working block. He also found a correlation between the start of a process relative to the arrival delay of the aircraft. The identified density function parameters of the starting times of processes like cleaning and fuelling are described under consideration of five arrival delay classes. This effect can be modelled within the second user level by incorporating a function block which generates the correct random numbers for the starting point of the process depending on the arrival delay. This start times define the beginning of the process. It has to be considered that the stochastic method can not be applied to a TA model with serial connections between the observed processes. Because of the inherent serial dependencies the stochastic functionality would be suppressed. The regarded processes would have to be modelled as parallel activities in the first user level.

3. BOTTOM UP APPROACH AND MODULARITY WITHIN THE EINVIRONMENT

In the previous chapter some of the functionalities of the presented environment regarding the top down approach were discussed. To use the broad knowledge of domain experts it is necessary to apply the bottom up approach.

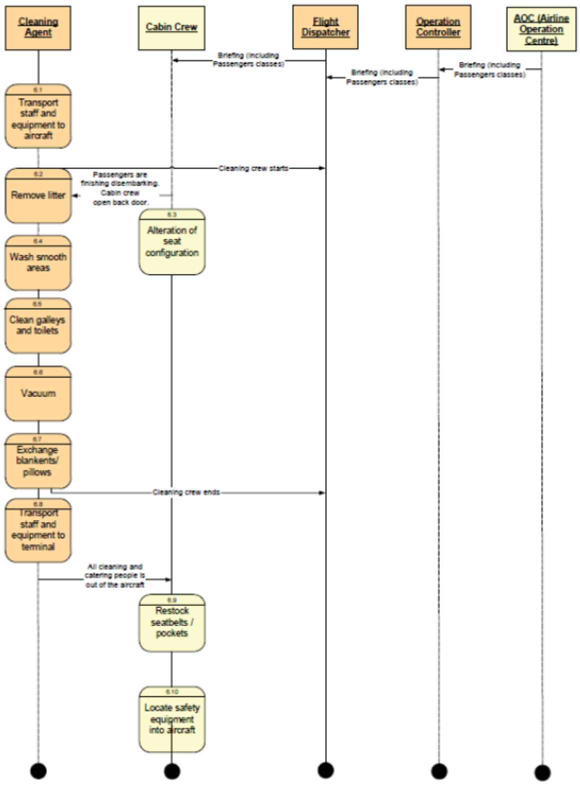

Figure 5 UML chart of the cleaning process

Because of the extended and profound understanding of particular processes and interdependencies a domain expert is able to judge a model with respect to realism.

The hierarchical modelling functionality can be applied to an operational concept of the turnaround developed by domain experts with operational experience [7]. This concept incorporates real operational sequences and responsibilities of the particular actors involved in the process at different levels. It includes a detailed description of the serial and parallel dependencies of every sub-process of the TA as well as the description of all participating partners or "working groups". Every working group is in charge of particular functions within the system. The connection between the several actors is described on a highly detailed level. The implementation of this concept, which is formulated in the form of UML charts, is demonstrated by the means of hierarchical modelling

Figure 5 shows the cleaning process of the TA concept described in [7]. The whole operation sequence (the vertical elements and connections in figure 5) is modelled on the first user level, shown in figure 6 and can then be integrated as a sub-process into a TA model similar to the one presented in figure 1. Blue blocks in figure 6 are sub-processes whereupon grey blocks are events. These events, in this case messages from the executing actor to the others actors (horizontal connections in 5), are necessary for a successful operation and thereby part of the model. The "paths of responsibility" of the different actors in figure 5 are denoted in italic letters in figure 6. The particular working blocks within the sub-model of the cleaning process can be configured by the domain expert as described in chapter 2.1.

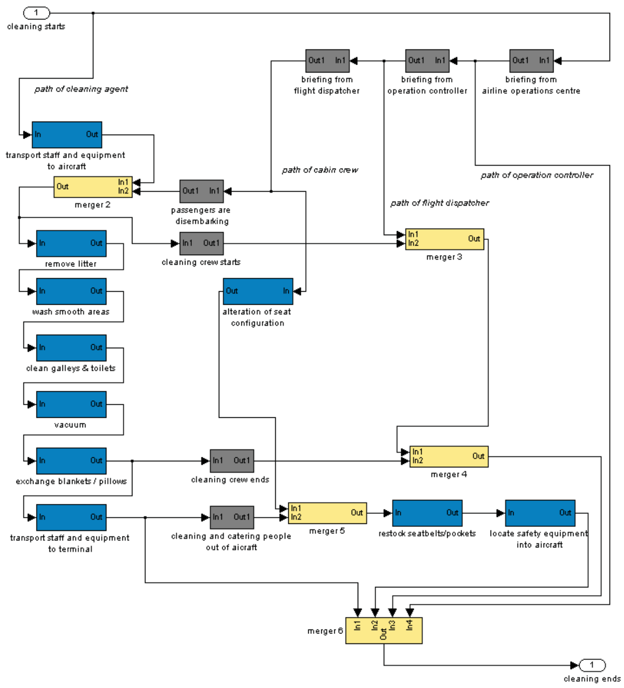

Figure 6 implementation of the example in figure 5 in the environment

As mentioned in chapter 1 a lot of systems at the airport show a discrete event character. For example the taxiing can be described with the help of a discrete event point of view. In this example it is an acceptable simplification to view the taxiway as a set of discrete sectors. An aircraft enters a sector which generates a messages and after a given time span, depending on the length of the sector, the aircraft type or a mean velocity, the aircraft leaves the sector again. Thus generating two events, one when it arrives at the sector, the other when it is abandoning the sector again. This for example could enable a departure manager to make use of a better estimation of the ETOT respectively EXOT (with ETOT=EOBT+EXOT) when calculating the TSAT on basis of TOBT and TTOT. A sector would be a working block with parameters described as above as well as the distance to the predecessor. Overtaking within a sector is not possible. The sector is also blocked in one direction during the time an aircraft rolls in the opposite direction. If a sector reaches the capacity it is also blocked for arriving aircrafts.

Besides the sectors there a working blocks for crossings, which inherit regulations of the right of way, junctions and forks as well as sources and sinks which are necessary to model gates or runways. In both latter cases it can be a source or a sink, depending on the kind of traffic (incoming or outgoing).

The deicing process can be executed on the gate or at a remote deicing pad. In the first case it would be integrated in the TA model as a detailed block which encompasses the deicing operation, as shown in figure 6. In this submodel the deicing time depends on the method like deicing or anticing. It also considers the weather conditions. This is because the holdover time (the time span till an aircraft has to be deiced again) is modelled depending on the used fluid (ADF) and the outside temperature.

When modelling the deicing process on a remote position, the modular character of the environment allows the concatenation with the taxi model. In this model the TA process would be modelled as a sub-model and could be connected as a module. The sources of the TA module are the particular gates of the airports. The sinks are sectors of the taxiway which lead to the runway. This submodel calculates the AOBT or EOBT (in case when running in forecast mode). The aircrafts then move through the taxiway which is modelled by sectors and finally reach a deicing pad which is connected in a modular way to the taxi model.

4. CONCLUSION

The modelling environment presented in this paper allows a hierarchical and compositional modelling on two different user levels. Besides the turnaround, on basis of modularity it enables the user to implement other topics into the simulation that hold a discrete event character like taxiing or deicing. It supports a what if probing by a parallel fast time simulation. In case of a generic APOC this can be tested by several agents.
In contrast to the presented environment a Monte Carlo simulation is suitable to measure the TOBT by probability measures. It is not applicable to the current modelling environment because of a relatively high time consume.

5. ABBREVATIONS

TA Turnaround
TAM Total Airport Management
A-CDM Aiport Collaborative Decision Making
AOBT Actual Off Block Time
APOC Airport Operations Centre
EOBT Estimated Off Block Time
ETOT Estimated Off Block Time
EXOT Estimated Taxi-Out Time
SOBT Scheduled Off Block Time
TOBT Target Off Block Time
TSAT Target Startup Approval Time
TTOT Target Take Off Time

6. REFERENCES

[1] EUROCONTROL, *Airport CDM Operational Concept Document*, Brussels, 2006
[2] Meier C. et al., *Total Airport Management*, DLR, 2006
[3] Cassandras, C.G.; Lafortune S., *Introduction to Discrete Event System*, Springer Science+Busines Media, 2008
[4] Förster P., *Beschreibung und Simulation ereignisdiskreter Systeme für Aufgabenstellungen aus dem Bereich des ATM und Airport Managements*, DGLR 2009
[5] Wu, C.-L.; Caves, R. E., *Modelling and simulation of aircraft turnaround operations at airports*, Transportation Planning and Technology 27:1, 2004
[6] Fricke, H.; Schultz, M., *Delay Impacts onto Turnaround Performance*, USA/Europe Air Traffic Management Research and Development Seminar, 2009
[7] Turnaround Integration in Trajectory And Network TITAN, *Operational Concept Document – Issue 1*, Madrid, 2010