OPTIMIZING THE GROUND HANDLING AT AIRPORTS: A STUDY CONCERNING A BUS DISPATCHMENT ALGORITHM

Jochen Wittmann, Carsten Esslinger, Dietmar P.F. Möller
University Hamburg, Department for Informatics, AB TIS

Corresponding Author: J. Wittmann, University Hamburg, Department for Informatics, AB TIS
Vogt-Kölln-Straße 30, 22527 Hamburg, Germany, email: wittmann@informatik.uni-hamburg.de

Abstract. Hamburg Airport is currently being equipped with a new traffic control system. With the help of such a system the actual position of air and ground vehicles can be located and thus the system enables an updated and exact geographical position of all vehicles. This functionality not only increases safety aspects at the airport but also serves as a basis for further optimization. For example, a route planning for ground vehicles based on their current geographical position, the current traffic situation on the apron, and the current task list for the fleed is possible.
For the example of the passenger bus dispatching unit an optimizing algorithm is worked out that has

- to be able to integrate frequent changes in the timetable for arrivals and departures under real time conditions
- to offer an interface that consolidates the needs of an automatic dispatching with the must of interactive interventions by the dispatcher
- to consider the geographical situation to calculate realistic driving times
- to consider the restricted hardware equipment that makes an overall optimized day schedule impossible

In this situation, the paper describes a solution, that introduces different states for the buses (as resources) and the diving tasks (as tasks in general) and permanently controls the criterion “shortest service time” with respect on the current position of the buses and the distances between the locations on the apron for a set of the n next tasks to be served.

The paper is structured as follows:

- Introduction to the application area “airport logistics” and to the objectives for optimization
- Specification of the data areas such as resource list, daily schedule for arrivals and departures, traffic graph for the routing, and the estimated traveling times.
- Afterwards, the dispatching problem itself will be specified: For resources and for tasks a state-transition model is introduced, that allows a formal treatment of the dispatching process.
- For the optimization algorithm itself, a set of tasks is introduced that holds the currently optimized jobs and that is automatically updated if external changes in the flight schedule or in the job sequence in general occur. Special attention is paid to the interactions of the human dispatcher who influences the set of currently optimized tasks by selecting the set of tasks currently treated by the optimization and even by manual assignments of resources.
- A separate section will describe the optimization algorithm in a pseudo-code version.
- The interface is described by a detailed explanation of a typical screenshot.
- In the final discussion, the experiences with the solution are referred with special emphasis on the conflict between completely automatic dispatching and the acceptance by the people working at the airport.

1. The Context: Ground Handling Processes at Airports

An essential part of the German LUFO [4] research program is to optimize air traffic processes not only by improving the “flying phase” itself but also to analyze and optimize the ground handling processes at the airports. Any reduction of delays in the schedule will imply an increase of efficiency for the whole transport system. There are calculations that come to a cost equivalent for each minute of delay what might be the main motivation for the airlines. The airports, however, have to face an increase of the number of starts and landings that causes bottlenecks concerning the capacity of the airports infrastructure. Therefore the main motivation must be to use the existing infrastructure most efficiently by accelerating the time on earth for each flight.

In this situation new traffic control systems especially for ground handling processes in analogy to the A-SMGCS ([1],[2]) currently are being equipped [3]. These systems locate the current position of air and ground vehicles and thus enable an online update of the exact geographical position of all vehicles. This not only increases safety aspects at the airport but also serves as a basis for further optimization. For example, an exact route planning for ground vehicles based on their current geographical position and the jobs currently being open becomes possible. Furthermore the efficient management of ground vehicles leads to lower fuel consumption and with this costs and emissions are reduced. These effects become more and more important because they contribute to the environmental friendliness of the whole transportation system, a parameter that increasingly is observed by passengers as well as by governmental institutions.

Though, the handling of an aircraft at an airport is a multifaceted process and because of the fact that several independent service providers are involved, it can be seen as rather complex [5]. These subcontractors, which are stakeholders, care e.g. about waste water and fresh water supply, catering, cleaning of the cabin, fuel service, push-back when the aircraft leaves the gate, and passenger bus services. To fulfill these tasks, the stakeholders not only have their own vehicles and personnel but are also responsible for the deployment of these resources.

In general, and in order to enable these services, so called dispatchers are responsible for dispatching the resources available depending on their current position and the set of tasks pending. For this procedure the dispatcher can make use of several software systems.

But these systems mainly give an overview over the current situation on the apron and the tasks already assigned to a certain resource, they do not actively make proposals, how to assign resources, however. For each task the dispatcher has to allocate every single ground vehicle based on information regarding position and status. He must also plan each task himself and has to record feedback from the drivers in the computer system. Further on, the communication with the mobile agents on the apron is implemented by radio communication and especially in the case of heavy traffic a permanent stress factor for the dispatchers.

In order to optimize the described procedure a new system for the disposition of ground vehicles must be introduced. But due to safety, technical and political reasons a fully automated system is not desirable. The existing system which can be noted as a passive support of the dispatcher has to be extended to an active support. This means that the dispatcher is not only provided with all the necessary information, he also receives support whilst planning different tasks. To this effect for the allocation of resources the new computer system should be able to make proposals, which the dispatchers can either accept or reject. Furthermore, the system has to be able to transmit the information automatically to the resources representing the mobile agents being active on the apron. Additionally, with the introduction of this technology the transmitted information can be saved in the vehicle and the driver will be able to get an overview of the tasks. The feedback coming from the driver is also directly connected to the computer system and can be evaluated therein.

This paper introduces the basic concept for solution in the context of the assignment of passenger buses to a certain flight. Main focus will ly on the treatment of the problem in general to be able to transfer the developed algorithms to the other services on the apron and on the discussion how far an automatic assignment may be useful and how much of human intelligence represented by the interactions of the dispatcher should be preserved.

In this situation an efficient software concept is needed which brings together the very special objectives of the innovative and continuously developing thematic scope and a scalable and payable software architecture to work with. This paper introduces such a software concept. It analyzes the special demands concerning the application context first, proposes an adequate architecture for the main levels of the concept, and discusses the usage of the object-oriented modeling and programming paradigm for this very particular application context in special.

2. Problem Specification for Passenger Bus Services

The workflow for the treatment of transportation demands concerning the bus transfer for passengers from a certain gate to the parking position of the aircraft on the apron and/or the way back from the aircraft to the gate can be modeled by the following workflow steps:

First, the bus is at its parking position and waiting for a job. In our case we can abstracted from the assignment of drivers to the vehicles, because in real world, the bus drivers are deployed in shift work and stay in the bus for the whole time. Therefore, vehicle and driver can be treated as one unit in the context of the resource assignment problem.

A general driving task for a bus consists of

- an identification of the flight to be served,
- the position and the time for picking up the passengers,
- the target position, where the passengers exit the bus
- the parking position, where the bus can wait for a new task.
- The driving times between pick up and exiting may not be specified explicitly, they can be derived by the average speed, the current traffic situation, and the distance between start and end position of each stage of the drive.

In consequence, the dispatcher holds a list of available resources, i.e. buses, with their current position and a list of pending tasks. For each task, he/she has to assign one of the resources and specify the transportation task by filling the elements listed above with the current data. The bus driver will get the task description by radio communication and has the choice to accept or to refuse the task. If he refuses, the dispatcher has to find another resource, if he accepts, the assignment was successful, the task is marked as assigned and its progress can be observed when time proceeds.

Modifications of this basic proceeding are as follows:

Normally, the buses will not go back to the parking position specified, but will start serving the next task of their list directly after the last passenger stepped out at the target position. This may be modeled by identifying the parking position of the task that has been finished with the pick up position of the subsequent job.

And, normally the dispatcher will not assign the tasks just in time, i.e. when they actually are pressing, but he tries to make the assignments in advance to be able to plan efficiently for a certain time interval in future. This planning might be figured by the workflow described as well, but it implies the situation that an already assignment might become invalid because the situation changed and the specified attributes of the task might to be adapted to the new situation. In turn, this fact makes the whole future planning invalid and demands for a revision of the all the tasks already marked as assigned ones.

This is, in short, the workflow that has to be covered by the software system. Additionally, there should be a component for online observation and updating the real position of the vehicles. This component is developed independent to the resource planning and optimizing part by the German Aerospace Center (DLR) partner in the project. However, every inconsistency between the current situation in the field and the planned schedule has to actuate a recalculation of all the assignments already scheduled.

3. Specification of the Datamodel for Bus Service Dispatchment

3.1 Data Structures and Top Level Data Flow

According to the description of the workflow above, the dispatchment module has to hold the following information to do resource planning:

1. The service specification as a list of services available and their allocation to the different flights to be served.
   This data structures assigns to each incoming or outgoing flight number the service, the airline has ordered for this flight. In case of passenger bus service, the transfer is ordered for those flights that will park on a position without direct access to the terminal. Additionally, the number of passenger to pick up is correlated to the flight and/or the type of aircraft assigned to the flight. From the number of passengers the number of buses has to be derived.

2. Set of transportation tasks
   This set holds all tasks currently managed by the system. A detailed differentiation will be made in the next subsection introducing the state such a task can have.
3. Set of resources
   This set holds all resources available. A detailed differentiation will be made in the next subsection introducing the state such resources can have.
4. Map of the airport, especially to derive distances between two positions on the apron.

Figure 1 sketches the basic dynamic of the dispatchment system by the top level data flow and the relations between the data structures listed: Triggered by the flight schedule or an on-line message from the airport information system a flight number will enter the dispatcher. In correspondence to its specification, the services that are necessary are attributed to the flight. Thus, a so called service demand that has to be served is defined. The allocation of the resources to the currently pending service demands will be explained in the following subsections. On the top level of the system that is figured here the dispatcher has the opportunity to access the map of the airport by a so called topographic query to get the (driving-) distance between two points as an answer. This paper focuses on the dispatchment and its interface, therefore a detailed view on the states and the state transitions for the objects in the set of transportation tasks and the set of resources is necessary.

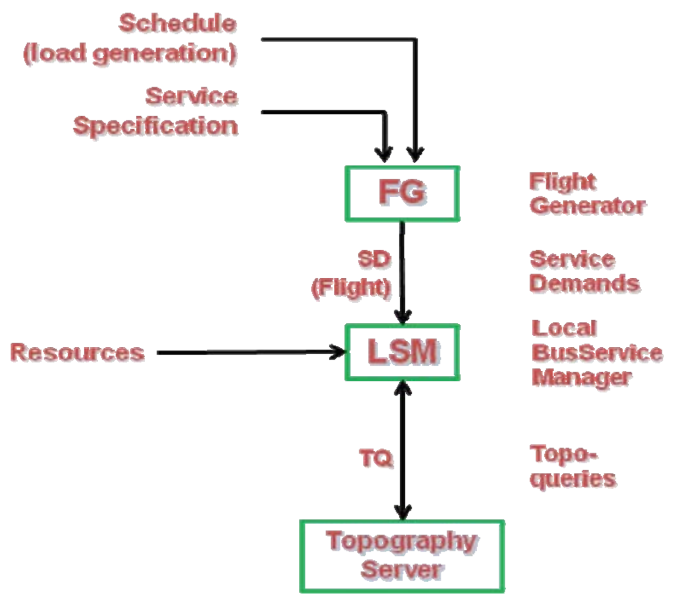

Fig. 1: basic dynamic of the dispatchment system

3.2 States and State Transitions for Transportation Tasks and Resources

To meet the requirements for the disposition of ground vehicles the statuses of the resources as well as the tasks have to be defined. This is necessary because this is the basis for the illustration of the workflow and with this the functioning of the system. A task which is accepted from a stakeholder has the processing status "open" because it has been accepted but not yet scheduled or processed. If the possible allocation of resources is being planned then the task receives the status "scheduled". The present approach of the deployment of resources depends on the decisions made by the dispatcher. With the help of the new system the status will be modeled explicitly. As soon as a resource allocation proposal is accepted by the dispatcher and a defined time before beginning the contract is reached, the proposed resource is assigned to the job. The task will now receive the processing status "in progress" and the information is transmitted to the ground vehicle. Once a ground vehicle sends the confirmation that the task has been processed the status of the contract changes to "completed".

Figure 2 shows the complete state transitions for the tasks with additional states that represent the internal progress of the dispatchment algorithm (“scheduling” and “under reserve”) as well as the possibility to assign resources completely by hand (differentiation of the state scheduled in “auto_scheduled” and “manual_scheduled”), and the online status “interrupted” to express any delay during the task is processed. In addition, the reset of those tasks being already scheduled back to the status open is represented for changes in the course of the real world situation at the airport. The figure emphasizes the role of the human dispatcher, who not only has to control the automatism but who actively has to accept (or reject) the proposals of the automatic dispatchment algorithm.

Like the tasks, ground vehicles can also be given a status. A resource that is in repair or that is not operational for any other reason receives the status "defective". If a vehicle is ready and does not yet have a task it is marked as "free". Once a free resource receives a task the status switches to "reserved".

The new computer system must not only be able to manage tasks but also be able to manage the resources. This will avoid inconsistencies which can arise with the current manual approach. Accordingly, information about the resources and the tasks must be made visible to the dispatcher. At the same time the current process of the tasks also have to be visible.

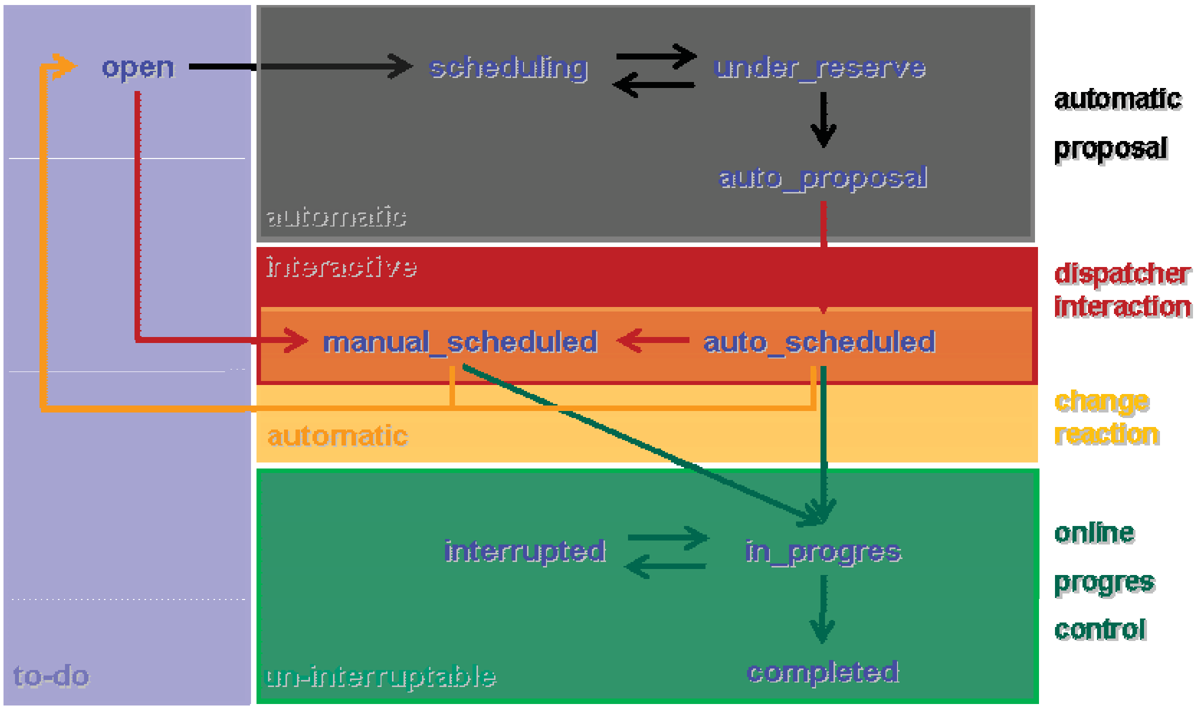

Fig. 2: states and state transitions for the task

3.3 User Interface

On the one hand the system should ensure that a defined number of contracts for dispatching are automatically available. On the other hand the dispatcher must have an opportunity to interact. As required, the dispatcher must be able to make the final decision whether the proposed tasks should be accepted or not. In addition, the manual assignment of a particular resource to a specific task must be made possible.

To meet the requirements for the representation of the resource and task information, a tabular description was chosen. Figure 3 offers a screenshot of the system. On the right side the vehicles are shown and on the left side the different tasks. Four groups are differed regarding the tasks: open, scheduled, in process and completed tasks, whereas each group has its own table.

Whilst all information for the resources is shown, there are differences in task tables. The information regarding flight number, start time, task position, target position and processing status are shown for all tasks. For the open (top table) and dispatched (second table from the top) tasks a checkbox, which is as standard deactivated for open tasks, is visible. If it is activated, the corresponding task switches to the list of scheduled orders, its processing status is "scheduled". If the option is disabled, the task switches back to the list with open orders. The system tries, if possible, not to fall under a defined number of tasks in the list. In this case the system automatically shifts open requests in the list of scheduled.

For all tasks a new proposal for the allocation is calculated every 10 seconds or if the number of tasks changes. This allows the proposals to be up to date. The dispatcher has the possibility to accept or reject proposals by using the check box. Another check box enables the dispatcher to instruct the system to ignore a task when making any further proposals. This feature provides the basis for a manual resource allocation.

Only such tasks can be issued if the allocation proposal, if made manually or automatically, has been accepted. The task has the status "in progress" when the system has carried out the disposition and is therefore included in the list of contracts in progress (third table from the top).

At this point, the scheduled resource receives the task and is clearly assigned to this task. At the same time data regarding the end position and the time of completion for this vehicle are stored in the database. The calculation of the time factor will be described together with the algorithm later in this paper. After a service job has been fulfilled the status changes to "completed". Tasks with this status are placed in the lowest table. The task has now undergone the whole process in the dispatchment system.

At this point, it is possible to mention the parameters by which the system becomes scalable to the current traffic demands on the one hand and the hardware and calculation power on the other. These parameters consist of the number of necessary contracts, the average speed of the vehicles, the operating time at the task and final position, as well as the so-called planning time for the task, which defines the time where the system calculates the allocation.

4. Dispatchment Algorithm

4.1 The Algorithm

The most important innovation compared to the current approach is the active support of the dispatcher. With the help of proposals made by the system the dispatcher will be able to coordinate the allocation of the resources more efficiently. These proposals are developed by an algorithm with the aim of carrying out the tasks at the earliest possible stage. The algorithm does so as follows: Through a query to the database, which contains the tasks, all orders with the status "scheduled" and are not to be manually scheduled are identified. The details of these tasks are saved in a list whereas the orders are sorted in ascending order according to the start time.

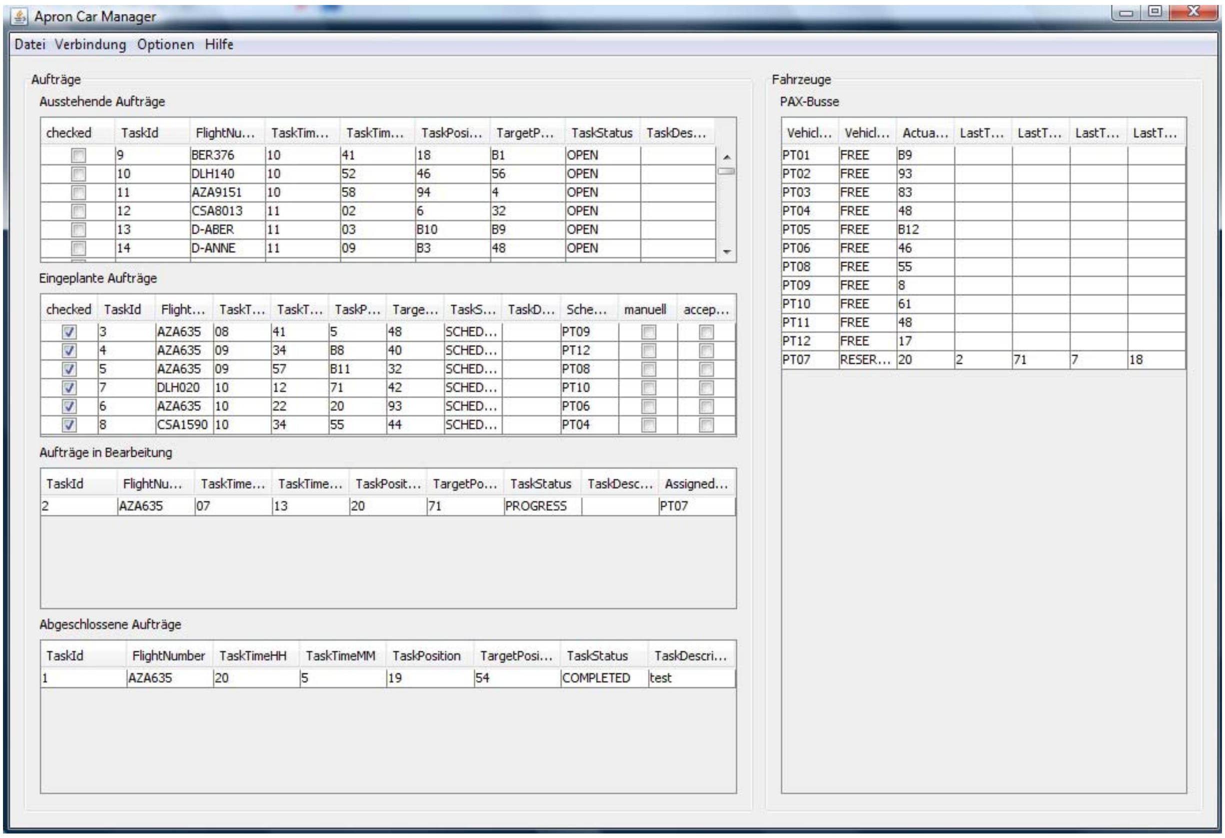

Fig.3: screenshot of the user interface

The algorithm does so as follows: Through a query to the database, which contains the tasks, all orders with the status "scheduled" and are not to be manually scheduled are identified. The details of these tasks are saved in a list whereas the orders are sorted in ascending order according to the start time.

For every identified order a list with the allocation proposals is drawn up.

Each of these lists will now be completed with proposals. A proposed task consists always of the vehicle number and the time on which the vehicle is expected to reach the needed position. This is also checked by a query to a database which has the information of the resources. All resources with the status "free" or "reserved" are chosen.

Free resources are handled as follows: The current position of the vehicle is determined. The journey time from the current position to the assigned position is calculated. In the course of this a table is accessed which has information about the actual distance between the position of the vehicle and the assigned position. An average speed is used to determine the travel time. The travel time is added to the current time in order to show the estimated time of arrival. The calculated time and the vehicle number are noted in the allocation proposal list.

Resources with the status "reserved" will also be taken into account by the algorithm but their information must be processed differently. A reserved vehicle cannot be used immediately since it is already occupied with at least one task. However, the database with information of the resources saves the position which the vehicles will have reached after completing the tasks. In addition, the estimated time of arrival at this position will be available. So, first of all, the travel time between the last assigned position and the new task is calculated. This takes place equivalent to the travel time calculation for the free resources. This travel time is not added on the current time but on the time the resource is expected to reach the final position of the last task. Also in this case the calculated time and the vehicle number are noted in the allocation proposal list.

As soon as all proposals for the free and reserved resources have been made these are then sorted according to the calculated time. For the next tasks lists of proposals are calculated and when all tasks are equipped with a list these will then be compared to each other.

The algorithm then starts with the suggestion list for the first task. Because no resource has been dispatched yet and the fastest available vehicle is at the top of the list, this vehicle is then scheduled for the task. The vehicle number will now be saved in the provided field in the database. Furthermore, the system calculates and saves the new position and the time when the resource has reached its final destination. This calculation is done using the distance table, the average speed and an assumed operating time. This is the time period which is used to serve the task position and target position. The first task has now been successfully dispatched and the system continues with the next proposal list of the following task.

The algorithm takes note of the resource that has been dispatched and checks if this vehicle has been proposed for the next task and is thus not available any more. If the resource is already scheduled, the next allocation proposal is reviewed. As long as there are still suggestions in the list for the task the algorithm will proceed accordingly.

If the case occurs that the list does not provide any more proposals then there is at least one resource that has been previously scheduled! This resource is treated as a reserved resource since a final position and an estimated time of arrival have been calculated.

In this way, even under extreme resource scarcity, the functioning of the system can be guaranteed because all tasks will be handled even with a slight delay.

4.2 Example

Assuming that the system has to allocate vehicles for three tasks but there is only one bus available, which is already planned for another task. The system can solve this resource conflict and at the same time keep waiting time as short as possible. First the three tasks are determined and these are sorted according to the start time and the task data is saved in a list. For each job an allocation proposal list is generated.

The algorithm starts to complete the list with proposals. The system determines the available buses, in this case one. For this bus the estimated time to achieve the job position is calculated. Along with the vehicle number this proposal is noted in the list. This also happens for the other two tasks. Once these three lists are fitted with proposals, the algorithm allocates these proposals to the different tasks.

The first task, which is of short-term notice, is assigned with the only available bus that is in this case also the soonest available. Within the system the algorithm calculates both the new position, as well as the new time as to which the bus is available again. While handling the second task, the system will notice that the only available bus is already assigned to another task. Since there is no alternative the bus is also assigned to this task.

The new position, as well as the new time as to which the bus is available again is also calculated and saved for the second allocation. The system behaves equivalent when handling the third task, so the only available bus is also proposed for this task. By doing so, the system makes sure that all tasks are being processed. Since the proposals are sorted by the start time, the bus will be allocated to the task which is of high priority according to the time available. This will help to reduce the average waiting time. With more information, e.g. the allocation of slots by traffic control for all aircraft, other forms of minimizing the waiting time could be implemented.

5. Discussion

With the help of the new system for the deployment of ground vehicles, various problems of today's approach can be solved. Inconsistencies caused by missing, incomplete or late synchronization of the systems used are eliminated through persistent data storage and the use of a central planning system. The time-consuming and error-prone communication via radio is replaced by a fast and direct communication between the planning system and the system integrated in the vehicle. Various orders can now be transmitted simultaneously to multiple vehicles. Even the transmission of several orders to one vehicle is possible without loss of information, because the order of information is stored locally. Also, the quality of assignments is improved by the proposals drawn up through the computer system. Nevertheless, the dispatcher retains control over any assignment, because the system does not operate automatically. The system only reacts consistent in borderline cases, such as only one available resource and ensures a time-optimized deployment. Because the system is parameterized, the choice of the parameters in order to control the system must be taken into account. Nevertheless, the possibility of influence allows the dispatcher to make the system ignore a task for a certain period of time. Such an operating error cannot be intercepted by the system. Such misconduct is, however, also possible with today's system and might only be avoided by a fully automated system.

The basic approach and the resulting algorithm introduced so far can easily be transferred to any of the other service providers involved to manage their dispatchments. In a further step of our cooperation with the airport Hamburg, we will merge the local dispatchment of the resources under the general view of the treatment of a certain flight with its ground handling workflow integrating the different service providers. For this step, there has to be an integrating and balancing instance or algorithm, because the views of the stakeholders differ and even diverge, building a real multi-criterial optimization problem with all its implications on objective functions and fairness in choosing partial sub-optimal alternatives.

References

[1] Eurocontrol: Definition of A-SMGCS implementation Levels version 1.1, 2005

[2] International Civil Aviation Organisation (ICAO): Doc9830 AN/452 Advanced Surface Movement Guidance and Control Systems (A-SMGCS) Manual, 2006

[3] Loth, S. et al.: Car Management on Aprons (CARMA) – Developing Integrated Solutions for Vehicle Management at Mid-Size Airports; in: Workshop on Aircraft System Technologies (AST 2007), Hamburg, Germany

[4] Luftfahrtforschungsprogramm des BWMi (LUFO) http://www.bmwi.de/BMWi/Navigation/Technologie-und-Innovation/Technologiepolitik/foerderbereiche,did=232982.html, (16.12.2009)

[5] Wittmann, J., Möller, D.P.F. , Gollnick, V.: Conceptual Aspects for a Valuation- and Optimization-System for Transportation Processes with Special Regard on Air-Bound Traffic. In: Möller, A., Page, B., Schreiber, M.: Environmental Informatics and Industrial Ecology, 22th International Conference on Informatics for Environmental Protection, Shaker-Verlag, Aachen 2008, pp. 351-358