WORKFLOW MANAGEMENT OF THE GROUND HANDLING AT THE AIRPORT THROUGH MODULAR SYSTEM OPTIMIZING

Marc Widemann[1], Yousef Farschtschi[1], Jochen Wittmann[2], Dietmar P. F. Möller[1]

[1]University of Hamburg, Faculty of Informatics,
Vogt-Kölln-Str. 30, 22527 Hamburg, Germany

[2]University of Applied Sciences, Faculty of Ecological Informatics,
Wilhelminenhofstr. 75A, 12459 Berlin, Germany

widemann@informatik.uni-hamburg.de (Marc Widemann)

Abstract

Because of the high density of population in most European countries, the possibilities to extend airports are slim. The biggest potential for improvement is to be found in the apron traffic management. An approach on how to realize a reusable solution concerning the ground vehicle management on middle sized airports is presented here. Each one of the stakeholders, like the luggage transport services, has resources in form of vehicles available to achieve his goals. To increase effectiveness of their service, an optimal planning of the resources tasks is essential. Modern localization technologies in combination with flight schedules can provide the information to achieve just that. The optimizer introduced in this article is able to link tasks from the flight plan with resources, by choosing the most suitable option according to the vehicles relative position to the tasks dispatch location. One big feature the architecture of this optimizer presents, is its data management. Saving the relevant task informations in a database, the parallel use of several of these optimizers by the service providers involved in the turnaround process, would actually map the whole dispatch related traffic on the aprons. This modularly obtained data is ready to use, to synchronize the many processes at the airport apron.

Keywords: Optimization, Modular, Workflow, Airport, Turnaround.

Presenting Author's biography

Marc Widemann. He achieved his diploma in 2009 in computer science at the University of Hamburg, where he is now working in the department for technical informatics systems as a scientific employee and Ph. D. student. On the basis of the experience collected during his master thesis in the field of optimization and workflow management at the airport, he has done more research in this context, as seen in this article and other publications (see e.g. [1], [2] and [3]).

1 Introduction

In line with the research project “Wettbewerbsfähiger Flughafen” (WFF)(meaning: competitive airport)[4], a new locating system was introduced to the Hamburg airport, which allowed to pin point the exact position of air and ground vehicles with the help of sensors. The so called Advanced Surface Movement Guidance and Control System (A-SMGCS) is an apron guidance system currently being tested at the Hamburg airport. It consists of a framework of transponders the majority of the airplanes are equipped with, allowing to communicate the location in addition to the identification information of the plane, and an accessory grounded radar system, that facilitates the recognition of the remaining air and ground vehicles like luggage, catering and fuel trucks and passenger busses on the aprons. The goal of this system in particular, is to achieve a better awareness of the situation on the ground. The system is reliable even with bad sight conditions like fog or snow and allows for real time visual display of the traffic condition.

This new level of traffic supervision allows to react in a more secure, intelligent and efficient fashion on the current situation of the traffic. A big part of this is to automate the scheduling strategies of the stakeholders, whose constraints are sufficiently known. Eventually this will also have an economic impact: by means of optimized routing, the services efficiency could be increased, fuel costs reduced, shorter idle times would allow for less delays, therefore boosting the airports quality of service and maximizing cost effectiveness. Due to the restrictions tied to the expansion of an airport[5], the danger exists for a lot of medium scaled airports like the Hamburg airport, to become a bottleneck in the current air traffic system. As up to date economic reports of the airlines suggest, the future growth of the air traffic will affect exactly these type of airports[6]. Should they not be able to keep up with the demand, besides the evident economic losses for the airports, customers would have to deviate to either major airports, or choose other means of travel like trains or buses. The key to stay competitive on an economic level, is to increase the efficiency of the use of the resources. One improvement opportunity at the airport Hamburg would be the procedural method of Ground Stars, the stake holder handling the luggage. Up until now, this service provider is still using a magnet board to visualize the task locations and voice radio for task assignment. These methods require highly trained personnel to assess the situation on the aprons to allocate the tasks correctly and even then, are still more susceptible to errors and unclear communication than a electronically assisted workflow.

The role of the University of Hamburg in this project consists of the fleet resource planning using the example of the luggage handling. The most important aspect of this commitment, is the implementation of an algorithm, that allows an automated resource scheduling, taking into account the conditions the apron traffic imposes. In doing so however, the problem of integrating this new system in the system landscape has to be attended to. An interface to the Communication, Navigation and Surveillance (CNS) has to be developed, to interact with the systems of the other project partners, and in doing so, allowing the communication process for the drivers on the aprons to be entirely electronically aided.

2 Interface functionality

The work on the WFF project has been subdivided into parts, who have been assigned to different project partners like the German Aerospace Center, the Technical University of Hamburg-Harburg and the University of Braunschweig. In the context of the project, the vehicles needed to be out fitted with portable devices to accommodate communications with the operation central (seen as “Vehicles” in Fig. 1). The communication has been implemented using simple messages in form of TCP/IP packets, relayed by WLAN between the devices in the vehicles and the so called Vehicle Communication Server (VCS). This server's function was to route the message from the central to the addressed vehicles and send back their answers. The already mentioned A-SMGCS was used in conjunction with a newly developed database baptized FB2000, which is holding all the flight plans, vehicle resource and even limited human resource informations, provided and updated by the airport operators.

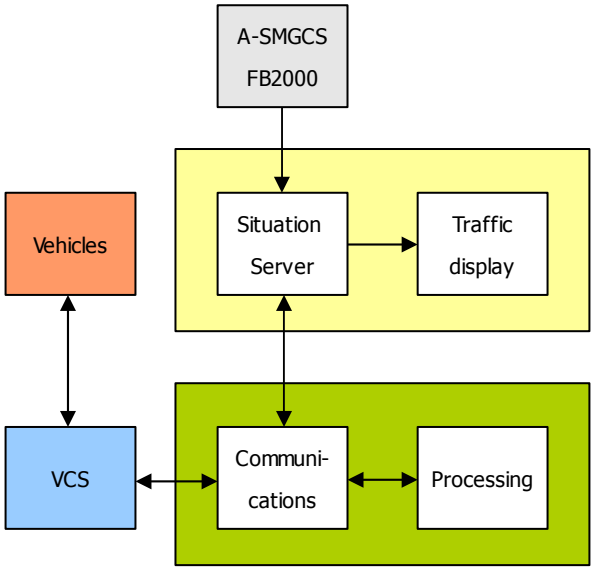

Fig. 1 System architecture

This collected data is used by a situation server, preparing and treating the information to be used by the other project partners, and the traffic display.

The field of activities of the University of Hamburg can be seen at the bottom-left of Fig. 1. One major assignment is the interconnection of the central's user interface (the traffic display) with the connected end points in form of mobile devices in the vehicles on the aprons. As aforementioned, the connection is realized via the VCS. The communications management interacts with the VCS through TCP/IP by establishing a stable access through a dedicated port. To allow the VCS to fulfill its purpose of routing the packets to the correct recipients, and facilitate evaluating the messages in the clients, a common protocol was defined and implemented in the TCP layer of the protocol stack. The header of each message to the resource contains the routing data, in particular the resources identification, while the message body holds the information the resource needs to schedule his work tasks:

- Task identification number
- Flight number associated to the task
- Starting time of the task
- Starting position
- Target position
- Optional task description

Replies from the clients basically have the same header to allow the processing unit in charge of the optimizing procedures explained later on in this article, to identify the sender. The message body holds the status information and position of the resource its current task:

- Task identification number
- Time the message was sent
- Vehicle position
- Confirm code: Task accepted / declined/ processed / discontinued / continued / aborted
- Status code: free / proposed / scheduled / pause / failure

Another function the communication management has to execute, is the maintenance of the situation server. This server holds all the data describing the current situation on the aprons at the airport. The traffic display depends on this data to indicate correct values. This information is also needed to calculate the optimal work schedule. Thus making the connection between situation server and communication management essential. Both systems working in the central, a common access to the database was chosen as a way to communicate changes in the actual condition of the environment. Changes in flight plans are announced by the situation server in a database table in the situation server's data management reserved for the communication management, and the optimized work tasks in turn, are reported to the situation server in a dedicated database table in the situation server's data management by the communications server. Thus allowing an always up to date description of the apron traffic situation. Handling of major problem cases like component malfunctions are not included in the scope of this project, as the system could only be run in a closed environment with test vehicles either way.

Finally, the main challenge of this project, was to implement a way to process all this situation information to optimize the workflow of the luggage service provider. To keep this procedure simple and reusable, the solution was fueled by an important concept.

3 Theoretical approach

Basically the problem presents itself as a search for the shortest path in a state space. The amount of all states can be represented by a directed graph, which has been modeled with all the potential positions of the aprons of Hamburg airport as seen in Fig 2.

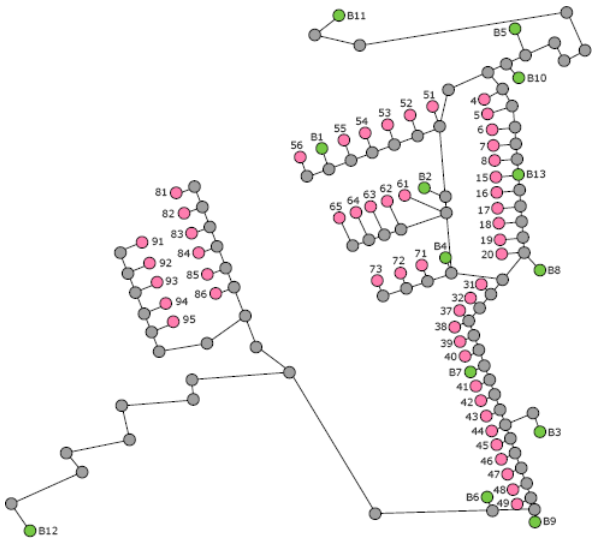

Fig. 2 Directed graph

The most common way to search for an optimum in a state space is the A* algorithm[7]. The problem with this algorithm is that is not always able to find a solution for all inquiries. For most problems, the search domain is expanding exponentially in relation to the length of the path accepted as the solution. And since all the examined nodes are kept in memory, the algorithm usually uses up all available memory before finding an optimal result. Other well known algorithms like the breadth-first or depth-first search, are not complete, thus making it impossible for them to find a solution in infinite graphs as it is the case here. So instead of using the point to point relation between the nodes with their adequate path costs to compute the best path in real time, a matrix was

created based on those informations, to contain all the possible combinations of positions, and the respective costs to travel between these. Tab. 1 shows an excerpt of this matrix.

Tab. 1 Distance matrix

m	4	5	5A	6
4	0	128.5	128.5	183.1
5	128.5	0	0	133.6
5A	128.5	0	0	133.6
6	183.1	133.6	133.6	0

The optimal path between any two locations was analyzed beforehand and entered in this distance matrix, allowing to circumvent the use of a typical path finding algorithm like A* with a quadratic time complexity, and reduce the problem of finding the optimal path to a binary search in an ordered list, with a logarithmic time complexity. Fig. 3 illustrates the advantage for our concept.

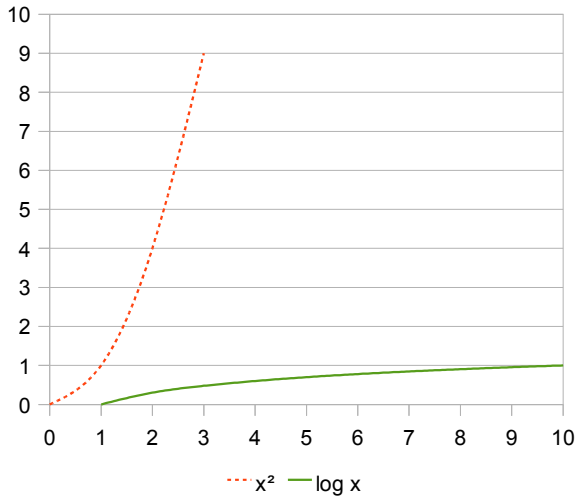

Fig. 3 Complexity comparison

As the number of nodes grows, the binary list search implemented in the application described in this article manages to solve the problem of finding the shortest path between to points far more swiftly. This concept allows for fast computing times for every process involving the task of finding the quickest route between different known positions of a locality, thus improving reusability for similar problems, regardless of the size of the state space.

4 Optimization methodology

The summarized information available for processing in the situation server database is now as follows:

- the usable vehicles
- the flight plan
- the accessible positions of the aprons

Additionally, the processing unit provides its own data management keeping track of the distance correlation among the apron positions, as well as the tuples of resources having denied the completion of a specific task for certain reasons.

This collection of data permits to create and allocate tasks to the resources with the help of an algorithm which is pragmatically described in Fig. 4 and in the following:

1) As the processing unit starts up, new tasks are being created from the flight plan by collecting information on the aircraft, the expected arrival or departure time, and the planned location at this time
2) To determine the number of cycles the algorithm needs to undertake to achieve its goal to respectively assign the optimal resources to their tasks, all the open tasks without any resources assigned are being counted.
3) Time being of the essence, the first task to undergo the process will be chosen in relation to his starting time, determined by the time of the take off or touch down of the associated plane.
4) Now the algorithm checks upon the resources. All free vehicles, not having refused this specific task are candidates for the execution.
5) The solution set composed of these vehicles and their respective distances to the target position is now sorted with merge sort according to the shortest distance (i.e. the lowest number), the first entry therefore being the optimal choice:
   a) If the list is of length 0 or 1, then it is already sorted.
   b) Otherwise divide the unsorted list into two sublists of half the size.
   c) Sort each sublist recursively by re-applying merge sort.
   d) Merge the two sublists back into one sorted list.

   Should the solution set contain equally optimal choices, the sorting procedure would randomly place one candidate at the top of the list.
6) If eventually the solution set is empty, an according message is produced. Else, the first element of this set is added as favored choice to the task. The algorithm continues with step

3) until all tasks have been processed and the count of step 2) reaches null.

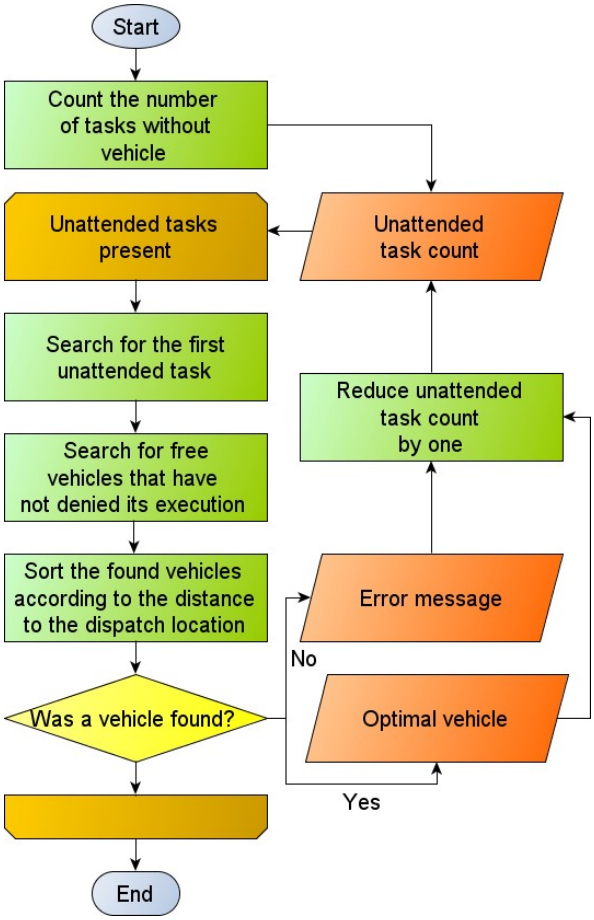

Fig. 4 Optimization algorithm

Once a whole cycle has been completed, the processed tasks are communicated to the controller through the display as explained in 2.2. He now may choose to follow the optimizers proposals, or meet his own decisions. The processing unit operates independently of its own recommendations. This cycle is repeated in a chosen time frame, as the service requires it.

Two further improvements might have been implemented in this algorithm:

- Originally, the algorithm was developed in such a way, that future task assignments were stacked, so as to arrange an hourly schedule for each of the resources of the service provider. This idea could not be further developed, because of the special circumstances the luggage handling has to work with. Although the weigh and number of baggage is known to the central, the volume can not be anticipated before opening the plane's luggage hatch, thus making the number of needed vehicles to handle the task unknown beforehand.
- Due to the time requirements of the project this optimizer was build for, there was no possibility to implement a mechanism considering traffic jams on the aprons on time. therefore the algorithm has no information to check for possible holdups on the route of the chosen resource to it's associated task's starting position, disregarding this eventuality entirely.

5 User Interface

Although the optimizer only holds the programming logic of the whole undertaking and is therefore meant to run in the background, an administrative user interface was designed to keep track of the internal processes. It allows to monitor the communication between involved partners and activities of the optimizing process. Displaying those informations facilitates amongst other things finding errors in the chain of actions and resolving them, although this feature is only interesting for the developers of the application.

The user interface is split up into 5 areas:

a) Open tasks: the tasks that have already been processed by the optimizer but are still waiting for user interaction are displayed here. The identifier, flight number, vehicle number of the resource deemed most suitable for the task, start and destination positions and the optional remarks associated to the task are shown .

b) Running tasks: Here, the tasks that are actually being worked on are displayed. As soon as the controller confirms the association of a resource to a task, this task disappears from the “open tasks” view discussed above and shows up here. In addition to the information already available, this view also displays the vehicle number of the resource chosen by the controller and different time stamps to protocol the time the vehicle was assigned, when the driver of the vehicle accepted the demand and optionally the time when the task was interrupted.

c) Processed tasks: This area shows all the already fully processed tasks. A tasks is fully processed when the vehicle assigned to it reports having finished delivering the luggage to it's destination. The time stamp displaying the time of this last interaction is added to this view .

d) Console: This area shows a protocol of all the activities the optimizer is involved in, like messages being sent or received, updates received from the situation server and the course of action of each optimizing cycle.

e) Status bar: this small area displays general information, like the last time an update from the situation server was received and how many tasks are respectively open, running and processed.

A coarse general overview of the applications user interface with the above numbering is shown in Fig. 5.

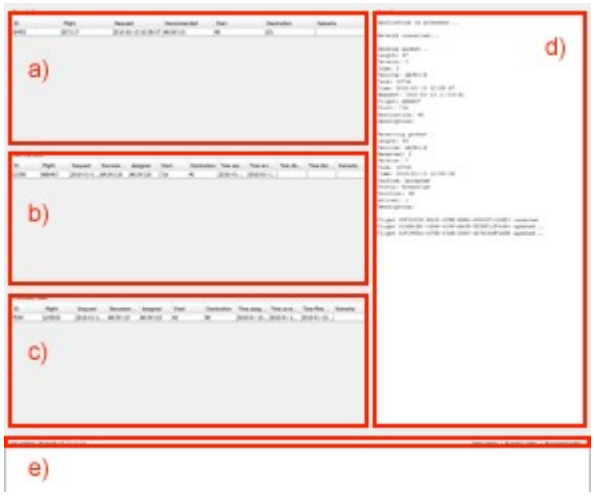

Fig. 5 User display

6 Example

An example will be used to further explain the optimizing process. Assuming there are three flights AB845, BA967 and BT251 that need dispatching (Tab. 2), the optimizing module for the luggage service provider will now try to find the most appropriate resource to handle the tasks.

Tab. 2 Flight table

Flight number	Flight position	Arrival/Departure
AB845	40	Arrival
BA967	87	Departure
BT251	16	Arrival

First of all, the optimizer will create the according tasks A001, A002 and A003 for these flights with the help of the information in the flight plan (Tab. 3). Arrivals are interpreted by the optimizer so that the start position of the task is the park position of the plane and the end is represented by the luggage cellar below the terminal. For departures the mapping is reversed.

Tab. 3 Task table before optimization

Task	Flight	Start time	Start position	End position
A001	AB845	12:00:00	40	cellar
B002	BA967	12:05:00	cellar	87
C003	BT251	12:10:00	16	cellar

Task	Status	Recommended vehicle	Assigned vehicle
A001	Generated		
B002	Generated		
C003	Generated		

For the sake of simplicity, only three resources ARJW123, ARJW456 and ARJW789 will be available for the execution of tasks in this short example (Tab. 4). Since all three vehicles are free, as seen in their status, they will be considered by the algorithm and compared to each other in the following process.

Tab. 4 Resource table before optimization

Vehicle	Position	Status
ARJW123	40	Free
ARJW456	56	Free
ARJW789	5	Free

As one can see, the data needed to identify the vehicles best suited for each task is now present. A comparison of the distances of each resource to the different task starting locations will reveal which one has the shortest path (Tab. 5).

Tab. 5 Distance table

Start	End	Distance
40	40	0.0
56	40	128.5
5	40	610.0
Start	End	Distance
56	cellar	100.8
5	cellar	710.9
Start	End	Distance
5	16	56.2

For each task's starting point a new inquiry is made, starting with the task with the earliest start time, in this case task the order is A001, then B002 and finally C003, with the respective starting positions 40, “cellar” and 16. With each new inquiry, one less comparison is made, since the optimal resource from the precedent inquiry is already selected, and thus not free anymore. Since position 40 is of course closest to 40, resource ARJW123 will be chosen to dispatch flight AB845 in task A001. Position 56 is apparently closest to the cellar, making resource ARJW456 the optimal choice to dispatch flight BA967 in task B002. Finally, resource ARJW789 is the only vehicle left, thus making it's distance to the task's starting point

automatically the shortest, therefore making it the favorite resource to dispatch flight BT251 in task C003. The results of these comparisons are transferred to the tasks, which now have recommended resources (Tab. 6).

Tab. 6 Task table after optimization

Task	Flight	Start time	Start position	End position
A001	AB845	12:00:00	40	cellar
B002	BA967	12:05:00	cellar	87
C003	BT251	12:10:00	16	cellar
Task	**Status**	**Recommended vehicle**	**Assigned vehicle**	
A001	Open	ARJW123		
B002	Open	ARJW256		
C003	Open	ARJW789		

In the same process, each time a resource gets chosen for a specific task, the vehicle is marked as “proposed”, to differentiate his status from the previous “free”, signalizing the algorithm that this resource is not available anymore for other tasks (Tab. 7). That is how the algorithm finds one less resource with each iteration while comparing the distances as seen in Tab. 5.

Tab. 7 Resource table after optimization

Vehicle	Position	Status
A001	40	Proposed
B002	56	Proposed
C003	5	Proposed

Now, the tasks with the recommended resources are reported to the situation server, which displays the new information on the traffic display. The operator has now new information available to help him in the process of managing the tasks of the luggage service. In this example he decides to follow the optimizers proposals for the first task and assigns vehicle ARJW123 to the task A001. Apparently he seems to have some information regarding either the resources or the tasks that lead him to disregard the optimizers proposal, and assigns vehicle ARJW789 and ARJW256 to the tasks B002 and C003 respectively. When the operator makes a choice whether or not to follow the optimizers propositions, his chosen resources are entered through the display in the situation server, which informs the processing unit of the updates trough the communication process (Tab. 8).

Tab. 8 Task table after user interaction

Task	Flight	Start time	Start position	End position
A001	AB845	12:00:00	40	cellar
B002	BA967	12:05:00	cellar	87
C003	BT251	12:10:00	16	cellar
Task	**Status**	**Recommended vehicle**	**Assigned vehicle**	
A001	Open	ARJW123	ARJW123	
B002	Open	ARJW256	ARJW789	
C003	Open	ARJW789	ARJW256	

The rest of the tasks dispatch process involves the vehicle's driver communicating with the operator, informing him of the task's progress, should he have accepted it's processing.

7 Summary and outlook

It has been pointed out how crucial it is to improve the efficiency of the utilization of available resources. Due to limited possibilities to expand today's airport resources, this is the most inexpensive alternative available to increase productivity, as only new software is needed, and some low priced hardware for the clients in the vehicles. The key of achieving this goal, is reducing idle times of resources and facilitate their management through the electronic support of existent processes.

Firstly, the optimization module described in this article allows the drivers on the turnaround to communicate electronically with the central, reducing communication overhead and errors, delivering standardized messages holding exactly the information needed to dispatch the tasks. This method is by far more reliable as the observed standard in Hamburg airport, still relying on voice radio, as affirmed by the employees of Ground Stars.

Secondly, to support the operators in the central, the optimization module will compute the most suitable resource – task tuples. To achieve this, the application will uses a matrix of distance relations entered in it's database in conjunction with a situation server monitoring the traffic situation to find the resource closest to the task's execution. This approach will reduce travel times, therefore also diminishing fuel consumption and increasing service quality for the customer. After finding a suitable solution, the result is proposed to the operator to aid him in his work.

Yet, dispatching an airplane at the airport is still an extensive process, even gaining in complexity due to the fact that a lot of different, unrelated service providers are working together to carry out this work. Each provider has its own resources and personnel

available, and their resource planning is the individual responsibility of each party. At this point, without technical improvements provided by the WFF project, the service providers at the airport Hamburg do not have any technical assistance what so ever, to aid them in synchronizing their work efforts. Currently, no major airports have comparable systems, due to the fact that different stakeholders are taking care of the tasks like luggage and passenger handling, catering and refueling on the aprons. Since the companies in charge of those duties do not want to share their working procedures and keep their data private, no efforts in developing such a system were made. Positive results regarding the particular services yielded by this new system might however stimulate the need for new concepts.

The goal of the university of Hamburg for this project, was not only to deliver a consistent approach for the resource planning of the different service providers, in relation to the aggregated situation in their area of activity. Another goal was also to demonstrate that the standardization of the operation schedules, allows to relate the different providers to another. With the help of common dimensions, the synchronizing and management of individual tasks involved in the same collective dispatching process is made possible. The theory behind this idea and it's applicational uses are demonstrated in article [2]. The modularization of local optimization processes presented in this article demonstrates the possibility to develop an optimization architecture specifically tailored to the particular needs of distinct service providers, and still leave the local optimized results open for a more global synchronizing, allowing to optimize whole chains of mutual dependent processes.

8 Acknowledgments

This work is financially supported in part by the BMBF via the Cluster of Excellence in Aviation Hamburg project L3 and the University of Hamburg. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding bodies.

9 References

[1] M. Widemann, Y. Farschtschi. Modulares Workflow Management und Optimierungssystem am Beispiel des ground handling am Flughafen. Master thesis, Hamburg, 2009. University of Hamburg, Department of Informatics.

[2] M. Widemann, Y. Farschtschi. Macroscopic modeling of a luggage stream in an airport terminal. EUROSIM 2010.

[3] Y. Farschtschi, M. Widemann. Global optimization of local workflow managers using the example of airport Hamburg. EUROSIM 2010.

[4] J. Konopka. WFF Wettbewerbsfähiger Flughafen. Deutsches Zentrum für Luft- und Raumfahrt e.V. http://www.dlr.de/pt-lf/Portaldata/50/Resources/dokumente/verbuende-lufo/VUE_2007_WFF.pdf

[5] G. Kirchstein. Flughafen-Ausbau wird zum Zankapfel der Politik. http://www.welt.de/wirtschaft/article4372697/Flughafen-Ausbau-wird-zum-Zankapfel-der-Politik.html.

[6] I.A.T.A. Passenger Traffic Recovers to Reach Six Month High. http://www.iata.org/pressroom/facts_figures/traffic_results/2007-01-05-01.

[7] P. E. Hart, N. J. Nilsson, B. Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics SSC4: 100–107. 1968.