Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 317–326
ISSN 1896-7094

Adaptive Temporal Planning at Airports

Pim van Leeuwen[1], Lian Ien Oei[2], Pieter Buzing[2], and Cees Witteveen[2]

[1] National Aerospace Laboratory NLR,
Anthony Fokkerweg 2, 1059 CM Amsterdam, The Netherlands
`leeuwenp@nlr.nl`
[2] Delft University of Technology,
Faculty of Electrical Engineering, Mathematics and Computer Science,
Mekelweg 4, 2628 BZ Delft, The Netherlands
`L.I.Oei@student.tudelft.nl`, `P.Buzing@ewi.tudelft.nl`,
`C.Witteveen@tudelft.nl`

Abstract. Airports are getting more and more congested with ground handling activities during turnaround as one of the most constraining factors. To alleviate this bottleneck, robustness of the planning of these activities is of paramount importance. In this paper, we present a new idea to solve a strategic planning problem in a way that allows unforeseen, real-time disruptions to be handled in a straightforward and elegant manner. To that end we apply Hunsberger's decoupling algorithm to a Simple Temporal Network representation of the ground handling domain.

1 Introduction

Europe-wide flight traffic is expected to double by 2020 [1]. Given this increase in air traffic, airports will become a major bottleneck in the air transport system. As expansion of airports is often impossible, airport authorities are seeking methods to increase airport capacity by making more efficient use of existing resources. Currently, one of the most important factors preventing this is the occurrence of disruptions, of which many occur daily. Without re-planning, these disturbances may cause a chain reaction and lead to grave consequences. Therefore, re-planning has become a daily necessity.

Given the complex and distributed nature of the airport domain, however, re-planning is not an easy task. It typically involves a large group of parties, each having their own (commercial) interests, resources, and planning constraints. Between these parties, many dependencies exist: changing something in one planning domain may have repercussions for all.

In European Union projects, such as LEONARDO [2], it became visible that in an approach that involved all parties, large communication overhead lead to problems when trying to solve tactical and real-time disruption scenarios. Thus, a general tool assisting all planners in solving disturbances at airports in these phases turned out not to be feasible.

Airport planning is typically subdivided into a number of domains: arrival management, departure management, stand allocation management, taxi planning. In all of these areas, extensive research has been conducted to improve planning and assist planners by means of decision support tools (e.g., [2,3]). Ground handling, denoting all processes that take place when an aircraft is at the stand[3] between flights, is a notable exception. Up until now, not much research has been conducted in this area. This is surprising, since ground handling is recognised as a common and important source of delays in the air transport system, second after only air traffic control (ATC) related delays [4].

This paper presents a new planning approach to support airport authorities in the establishment of a robust pre-tactical[4] ground handling plan for servicing scheduled aircraft. These services (e.g., boarding, fuelling, cleaning and baggage loading) are performed during the so-called *turnaround*: the process of (un)loading and servicing an aircraft at the stand between two flights. Usually several agents (such as a boarding operator, fuelling operator, catering operator) are involved in this process, each performing a part of it.

In our approach the turnaround process is modelled as a Simple Temporal Problem (STP) [5]. We propose a decomposition framework to solve tactical disruptions to the pre-tactical plan, which offers additional flexibility in repairing such disruptions. First, this temporal plan can be decomposed into a task-based temporal plan and a resource constraint system. This allows us to distinguish task repair from resource repair actions. Furthermore, using a groups of actors, we use a temporal decoupling technique to decouple the temporal plan into independently solvable sub plans that allow parties to solve many tactical disruptions locally, making plan co-ordination and negotiation between parties superfluous. Given the large number of plan disruptions occurring daily at airports, and the increase expected in air traffic, such a plan repair tool seems a valuable asset.

The outline of this paper is as follows. In Section 2 we present the basics of the STP approach and its application to ground handling processes on airports. In Section 3 we present our plan repair framework. In Section 4 we conclude and present some suggestions for further research.

2 Background

2.1 The airport turnaround process

The airport turnaround process comprises all ground handling activities that need to be performed at an aircraft when parked at a stand between two flights. These activities need to be performed between in-block (when the aircraft arrives at the stand) and off-block (when the aircraft leaves the stand). Many different ground handling actors (also called *service providers*) are involved to perform these activities (e.g., the cleaning operator, and the fuelling operator). They all have their own (commercial) interests, resources, and planning constraints.

When planning their activities, these actors have to comply with a large number of temporal dependencies that exist between these activities (e.g., catering

---

[3] The term *stand* will be used for both gates and remote stands.

[4] The strategic and pre-tactical planning phases range from half a year until two hours before the day of operation. After that the tactical and real-time phases follow.

and cleaning should take place between de-boarding and boarding). Furthermore, the ground handlers have to take into account so-called *norm times* (specified by airlines), and *minimum service times* (specified by aircraft manufacturers), that prescribe for each aircraft type the start time and duration of each activity.

2.2 Simple Temporal Problems and temporal decoupling

Since the planning of the turnaround process involves a large number of temporal dependencies, we now discuss the *Simple Temporal Problem (STP)* [5] formalism for representation of and reasoning with temporal problems. In general, a *temporal problem* is a problem in which time constraints are involved. In particular, an STP consists of a finite set $X = \{x_0, \ldots, x_n\}$ of time point variables (representing events), and a set of constraints $C = \{c_{ij} \mid i, j \in 0, \ldots, n\}$ between these variables. A constraint c_{ij} is represented by an interval $I_{ij} = [a_{ij}, b_{ij}]$, abbreviating the inequalities $a_{ij} \leq x_j - x_i \leq b_{ij}$ or the expression $x_j - x_i \in [a_{ij}, b_{ij}]$, that indicates that x_j has to occur at least a_{ij} and at most b_{ij} time units after x_i. The time point x_0 is a special time point, called the *temporal reference point*, that denotes a fixed point in time. As it is usually assigned the value 0, it is also called the *zero time point variable*, and also referred to as z. A task t_i, which has a duration, can be represented by two events (its start and end), and a constraint that indicates how long the task will take. A tuple $T = (\tau_0, \ldots, \tau_n)$ is called a *solution* if the assignment $\{x_0 = \tau_0, \ldots, x_n = \tau_n\}$ satisfies all the constraints in C. An STP is *consistent* if at least one solution exists.

There are several types of graphs to represent an STP. In each of these, the vertices represent the time point variables and the edges represent the constraints. A *Simple Temporal Network (STN)* is a directed graph in which each edge $x_i \to x_j$ corresponds to the constraint c_{ij} and is labelled by the interval $[a_{ij}, b_{ij}]$. In a *distance graph* each edge $x_i \to x_j$ is labelled by a weight a_{ij} and represents the linear inequality $x_j - x_i \leq a_{ij}$. As each constraint c_{ij} corresponds to two such linear inequalities, each edge in the STN corresponds to two edges in the distance graph. If in the distance graph there exists a path from x_i to x_j, then implicitly there also exists a constraint between x_i and x_j, equal to the length of this path, which is the sum of its edge labels.

In a *z-partition* the set of time point variables X is partitioned into two or more subsets $X_1, \ldots, X_n$ that all have only the zero time point variable z in common, and whose union constitutes the original set X. A *temporal decoupling* of the STN $S = \langle X, C \rangle$ is a set of consistent STNs $S_1 = \langle X_1, C_1 \rangle, \ldots, S_n = \langle X_n, C_n \rangle$ such that $X_1, \ldots, X_n$ z-partition X, and any solutions for $S_1, \ldots, S_n$ may be merged to form a solution for the original STN S. In other words, temporal decoupling guarantees that even if each of the sub-STNs S_i is solved completely independent from the other sub networks S_j, the simple union of the individual solutions constitutes a solution of the original network S.

In 2002, Hunsberger [6] developed an algorithm that given an STN $S = \langle X, C \rangle$ and a partitioning of X produces a temporal decoupling of S. Here the global idea of this algorithm will be sketched, using an example with two subnetworks. The general case is analogous. Suppose that STN S is z-partitioned by

S_X and S_Y. An edge in the distance graph representation is called an xy-edge, if it connects a time point x in S_X with a time point y in S_Y. If there exists a path from x to y through z, that has a length equal to or shorter than the length of the xy-edge itself, the xy-edge is said to be dominated by a path through zero and may be removed. The idea of Hunsberger's algorithm now is to add constraints c_{xz} and c_{yz} for each xy-edge, until all xy-edges are dominated by a path through zero, and thus have become redundant and may be removed.

2.3 Modelling the turnaround process as an STN

The airport turnaround process can be modelled as an STN. In this model, for each aircraft, a set of time point variables X can be defined, that contains x_0, the in-block and off-block times, and the start and end times of all ground handling activities that have to be performed. The set of constraints C can be obtained by combining the temporal dependencies between activities with the norm times and minimal service times of all activities. The norm times specify the earliest start time and maximum duration of each activity. The minimal service times specify the minimum duration of the activities. For planning of all ground handling processes at the entire airport, a global STN can be constructed by combining all STNs of individual aircraft. The temporal reference point x_0 (common to all individual STNs) can be used to link the networks. The following example shows how this is done, and how the global STN can be decoupled into separate sub-STNs for each type of ground handling service.

Example 1. Aircraft X (KL310, type B737-300) is scheduled at Stand A17 to go in-block at 12:00 and off-block at 13:15. Aircraft Y (LH200, type MD11) is scheduled at Stand A23 to go in-block at 12:05 and off-block at 14:10. For both aircraft two ground handling services have to be planned: fuelling and boarding. Fuelling has to take place before boarding. Boarding has to end at most 15 minutes before off-block. Table 1 shows which norm and minimal service times apply for these types of aircraft.

Table 1. Norm and minimal service times for Aircraft X and Y.

Activity	Norm start time	Min. service time	Norm duration
B737-300 fuelling	8 min. after in-block	10 min.	37 min.
B737-300 boarding	32 min. after in-block	5 min.	18 min.
MD11 fuelling	11 min. after in-block	17 min.	59 min.
MD11 boarding	80 min. after in-block	16 min.	36 min.

The data for Aircraft X can be modelled as an STN as follows: there are seven time point variables $X = \{x_0, \ldots, x_6\}$, where x_0 = temporal reference point, x_1 = in-block, x_2 = start fuelling, x_3 = end fuelling, x_4 = start boarding, x_5 = end boarding, and x_6 = off-block. Let x_0 = 12:00. From the precedence constraints, and the in-block, off-block, norm and minimal service times, we can derive the following constraints that exist between these variables: $c_{01} = [0, 0]$, $c_{06} = [75, 75]$, $c_{34} = [0, \infty]$, $c_{12} = [8, \infty]$, $c_{23} = [10, 37]$, $c_{14} = [32, \infty]$, $c_{45} = [5, 18]$, and $c_{56} = [0, 15]$. For Aircraft Y an STN can be obtained in a similar way, after which both STNs can be combined. This global STN is depicted in Figure 1. In this figure, the variables $x_1, \ldots, x_6$ correspond to the plan for Aircraft X, and the variables $y_1, \ldots, y_6$ correspond to the plan for Aircraft Y.

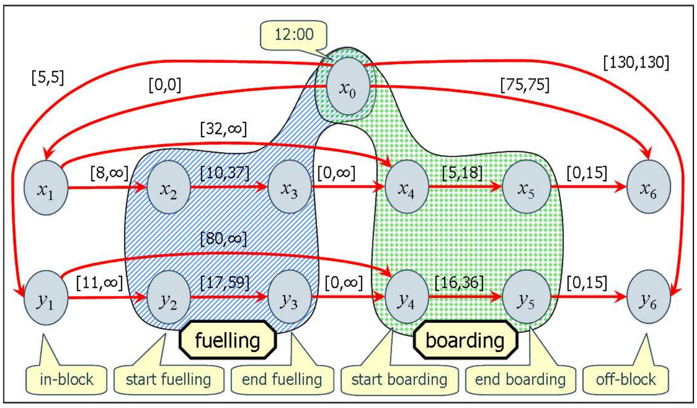

Fig. 1. An STN for Aircraft X and Y and a possible decoupling.

The figure also shows how the STN can be partitioned, such that sub-STNs for the fuelling operators (shaded) and for the boarding operators (spotted) can be decoupled from the network. This can be done using Hunsberger's decoupling algorithm. To decouple a sub-STN, extra constraints have to be added, that make all xy-edges redundant, so they can be removed from the graph. For the fuelling sub-STN these are $x_1 \rightarrow x_2$, $x_3 \rightarrow x_4$, $y_1 \rightarrow y_2$, and $y_3 \rightarrow y_4$. These extra constraints correspond to arrangements the different parties make beforehand, to make sure their plans will not interfere. For example, the fuelling and boarding operator can agree that for Aircraft X fuelling will be finished before 12:56 ($c_{03} = [-\infty, 56]$), whereas boarding will not start before 12:56 ($c_{04} = [56, \infty]$). After adding these two extra constraints, the constraint $x_3 \rightarrow x_4$ can be removed from the network. Once all xy-edges have been removed this way, the fuelling sub-STN can be decoupled from the rest of the network.

3 A Framework for STN-based Plan Repair

As remarked in the introduction, one of most important factors preventing more efficient use of resources in the turnaround process is the occurrence of disruptions. In terms of our STN-model of the turnaround process, a disruption causes a change in the constraints of a temporal event such that the current STN becomes inconsistent. To restore consistency, we have to adapt the STN by changing one or more constraints. Determining the minimal number of adaptations (minimal repair) to restore inconsistencies is an intractable problem [Buzing, PhD thesis, forthcoming]. We therefore have to rely on *heuristics* to perform plan repair on an STN. An example of such a heuristic is *right-shifting* one or more tasks. This can be achieved by the basic constraint propagation methods in [5]. E.g., suppose x_1 is the in-block time of a flight and $c_{01} = [30, 40]$ is a temporal constraint in C. According to the schedule the aircraft should be at the gate at 12:30. However, we have 10 minutes "slack" time. If the aircraft arrives five minutes late, this can be accommodated by right-shifting the in-block time accordingly.

However, one should not concentrate on minimal repair planning alone. Even more important in applying plan repair in complex application areas, like airport planning, is that a *minimal number of parties* is involved in the plan/schedule repair process. Here, temporal decoupling (described in Section 2.2) can be used to decouple the plan of one party (e.g., the fuelling agent) from all other agents. If a disruption occurs, we can investigate if one the decoupled subnetworks has become inconsistent. If so, we can try to repair these subnetworks locally.

Often, however, it is not possible to repair an inconsistent STN by applying simple repair heuristics and it requires another view of the way the STN has been composed. Therefore in this section we will develop a hierarchical multi-actor STN-framework that enables us *(i)* to determine a minimal subset of actors that should be involved in repairing the current temporal plan and *(ii)* to use a set of plan repair heuristics that can be used by these agents to repair their individual temporal plan without affecting the other parties.

The idea of our model is to decompose the temporal planning problem underlying the resulting STN into a *task component*, a *resource component* and an *actor component*. Based upon this decomposition, a hierarchy of three STNs can be distinguished and the planning of repair actions can be based upon the inconsistency of a subset of them. We will discuss our model in the next subsection.

3.1 Decomposing an STN

In any temporal plan we can distinguish at least three components. First, we have a set of activities or *tasks* that have to be performed. Typically, these have a duration and temporal relations (before, after, during) between tasks. The execution of tasks requires *resources*, e.g., vehicles or processing units. Each of the resources can perform a limited amount of tasks at a time (its capacity). Such capacity constraints often will generate additional constraints besides the temporal constraints. Typically, *agents* have to use these resources to execute tasks they are responsible for. Often, agents are organised in groups where the members within a group are cooperative, whereas groups (companies) are assumed to behave self-interested and as independently from the other groups as possible. Such groups induce a partitioning of the set of agents such that each partition wants to have a plan that can be executed independently from the others.

We will discuss the consequences of this decomposition into task, resource and agent properties. Here we assume that a basic STN $S = \langle X, C \rangle$ has been given, representing a set of tasks where resource constraints have been accounted for. We will start by decomposing S according to the set of actors/agents.

Actor decomposition Assume we have a basic STN $S = \langle X, C \rangle$ whose tasks have to be carried out by agents using the resources distinguished. Here, we have a set $A = \{A_1, \ldots, A_n\}$ of agents. Each agent A_i belongs to exactly one group of agents $\mathcal{A}_j$, $j = 1, \ldots, k$. These groups $\mathcal{A}_i$ want to be independent in executing their tasks and are responsible for a disjoint subset X_i of variables. Therefore, the set $\mathcal{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$ induces a partitioning $[X_1, X_2, \ldots X_k]$ of the temporal variables in S. The idea of *temporal decoupling* S is to construct for each group of actors $\mathcal{A}_i$ its own STN $S_i = \langle X_i, C_i \rangle$ such that any solutions chosen by the groups $\mathcal{A}_i$ for their own STN can always be merged into a total solution of S.

Regarding plan repair, temporal decoupling creates a repair hierarchy: if a disruption occurs, the original STN S might remain consistent, while one or more of the sub-STNs S_i are inconsistent. Then either one of the groups should try to repair its plan, e.g., by right shifting its activities, or some parties should try to reconsider the decoupling of their activities. If the original STN S is also inconsistent - and thus one or more sub-STNs are also inconsistent, the repair cannot be achieved by the reassignment of activities to the parties involved alone.

Task and resource decomposition If we abstract from the set of (independent) actors involved, a basic STN $S = \langle X, C \rangle$ can be viewed as the result of two constraint systems: a task constraint system and a resource constraint system. The task constraint system contains all the information about the temporal dependencies between the tasks to be performed, *irrespective* of the resources available. In fact, on this task level, we assume an infinite amount of resource tokens for each type distinguished. Such a task constraint system can be represented as an STN $S_t = \langle X, C_t \rangle$. For example, in our turnaround model, this STN exactly describes what has to be done and which constraints apply between the tasks to be achieved without paying attention to the actual resources available.

The resource constraint system describes the set of resources available together with the type of task that can be processed using the resources and the processing constraints that apply to these resources. This system can be easily represented as a tuple $S_r = \langle X, C_r \rangle$ where C_r describes the resource constraints that apply in performing the events X. For example, if two planes have to be fuelled, the given time constraints for fuelling might overlap. If, however, there is only one fuelling agent, a resource constraint preventing the simultaneous occurrence of two or more fuelling events would enforce a strict separation of those time constraints. Notice that often the resource constraints will encode *mutex relations* stating that one or more temporal variables cannot overlap. Such a constraint often needs to be represented as a disjunction of time intervals instead of simple intervals like in an ordinary STN[5]. The resulting STN $S = \langle X, C \rangle$ can be seen as the result of both the task and the resource constraint system and it is not difficult to see that inconsistency of S_t will imply inconsistency of S.

---

[5] For a detailed description of these resource constraints, see the forthcoming PhD thesis of P.C. Buzing, Faculty of EEMCS, Delft University of Technology, 2008.

With respect to plan repair, this part of the hierarchy can be used as follows: whenever S becomes inconsistent, we also check S_t. If S_t is consistent, it indicates that we might repair the plan by satisfying the resource constraints in an other way or by relaxing the resource constraints, for example by adding one or more resources. If, however, S_t is also inconsistent, this clearly indicates that we should adapt one or more tasks, for example by changing one or more temporal constraints or removing one or more tasks.

The hierarchy that results from these decompositions is depicted in Figure 2. Here the most general STN is the task-based STN S_t containing only the task constraints. This STN can be refined to the basic STN S by adding the resource constraints S_r. The basic STN S can be refined into a set of subnetworks S_i. If a disruption occurs, we first try to establish which STNs are affected. Suppose that inconsistencies are detected on the lowest (sub-STN) level. To determine the right level of repair, we first check whether or not the inconsistencies are still present at the basic STN or the task STN level. If not, we should be able to restore consistency either by repairing the individual sub-STNs or changing the agent partitioning. If, on the other hand, the basic STN is also inconsistent, but the task level STN is not, we have to change either the way resource constraints are satisfied or the resource constraints themselves. Finally, if the task level STN is inconsistent, we should change the tasks and/or their temporal relationships.

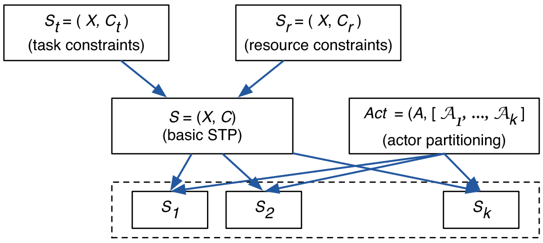

Fig. 2. Hierarchical decomposition and partitioning of an STN S into a task constraint system S_t and a resource constraint system S_r. We can use S and a set of actors A with their group decomposition to compose a set of sub-STNs S_i by temporal decoupling.

The advantage of this decomposition now should be clear: by using the actor decomposition we control the number of parties involved in the plan repair process, while the resource and task decomposition enables us to control the minimality of the repair measures themselves.

Example 2. Figure 3 shows a situation in which Aircraft X, Y and Z have to be fuelled by two fuelling operators, according to the task constraints depicted by the solid arrows. The actor partitioning is as follows: Fuelling operator 1 will fuel Aircraft X, and Fuelling operator 2 will fuel Aircraft Y and Z. The fuelling operators each own one fuelling vehicle, so only two aircraft can be fuelled simultaneously. This is expressed by the resource constraint $y_3 \rightarrow z_2$

(dashed arrow), which indicates that Aircraft Y has to be fuelled before Aircraft Z, and that it takes 5 minutes to travel from one aircraft to the other. The network is decoupled into two subnetworks, one for each fuelling operator, so the fuelling operators can each make their own schedule. Fuelling operator 1 decides to fuel Aircraft X between 13:10 and 13:20. Fuelling operator 2 will fuel Aircraft Y between 12:10 and 12:55, and Aircraft Z between 13:00 and 13:45.

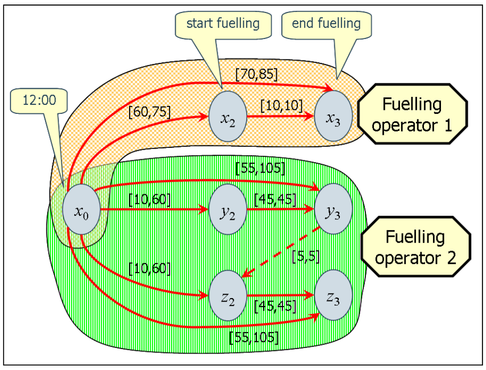

Fig. 3. Fuelling three aircraft by two fuelling operators.

Now these plans may be disrupted in several ways. We present some examples of disruptions and how they can be solved using the decomposition framework.

- *Aircraft X has a 15 min. delay.* Constraint $x_0 \rightarrow x_2$ is reduced to $[75, 75]$. Fuelling operator 1 can solve this disruption locally by right-shifting the task of fuelling Aircraft X, so the aircraft will be fuelled between 13:15 and 13:25.
- *Aircraft Y and Z both have a 20 min. delay.* Constraints $x_0 \rightarrow y_2$ and $x_0 \rightarrow z_2$ are both reduced to $[30, 60]$. The basic STN has become inconsistent (because $30 + 45 + 5 > 60$), but the task STN is still consistent - only the resource constraints have been violated. No resource constraints can be found that enable the two fuelling operators to fuel all aircraft, so additional resources have to be added, e.g., Fuelling operator 3 will fuel Aircraft Y.
- *Aircraft X has a 20 min. delay.* Fuelling Aircraft X cannot start before 13:20, so it will take at least until 13:30, whereas it has to be finished before 13:25. Thus, the task STN has become inconsistent and has to be revised.

4 Conclusions and future work

In this paper, a new approach has been presented to solve the strategic and pre-tactical planning and repair problem of ground handling services at airports. Central to this approach is a decomposition method that first of all allows us to view a temporal plan as the result of a task constraint system and a resource constraint system. Secondly, it allows for partitioning a temporal plan into several independent subplans that can be solved locally and merged again into a conflict-free overall plan. The main advantage offered by this approach lies in its capability to repair disruptions. Here, the hierarchical decomposition allows us to select the right level of repair and the temporal decoupling allows parties to re-plan locally, making plan co-ordination and negotiation between parties largely superfluous. Given the large number of plan disruptions occurring daily at airports, this planning approach seems a valuable asset.

Currently, a prototype is being developed to implement this new approach. Its main goal is to show that decomposition offers important re-planning advantages. It aims to show that, given a disruption, less time is required and fewer actors are involved when re-planning a pre-tactical ground handling plan based on decoupling as opposed to current modes of operation. Future work will include extensive testing of the prototype against a small-scale, yet realistic scenario.

As a next step, research may focus on the level of decoupling. What happens if we partition the domain to the next level (e.g., each fuelling vehicle requiring its own decoupled plan)? Alternatively, one may group certain service providers together (e.g., cleaning and catering) for reasons of efficiency. Another topic concerns the implementation of a mechanism to upscale the level of decoupling when local re-planning is not feasible. In such a mechanism, a new decoupling is produced to group two (or more) service providers together if either one of them cannot find a re-planning solution individually. A final subject of further research is the refinement of the plan hierarchy, for example by allowing each of the parties to use its own resource constraint system and the inclusion of a task assignment system that given the resources and the tasks distinguished determines which task has to be executed using which resource.

After proving the new concept by means of a prototype, we intend to proceed towards a small-scale operational application. In the near future, such an application might be used by airlines and service providers to support their planning of ground handling activities at a real airport.

References

1. EUROCONTROL: *Launch of new collaborative decision making tool for airports,* website (2004).
2. van Leeuwen P.: *Requirements and design document for LEONARDO's collaborative decision making multi-agent system CDMMA,* Technical Report NLR-TR-2003-653, National Aerospace Laboratory NLR (2004).
3. Buzing P., ter Mors A. W., Witteveen C.: *Multi-agent plan repair with DTPs,* In Brown K. N., ed.: Proceedings of the 23rd Workshop of the UK Planning and Scheduling Special Interest Group, Cork, Ireland (2004) 3–14.
4. Wu C. L., Caves R. E.: *Modelling and optimization of aircraft turnaround time at airports,* Transportation Planning and Technology **27** (2004) 47–66.
5. Dechter R., Meiri I., Pearl J.: *Temporal constraint networks,* Artificial Intelligence **49** (1991) 61–95.
6. Hunsberger L.: *Group Decision Making and Temporal Reasoning,* PhD thesis, Harvard University, Cambridge, MA, USA (2002) Harvard Tech. Report TR-05-02.