Actor-Agent Based Approach to Train Driver Rescheduling

Erwin J.W. Abbink\textsuperscript{a} David G.A. Mobach\textsuperscript{b} Pieter J. Fioole\textsuperscript{a} Leo G. Kroon\textsuperscript{a,c} Niek J.E. Wijngaards\textsuperscript{b} Eddy H.T. van der Heijden\textsuperscript{b} 

\textsuperscript{a}Netherlands Railways, NSR Logistics Innovation, P.O. Box 2025, 3500 HA Utrecht 
\textsuperscript{b}D-CIS Lab / Thales Research & Technology NL, P.O. Box 90, 2600 AB Delft 
\textsuperscript{c}Rotterdam School of Management, Erasmus University Rotterdam P.O. Box 1738, NL-3000 DR Rotterdam

Abstract

This paper describes the design of a socio-technical research system for the purpose of rescheduling train drivers in the event of disruptions. The research system is structured according to the Actor-Agent paradigm: Here agents assist in rescheduling tasks of train drivers. Coordination between agents is based on a team formation process in which possible rescheduling alternatives can be evaluated, based on constraints and preferences of involved human train drivers and dispatchers. The research aim is to explore the effectiveness of a decentralized, flexible actor-agent based approach to crew rescheduling. The research system is realized using the Cougaar framework and includes actual rolling stock schedule data and driver duty data. The current reduced-scale version shows promising results for the full-scale version end 2008.

1 Netherlands Railways: Planning and Rescheduling

Applied research on advanced autonomous systems in a real-world domain provides a stimulating environment to demonstrate ‘state of the art’ research results and address the encountered pragmatic and fundamental challenges. The cooperation between Netherlands Railways (NS) and the D-CIS Lab addresses a complex situation: how to reschedule tasks of train drivers in response to disruptions in their schedules. The aim is to arrive at a research system for ongoing experimentation by NS. This paper provides a brief overview of the problem domain, the design of the actor-agent based solution, and the current midway implementation with a brief comparison with related literature.

1.1 Planning

The railway operations of Netherlands Railways (NS) are based on an extensive planning process. After the planning process, the plans are carried out in the real-time operations. Preferably, the plans are carried out exactly as scheduled. However, in real-time operations plans have to be updated permanently in order to deal with delays of trains and larger disruptions of the railway system.

In the operational planning process of NS, the timetable is planned first. The rolling stock and crew schedules are planned consecutively. The timetable consists of the line system and arrival and departure times of trains. The departure and arrival times of trains are such that the timetable is cyclic with a cycle time of one hour. The rolling stock planning supplies each train in the timetable with sufficient rolling stock for transporting the forecasted number of passengers. The crew scheduling stage supplies each train with a train driver and with sufficient conductors. In the past years, NS has successfully applied novel Operations Research models to significantly improve the crew scheduling process [1]. In this paper we focus on an actor-agent based approach for rescheduling of \textit{train drivers}.

NS train drivers operate from 29 crew depots. Each day a driver carries out a number of tasks, which means that he/she operates a train on a trip from a certain start location and start time to a certain end location and end time. The trips of the trains are defined by the timetable. The tasks of the drivers have
been organized in a number of duties, where each duty represents the tasks to be carried out by a single driver on a single day. Each duty starts in a crew base, and a hard constraint is that the duty returns to the same crew base within a limited period of time. Also several other constraints must be satisfied by the duties, such as the presence of a meal break at an appropriate time and location, and an average working time per depot of at most 8:00 hours. Initially, duties are anonymous, which means that the allocation of drivers to duties is still to be made. The latter is handled by the crew rosters, which describe the sequence of duties that are carried out by the drivers on consecutive days.

The total number of train drivers is about 3000. Each day, about 1000 duties are carried out. Furthermore, at any moment in time, the number of active duties at that moment is about 300. Note that, apart from the operational planning process described above, there is also a strategic planning process, which falls outside the scope of this paper.

1.2 Timetable and rolling stock rescheduling

In case of delays or a disruption of the railway system in the real-time operations, the original timetable, rolling stock circulation and crew duties may become infeasible. A disruption may be due to an incident, or a breakdown of infrastructure or rolling stock. On the Dutch rail network, on average three complete blockages of a route occur per day. Rescheduling is required to cope with these situations.

For example, consider a train line between stations S1, S2, S3 and S4. Under normal circumstances, trains on this line are operated from S1 to S4 and from S4 to S1. A train that arrives in S4 returns to S1, and vice versa. However, if there is a breakdown of the infrastructure between S2 and S3, then temporarily no trains can be operated between these stations. In such a situation the timetable is modified by cancelling trips between these stations. Furthermore, the standard strategy to reschedule the rolling stock is to introduce returns of trains in stations S2 and S3: a train that arrives in S2 from S1 returns to S1, and a train that arrives in S3 from S4 returns to S4 (see Figure 1a).

Due to the cyclic nature of the timetable and the structure of the rolling stock circulation the basic principles of timetable and rolling stock rescheduling are rather straightforward. Since crew duties do not have a cyclic nature, crew rescheduling is more complicated (see 1.3). A further complicating issue in a disrupted situation is the fact that the exact duration of the disruption is usually not known exactly. That is, the initial estimate of the duration of the disruption often turns out to be incorrect. As a consequence, the rescheduling process must be carried out several times.

1.3 Train driver rescheduling

Due to delays of trains or rescheduling of the timetable and the rolling stock a number of duties of train drivers may become infeasible. An infeasibility of a duty is due to a time conflict or a location conflict. In both cases, a conflict occurs between two consecutive tasks in the duty.

A time conflict occurs if the end location of the first task coincides with the start location of the next one, but the end time of the first task is later than the start time of the second one. This is due to a delay of the train corresponding to the first task. If after the first task the duty prescribes a transfer of the driver to a task on another train, then the driver is too late for carrying out the second task. In order to make the duties more robust against such time conflicts, they contain a certain buffer time between each pair of consecutive tasks that are carried out on different trains.

A location conflict occurs if the end location of a task in a duty differs from the start location of the next task in the duty. This may be due to the fact that some tasks in the original duty were cancelled because of a disruption. Again, consider the example of the line between stations S1 to S4 (see Figure 1b). If an original duty contains the tasks S1-S2, S2-S3, S3-S4 on a train in one direction, and the tasks...
S4-S3, S3-S2, and S2-S1 on a train back, but the tasks S2-S3 and S3-S2 have been cancelled, then the duty has two location conflicts: the tasks S3-S4 and S4-S3 have to be transferred to another duty, since they cannot be carried out by the originally assigned driver. Furthermore, the resulting hole in the duty between the tasks S1-S2 and S2-S1 can be filled with other tasks. To get more flexibility in the rescheduling process, the final task S2-S1 in the duty may also be transferred to another duty.

Note that at least the duties that are directly affected by the disruption must be rescheduled. But usually also a number of other duties are rescheduled in case of a disruption. Without rescheduling these additional duties, it may be impossible to find an appropriate solution satisfying the operational rules. Moreover, in several crew depots a number of stand-by train drivers are available that may take over parts of duties of other drivers in case of a disruption of the railway system. If it is still impossible to find an appropriate driver for each trip in the modified timetable, then the consequence is that the uncovered trips will have to be cancelled. This requires the rolling stock to be rescheduled again.

Currently, the rescheduling process is carried out in four operational control centres of NS: each region has its own operational control centre. However, this organization requires extensive communication between these centres, since many trains and duties operate in more than one region. In order to reduce the communication between the control centres, the process will be reorganized, and carried out in one control centre in the near future.

2 Socio-Technical Design

In this section the actor-agent based solution to train driver rescheduling is described. First, the context of the system is described; after this, the main principle underlying the actor-agent based rescheduling process is introduced. Subsequently, the actors and agents are introduced, as well as the concept of agent-teaming for rescheduling. Finally, the team formation process is described. Throughout this section, a train-network disruption scenario is used to illustrate the introduced concepts.

2.1 Design context

The system is designed according to the actor-agent paradigm [8], which explicitly recognizes both human actors and artificial agents as equivalent team members, each fulfilling their respective roles in order to reach the team objectives. The actor-agent based design process provides the system with several useful global system characteristics. First, the decentralized approach in which agents use local knowledge, world views, and interactions, contributes to an open system design. This openness facilitates easy reconfiguration and/or adaptation to changing system requirements. Second, combining humans and agents within the system design allows for integrating them at their appropriate abstraction levels: Human dispatchers at the strategic/management level, train drivers at the level of defining and guarding their personal interests, and their respective agents at the level of implementing the strategic/management decisions and resolving actual schedule conflicts.

The prototype system currently being developed focuses on rescheduling train driver duties in real-time over the course of a single day. The term schedule is used in the remainder of this paper to indicate duties assigned to train drivers. It is assumed that any rolling stock plan modifications to cope with disruptions (see Section 1.2) have been implemented, and a new rolling stock plan is in place, to which the driver schedules must be adapted.

2.2 Principle: Resolving conflicts by exchanging tasks

The basic principle underlying the solution process is that of task exchange. Each driver’s schedule consists of a number of tasks (i.e. train driving activities). If in the event of a disruption a driver can no longer perform one or more tasks due to a schedule conflict (location-based or time-based), these tasks are taken over by another driver. In turn, this driver may have to hand over tasks which conflict with the newly accepted tasks to another driver.

To further illustrate the principle, a small scenario is introduced which is used throughout the remainder of this section: The scenario consists of a delayed train (+30 minutes), as a result of which a single driver (designated ‘Dordrecht 109’, or Ddr-109) is directly affected. Figure 2 shows the effect of
the disruption on the schedule of the affected driver: As a consequence of the delay the driver arrives too late in Asd (Amsterdam) on trip A, which results in a time-based conflict with trip B.

The solution process results in a number of drivers exchanging tasks, eventually resulting in all conflicting tasks being reassigned to other drivers. In the following sections, the actors and agents involved in the exchange process are introduced, and the exchange process is described in more detail.

### 2.3 Actors, agents and teams

The following actors and agents involved in the rescheduling process are distinguished (see Figure 3):

- **Dispatcher-actor**: Responsible for the overall rescheduling process. When a disruption occurs, the dispatcher specifies global rescheduling parameters (e.g. number of stand-by drivers that may be used, maximum overtime allowed for driver-agents), monitors the rescheduling process, and evaluates the proposed rescheduling solutions.

- **Driver-actor**: Responsible for execution of a schedule. A driver-actor imposes constraints on the rescheduling process based on the preferences he/she may have with respect to performing his/her duties. These constraints can be hard (e.g. familiarity with rolling stock types) or soft (preferences for certain lines). Each driver-actor is associated with a driver-agent with which he/she interacts in order to reflect personal preferences in the rescheduling process.

- **Dispatcher-agent**: Presents a dispatcher-actor with a management view on the rescheduling process and coordinates the rescheduling process on the level of the team formation process. Rescheduling proposals are presented by the dispatcher-agent to the dispatcher-actor.

- **Driver-agent**: Responsible for resolving conflicts arising in schedules due to disruptions. Each driver-agent is linked to a specific driver-actor which it represents in the rescheduling process. Driver-agents engage in a team formation process in order to find a suitable team configuration in which tasks are exchanged. Driver-agents directly affected by disruptions assume the role of team leader, and other driver-agents join teams when they can help to solve a conflict.

- **Network/duty-analyzer-agent**: Maintains an up-to-date view of the rail network, reflecting any changes in timetable and rolling stock due to disruptions. Driver-agents interact with a duty-analyzer-agent to determine whether it is possible to incorporate tasks of other agents into their existing schedules (i.e. whether it is possible to take part in a task exchange). To this end, the duty-analyzer-agent attempts to find a route for the driver-agent through the rail network on the currently available timetable and rolling stock. Adding tasks to an existing schedule may entail dropping existing tasks from the schedule. The duty-analyzer agent determines the minimum number of tasks to drop, thus maintaining as much of the original schedule as possible.

![Figure 2: Schedule conflict](image)

![Figure 3: Overview of actors and agents](image)
2.4 Team formation process

The coordination mechanism used by driver-agents to find proper sequences of task exchanges is based on team formation: When a disruption occurs, all driver-agents are informed of the impact of the disruption on the current timetable and rolling stock schedule by the dispatcher-agent. A driver-agent affected by the disruption starts a new team and invites other agents to join the team. Driver-agents will accept the invitation if a task exchange is possible. In turn, these agents may invite other agents if additional task exchanges are necessary. Ultimately, a team leader compares and chooses the best team configuration. For reasons of space, the configuration protocol is not described in detail in this paper. Instead, the protocol is described below in terms of the four main phases.

**Phase 1: Discovery:** When a driver-agent determines that a disruption directly affects the driver’s schedule (i.e., the specified train service is associated with a task in the driver’s schedule), the agent assumes the role of team leader. The responsibility of a team leader is to establish and analyze possible team configurations which resolve the agent’s schedule conflicts. All team leaders report their new team leader status to the dispatcher-agent.

In the example scenario, driver-agent Ddr-109 has determined that the delayed task leads to a conflict in its schedule and announces itself as new team leader. The task that it needs to exchange in order to resolve the conflict consists of the trip Asd-Ddr (see Figure 4).

**Phase 2: Team extension:** In this phase, a team leader announces the conflicting tasks to other driver-agents. Each driver-agent then determines whether the announced tasks can be fitted into their schedules. This starts a recursive team extension process in which each team is extended with additional team members able to take over tasks from agents already participating in the team: In case a driver-agent has determined that tasks can be taken over conditionally and that it is worthwhile to join the team, the set of new conflicting tasks of this agent is again announced to other driver-agents. This leads to a recursive addition of layers of team members to the team, resulting in a team consisting of multiple task exchange configurations. In this team extension process, it is possible for driver-agents to participate multiple times in task exchanges within the same team (and in other teams). This allows for teams to discover configurations in which driver-agents ‘trade’ tasks.

Returning again to the scenario, possible team configurations (dashed lines) for exchanging tasks are shown in Figure 4. For each driver-agent, the task that is being exchanged is shown. The figure shows that initially, two driver-agents join Ddr-109’s team: Asd-102 and Zl-102. These agents announce their respective tasks, which invites additional agents to join. Note that driver-agents Asd-102 and Ddr-109 each participate twice in the extension process.

**Cost function:** A cost-function assigns costs to a task exchange based on the status of the driver-agent and the impact of the task exchange. The cost function is strictly increasing and assigns costs to the following elements:

- Extending a schedule past the original end time, and introducing overtime in a schedule;
- Losing meal breaks;
- Replacing stand-by tasks as opposed to regular free time in a schedule;
- Team configuration: Joining as a new team member as opposed to a recurring team member.

**Phase 3: Choosing final team configuration:** The team extension process is considered complete when a sequence of task exchanges is determined in which all conflicts have been resolved, or any remaining conflicts are sufficiently shifted forward in time to be resolved at a later point in time (re-introduced as new conflicts later). At this point, the recursive team formation process is ‘backtracked’: Each layer within a team selects the task exchange associated with the lowest cost. In Figure 5, the costs associated with each potential task exchange in the scenario are shown as labels of the edges. Ultimately, a team leader receives an overview of the potential team configurations from each driver-agent in the first team layer. By comparing the costs of the configurations the team leader selects a final team (solid lines).
Phase 4: Finalizing solution

Once a configuration has been selected, the team leader notifies the involved driver-agents that the configuration has been accepted. When all team leaders have determined a suitable solution for their specific conflicts resulting from the disruption, these solutions are presented to the dispatcher.

2.5 Managing the team formation process

Considering the number of driver-agents involved in the rescheduling process, as well as the high degree of connectivity in the rail network, the number of possible team configurations examined in this manner is very large. In addition, driver-agents are designed to participate in multiple teams and team configurations within these teams to maximize the chance of finding favorable configurations, allowing temporary conflicts. Two mechanisms are applied to manage the dynamic team formation process:

1. Commitment levels: During a task exchange process a driver-agent increases its commitment level to this task exchange. Every increase makes it more difficult for that driver-agent to decommit from the task exchange. In the final commitment level (i.e. ‘full commitment’) a driver-agent must ensure that any ongoing task exchanges that overlap with the fully committed task exchange are aborted.

2. Task exchange strategies: At several points in the team formation process, driver-agents apply strategic knowledge to determine the best course of action. Although driver-agents are considered to be self-interested with respect to the driver’s preferences, these strategies are aimed to guide the team formation process to find solutions that have globally favorable properties, and to dismiss less favorable solutions early on in the solution process. Examples of strategies used by driver-agents are:

- **Cost function**: By assigning different costs to the cost function elements, team configurations with specific properties can be favored in the configuration process. For example, increasing the cost for accepting overtime in a schedule will lead to solutions containing overtime to be dismissed in favor of solutions that modify schedules without introducing overtime.

- **Interest determination strategy**: Determines whether a task exchange is worthwhile before joining a team. A scoreboard mechanism is applied to publish current team scores and inform potential new team members.

- **Decommitment strategy**: Determines how concurrent, overlapping task exchanges are handled.

3 Current Implementation

The described actor-agent based solution is currently being realized: The research system is to be delivered at the end of 2008. This section provides a brief synopsis of our first findings. For the implementation of the agents in the prototype system, the Cougaar [4] agent framework is used. Development of the prototype follows an iterative process, each iteration consisting of adjusting system design and requirements, implementing design-changes, and evaluating system behavior.

In the implementation, interaction with a dispatcher-actor is achieved by means of a GUI which presents schedule representations resembling those of rescheduling tools currently used by dispatchers. Currently, the GUI can also be used by a dispatcher-actor to introduce specific disruption scenarios into the system for testing purposes. In order to run realistic scenarios, a dataset containing a timetable and driver/rolling stock schedules for a full day has been provided by the NS. The current version (September 2008) of the research system is able to find solutions for relatively large disruptions. To illustrate this, results of an example scenario are presented: The scenario consists of a complete blockage between Groningen and Zwolle from 17:00 to 18:00. The number of cancelled train services due to this blockage is 11, which leads to 11 driver-agents to act as team-leaders. Table 1 shows the results of various runs with different driver-agent populations. Some remarks can be made concerning these results:
The additional spare driver-agent added in run 2 eliminates the overtime generated in run 1. The time needed to find a solution is also reduced.

The additional driver-agents added in run 5 represent train drivers that are located far from the actual disruption location. Team configurations containing these agents are quickly discarded in the solution process. The calculation time remains the same as in run 4.

<table>
<thead>
<tr>
<th># driver-agents</th>
<th># task exchanges</th>
<th>total # team members</th>
<th>overtime (min)</th>
<th>calculation time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (no spare drivers)</td>
<td>20</td>
<td>15</td>
<td>96</td>
<td>5:50</td>
</tr>
<tr>
<td>2 (1 spare driver)</td>
<td>16</td>
<td>14</td>
<td>0</td>
<td>3:00</td>
</tr>
<tr>
<td>3 (no spare drivers)</td>
<td>20</td>
<td>14</td>
<td>0</td>
<td>9:10</td>
</tr>
<tr>
<td>4 (2 spare drivers)</td>
<td>equal to 3.</td>
<td>equal to 3.</td>
<td>equal to 3.</td>
<td>6:40</td>
</tr>
<tr>
<td>5 (2 spare drivers)</td>
<td>equal to 3.</td>
<td>equal to 3.</td>
<td>equal to 3.</td>
<td>6:40</td>
</tr>
<tr>
<td>6 (2 spare drivers)</td>
<td>equal to 3.</td>
<td>equal to 3.</td>
<td>equal to 3.</td>
<td>11:30</td>
</tr>
</tbody>
</table>

Table 1: Disruption scenario results

4 Related work

Traditionally, Operations Research approaches are employed in the field of crew rescheduling. Jespersen-Groth et al. present an overview of railway disruption management, including crew rescheduling in [5], describing both the process itself and the directly involved organizations. The authors mention the lack of computerized support for railway disruption management, and a case is made for the use of Operations Research techniques in the disruption management process. Furthermore, the paper presents a comparison with disruption management in the airline industry, in which similar rescheduling goals are distinguished.

Agent-based crew-rescheduling is a relatively new area of research. De Weerdt et al. state in their overview of multi-agent planning [7] that although most researchers recognize the importance of dealing with changing environments, most planning approaches still assume fairly stable worlds. The authors mention contingency planning (plan for all contingencies that might occur) as a traditional approach of handling changes in the environment. As in many situations planning for all possible contingencies is not feasible, the authors argue that so-called plan repair approaches are more realistic: Detecting deviations from the original plan through monitoring, and adjust the plan as needed. DesJardins et al. [2] present an overview of approaches in the field of distributed planning. In the paper, approaches are classified according to the properties they share with cooperative distributed planning (emphasis on forming a global (optimal) plan) and negotiated distributed planning (emphasis on satisfying local goals). The authors argue that only recently research in this field has been concerned with coping with dynamic, realistic environments. To cover this emerging work, the authors introduce the distributed, continual planning paradigm. This paradigm considers planning to be a dynamic ongoing process combining both planning & execution. The work presented in this paper fits this paradigm, as the crew rescheduling process is performed in real-time and disruptions continuously require agents to revise their schedules to cope with new circumstances.

Mao et al. [6] recognize the need for short-term operational planning and scheduling methods in the domain of airport resource scheduling, and present an agent-based approach based on two coordination mechanisms: decommitment penalties and a Vickrey auction mechanism. The coordination approach used in this paper is based on a combination of similar mechanisms: The driver-agent interaction protocol has auction-like properties (agents report costs (i.e. bid) for taking over tasks), and decommitment penalties are determined based on increasing commitment levels. In literature, coordination approaches based on negotiation concepts are often divided in cooperative and non-cooperative (self-interested) approaches. Although driver-agents in our model can in some respects be considered as self-interested agents (driver preferences are included in the agent’s cost function), the agents cooperate to achieve the global goal of resolving disruptions, and agents do not engage in direct competition.

The work presented in this paper can also be viewed in the research context of personnel scheduling. Ernst et al. present an overview of application areas, models and algorithms in this area [3]. Application areas they mention include: Transportation systems, call centres, health care systems, and

---

1 System configuration: Intel Pentium 3.4 GHz, 2.0 GB RAM
emergency/civic services. Their overview does not include crew-rescheduling approaches in any of these areas. The authors indicate the railway crew scheduling process as a relatively new area of research. Furthermore, the need for more flexible algorithms is recognized, capable of handling changing (work) environments and individual preferences.

5 Future work

A proof-of-concept version of the system described in this paper has been successfully demonstrated in December 2007. Currently, the prototype is extended to include all train drivers and more elaborate disruption scenarios. The effectiveness of the team-based task exchange approach is already showing its first promising results; more thorough analyses are planned to be conducted in the final quarter of 2008 when the research system is linked to real-time disruption information. On the longer term extending the system to other rescheduling tasks, such as the rescheduling of conductors, is foreseen.

Acknowledgments

The authors express their gratitude to NS and Cor Baars (DNV / Cibit) for starting this project. The following D-CIS Lab colleagues provided valuable contributions: Louis Oudhuis, Martijn Broos, Sorin Iacob, and Thomas Hood. The research reported here is part of the Interactive Collaborative Information Systems (ICIS) project (http://www.icis.decis.nl/), supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024. The ICIS project is hosted by the D-CIS Lab (http://www.decis.nl/), the open research partnership of Thales Nederland, the Delft University of Technology, the University of Amsterdam and the Netherlands Organisation for Applied Scientific Research (TNO).

References