M A S S I E F

Modelling Disruption Management as a Multi-Agent System to Improve the Prediction of the Function Recovery Time

Terry Nick Bergsma

Master Thesis

MSc. Artificial Intelligence

Information and Computing Sciences

Faculty of Science

Utrecht University

Supervised by:

Dr. Mehdi Dastani, Utrecht University

Laurens Koppenol, ProRail

Wilco Tieleman, ProRail

Universiteit Utrecht

ProRail

Abstract

When a disruption occurs on the rail track, the disruption management of ProRail is responsible for recovering the function of the failed infra object as safe and soon as possible, so the hindrance is minimalized. An important aspect of the disruption management is predicting the function recovery time (FRT). The four main parties of the disruption management faces challenges in estimating the FRT, due to decision making on invalidated information and the lack of information sources. The nature of this problem fits the characteristics of a multi-agent system (MAS) simulation. In the present study, I have built a MAS, which simulates the disruption management deterministically. I extended this baseline model with algorithmic modules and adjusted communication lines between the agents, which aimed to improved decision making on the predicted FRT. I have tested the extended MAS on five scenarios in which a switch was disrupted. The extended MAS predicted the FRT better than the original prediction in four of the five scenarios. The performance of the presented MAS is a proof of concept, showing that MAS modelling and extending the model, makes it able to generate a better prediction on the FRT.

Acknowledgements

This final thesis project wasn't the same without the support of many great people. I express my gratitude and appreciation in the following.

In the very beginning, Paul vd V gave me the opportunity to do my research at ProRail DataLab. On my first day at ProRail, Paul D created a warm welcome, guided me in the complex world of railways and supported me in choosing the right track. I thank both Pauls.

Moreover, I would like to thank my daily supervisors at ProRail, Wilco and Laurens. Wilco is a person full of good ideas and great patience. Laurens has the ability to keep things simple and has no limits in his humour. Both smart and sympathetic guys who played an essential role in doing my research.

I also thank Mehdi, my supervisor from Utrecht University. He taught me the first principles of the multi-agent system paradigm during the starting course of my Masters. His knowledge in this domain of AI is enormous. I appreciate his visit to ProRail and his critical mindset.

Writing my thesis was a process of ups and downs. Looking for a new topic after cancelling the initial subject in the 5th week wasn't easy. I've struggled with believing that I was doing the right thing. It made me worry sometimes. Lucky, I received support from my supervisors, colleagues, friends and loved ones.

Therefore I thank my parents, *myn heit en mem*, for making the Wasgij puzzle and always asked me how my thesis was going. Also my older brother, *myn grutte broer*, for sharing his experience in writing a MS thesis and helped me in keeping my balance. Next, my little brother, *myn lytse broer,* he just started studying medical science. He remined me to appreciate and enjoy the wonderful life as a student, even during thesis times, and make the most out of it till the very end. Lastly, my girlfriend, *myn leave*, she gave me the great idea to go studying the Master's Artificial Intelligence, which is characteristic for the special role she plays in my life.

Above all, I enjoyed every day being at ProRail. I've met so many nice people, from ICT, operation, HR, managers, trainees and fellow interns. I'm especially grateful for the connection I had with the people of DataLab. The young, open and personal atmosphere made me felt at home. It were the small things that made it such a great time. Daily table tennis session, juggling, eating candy and making silly jokes. Also spontaneous unusual office activities, like Paula's rollover and wrestling with Michiel. This all will become one big sweet memory.

Terry Nick

Contents

1 Introduction

This report gives an overview of my research project at ProRail into modelling disruption management as a multi-agent system, with the aim to make better prediction on the function recovery time (FRT). The project is named MASSIEF (meaning massive in Dutch), which is an acronym that stands for *Multi-Agent System Storing Incident En Functiehersteltijd* (translated to English: Multi-Agent System Disruption Incident And Function recovery time). Firstly, I will give a brief motivation, shortly describe the problem and present the research question. At the end of this chapter, I will give an outline of this report.

1.1 Motivation

Disruptions in the Dutch railway network are still inevitable. ProRail, the Dutch railway infrastructure manager, has to deal with 10.800 rail infrastructure related disruptions a year. These disruptions cause delays for train travellers. To minimize delays, ProRail has the aim to solve disruptions as soon, safe and proper as possible. At the moment a disruption occurs, a group of officials and experts comes into play, called the disruption management team. This team is responsible for solving disruptions in a fast and safe way. An important element in the solving process, is predicting the moment on which the disruption is repaired, called the prognosis at ProRail.

A reliable prognosis makes it possible to anticipate optimally on the disruption, in the sense of rescheduling trains, informing train travellers or minimizing hindrance in some way or another. In general, an accurate prognosis and knowing that it is accurate, leads to minimal impact. However, the disruption management faces challenges in making reliable prognosis. Evaluation reports on disruptions and related data analysis have made this clear.

1.2 Problem description

Considering the performance of the disruption management, it turns out that the disruption management experience difficulties in sharing information properly and making decisions based on validated information. This leads directly to inaccurate prognosis about the function repair time. Inaccurate prognosis leads to a higher impact of disruptions on train travellers. In the light of this challenge, ProRail is interested in optimal information sharing and validating the correctness of information between the disruption management team.

1.3 Research Objective

This study aims to make a model that uses additional information according to disruptions of infra and a more effective use of information is realised among the disruption management team.

Therefore, the following research objective was formulated:

> *The more extensive and smarter usage of information by the disruption management team, in favour of facilitating the estimation of the function recovery time.*

1.4 Research question and approach

To achieve the research objective, it is necessary to explore the possibilities to use the current information more beneficial and add sources in the disruption management process. It turns out that in the disruptions recovery process, information is distributed among the disruptions management in an improper and ineffective way, which leads to poor decision making and difficulties in estimating the disruptions recovery time.

Moreover, ProRail has access to information that is currently not used during the disruptions process. To be more precise, information about the engine power for switches, the position of a relay, the movement of switches, exact location of the disruption and predictive models developed for function recovery time. This leads to opportunities to use this additional information in the disruption management.

In short, information that can be gained during the process is hard to share properly, which leads to misinformed people and unnecessary waiting time for information that was already available. As a consequence, people make incorrect decisions and delays the disruption management process.

In the light of these demands, the main research question is as follows:

- Can the disruption management system of ProRail be modelled as a multi-agent system to evaluate the quality of estimated function recovery time when using information smarter through better information sharing among the participating agents.

From this main question, several sub-question are formulated

- How can the disruption management system be modelled as a multi-agent system?
- How can the multi-agent system use information in a smarter way, i.e. how to make the information sharing protocol smarter?
- How to evaluate the modified multi-agent system to check if it improves the quality of the estimated function recovery time?

Once the puzzle is solved, it should give more insight in the disruptions management and facilitate the process of creating reliable repair time prognosis, so it becomes possible to make an alternative train plan in an early stage, which minimizes the delay for train travellers.

1.5 Outline

I will give some background to the disruption processes at ProRail as well as multi-agent systems and predicting function recovery time in Chapter 2. In Chapter 3 I will argue methodological choices. Chapter 4 contains the experiments, evaluation and discussion. In Chapter 4 I will explain the process of developing a multi-agent system (MAS), how this is used for simulation experiments and present the performance of the MAS, which are the results of this study related to predicting FRT. Moreover, I will evaluate the results and I dive into the discussion of the results, limitations and validity of this study. Finally, I will conclude in Chapter 5 by answering the sub-questions and main research question, followed by a few possible future studies.

2 Literature and Background

In this chapter, I will discuss the disruption management at ProRail, zooming in on the participating parties and important principles. Moreover, relevant literature on multi-agent systems (MAS), the development of such systems and prognosis in MAS will be discussed.

2.1 Disruption management – Current Situation

Disruptions and incidents on the rail way are inevitable. Therefore, the disruption management, that has to deal with these issues, plays an important role at ProRail. The disruption management has the mission to solve and communicate as efficient, effective and costumer directed as possible, according to a disruption on the rail infra (Handboek storingsmanagment, 2017). The management consist of several parties, each with its own role, expertise and responsibility. In the next section these roles will be described in more detail. Which parties are participating in the process depends on the type and impact of the disruption. If a disruption causes substantial delay or harms people and nature, the decision has to be made on a higher level, so different parties are participating compared to a more regular disruption. To solve an infrastructure disruption, the parties communicate, share information and make decisions with each other. This process could be distinguished in several steps. In each step, the participating parties have to perform specific tasks, according to their role.

2.1.1 Parties

In this subsection, I will describe the participating parties in the disruption management process. For each party, their role and responsibility will be described.

Dispatcher

The Dispatcher plays a central role for a safe transportation of passengers and goods. In the transportation process, the Dispatcher has the goal to let the trains drive as safely as possible. A Dispatcher has its own area of the rail infrastructure to work in. In this specific area, the Dispatcher is capable of controlling switches and signs. Onwards, the Dispatcher is responsible for arranging a safe work place for people that have to enter the rail track to repair, for instance, a failed switch. So the Dispatcher determines the moment on which it is safe to enter the rail track. If a disruption occurs somewhere, the notification of this is received by the Dispatcher via his own equipment or by a phone call of the train driver. The Dispatcher is part of the traffic control team, which is responsible for the logistic part of the rail infrastructure.

Meldkamer Spoor (*MKS*)

The MKS is responsible for coordinating the disruption management process. The MKS coordinates the process by communicating, sharing information and having an overview of the situation. In the beginning of the process, the MKS makes a *Rapport van Onregelmatigheid* (RVO, which stands for *Report of Irregularity*) in cooperation with the Dispatcher. A RVO is a concise report, containing all the information that is needed for the Contractor to go to the location of the disruption. During the process, the MKS asks for the estimated time of arrival (ETA) of the Contractor and the general leader, current status of the repairing process and estimation of the time it will take to the repair the disruption. Overall, the MKS functions as a central organ that should have a clear overview of the progress being made.

General Leader (*GL)*

The GL has the task to coordinate the operational part of the disruption management process. In the light of this goal, the GL is present at the place where the infra disruption is occurring. The GL discusses with the Contractor the progress of the process. The GL is allowed to overrule decisions, since the GL has an overview of the process, by staying in close contact with relevant parties. The GL and MKS communicate closely with each other about the function repairing process.

Contractor

The Contractor is responsible for maintenance of the infra and the final repairing of the disruption. The Contractor is contacted by the MKS if a disruption occurs. Most Contractor have a PGO contract with ProRail. A PGO contract is based on the idea that a Contractor gets fee for infra failures and receives a bonus if the number of disruptions is lower than expected. So the better the performance on these aspects, the more they get paid. The idea behind this type of contract is that a Contractor is motivated to perform the best they can.

2.1.2. System's sequence

The parties that are active during the disruption system, could be view as a chain-like structure. In the system's structure, parties follow protocols that describe what actions to perform in which order. The process of repairing a disruption could be partitioned in states, that are connected in a chronologically way. In Figure 1 the states that are part of the disruption management are presented. The whole disruption management could be divided in these timeslot. This is a simplistic and clear way of looking at the time intervals of which the disruption management process consist of.

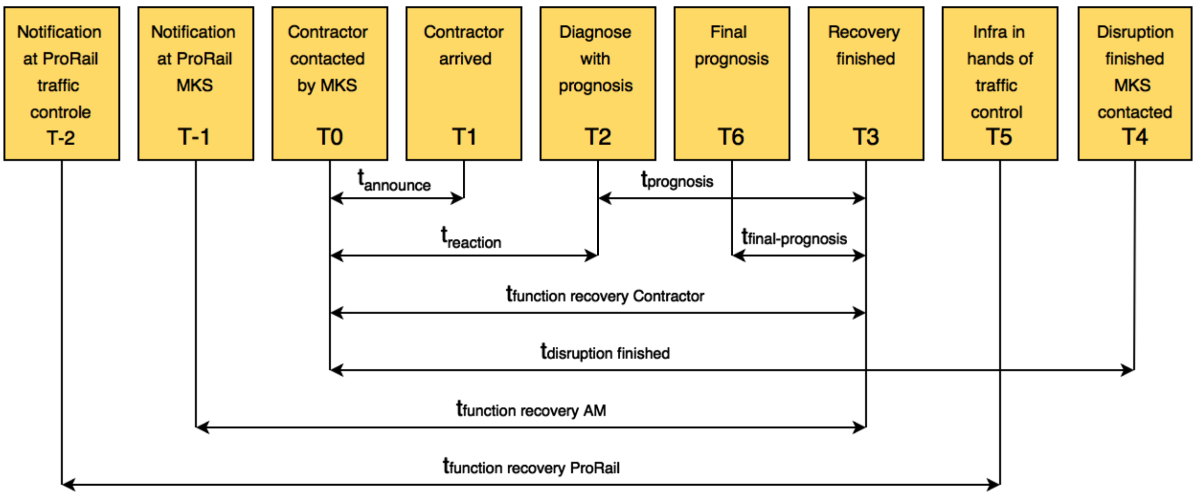


Figure 1: Disruption management portioned in time slots

The process begins with a notification of a disruption, received by the MKS or the Dispatcher. Depending on the type of disruption, the notification could automatically be generated by a control system or by a phone call of someone, for instance a train driver that sees that something is broken on the rail track. After the notification, the Contractor is contacted. An RVO is send to the Contractor.

With the information from the RVO, the Contractor knows where to drive to, when to start driving and which equipment's probably will needed to repair the disruption. This process corresponds with T0 in Figure 1. This forms the starting point of four time slots. Namely, the time is takes for the Contractor to drive to the location ($t_{announce}$). The estimated time of arrival is at T0 formulated by the Contractor and send to the MKS. Another important timeslot is about the time it takes to repair the function by the Contractor ($t_{function\ recovery\ Contractor}$). This is simply the moment from receiving the RVO and the moment in which the function is repaired completely. The end point of this timeslot is a crucial moment in the system's sequence. Since the main goal of the disruption management is repairing the function and to get the rail track working again as soon as possible. Besides this main goal, the failure management has the aim to predict the time on which the function is repaired. In a following section an elaboration on this predicting will be given. The main point of this subsection is, due to the chain-like structure of the system, timeslots are connected and actions performed in one timeslot will influence actions in upcoming ones.

2.1.3 Priority level

In general, a failure notification is processed into an RVO, which consists a priority level. This priority level is determined by the Dispatcher and the MKS. At ProRail, the priority level could be divided into six categories, labelled as 1, 2, 4, 5, 8, and 9. Priority levels 4, 8, and 9 are the non-urgent ones, which lays out of the scope of this research. Priority level 1 and 2 are related to urgent disruptions. Urgent in this sense means that the repairing process of the function should be started immediately, since further damage, delay or risks to the environment is a possible consequence. In this situation, the Contractor is directly contacted and starts repairing the function as soon as possible. Priority level 5 is marked as urgent as well, but with a time appointment. In this situation, there is no necessity to repair the function immediately, but decisions on what to do has to be made directly. That is the reason why priority level 5 is labelled as urgent as well. In practice, most of the time, disruption with priority level 5 results in repairing the function at night, since during night times train traffic is barely executed. To place the priority level in the broader picture of the disruption management, the level of priority determines the tasks of the Contractor in relation to repairing the function, prediction the function recovery time and the way of communication.

2.1.4 Entering the rail track

In relation to the level of priority, the Contractor and the GL enters the rail track on a specific moment of time. This specific moment of time is determined by the Dispatcher. A party is only allowed to enter the rail track if the Dispatcher has explicitly given permission to do so. The Dispatcher has to find a gap in the train schedule on which the Contractor can safely, and with minimum impact on train traffic, enter the rail track. At ProRail this principle is called a BUTA, which stands for the period of time on which the rail track is not been used by trains. In urgent situations, priority level 1 or 2, a BUTA could lead to cancelling trains. So, in the system's sequence, a BUTA is a factor that influences the time on which parties are allowed to enter the rail track, which has influence on the starting times of upcoming actions.

2.2 Function recovery time

In Figure 1 the function recovery time is the time from T0 to T3. Parties in the disruption management are interested in the prediction of the function recovery time. This prediction gives valuable information about the moment on which trains are allowed to use the rail track again. The more accurate the prognosis, the better. Moreover, the earlier the prognosis is given, the better. It is valuable to have an accurate and early prognosis because the train traffic could in this case anticipate on the disruption as proper as possible. The first prognosis is generated automatically right after the disruption intake. This initial could be adjusted by the GL, if he has reasons to do so. The second prognosis is given by the Contractor and GL, after the failure cause has been found. This prediction is highly made on expertise and experience. The prediction depends on the available equipment that have to be used during the repairing process. The final prognosis is given at least half an hour before the function is completely repaired. The reason behind this half an hour is the time it takes to let the train traffic most efficiently anticipated on the disruption. The Dispatcher needs at least 30 minutes to reschedule the trains in such a way that hindrance is the lowest.

2.2.1. Data

During the disruption management, specific information is logged. A platform called SpoorWeb serves as an integrator of all information that is relevant for the process of repairing the function. Loggings in SpoorWeb give a clear overview of specific actions and information that is related to the process. For instance at which time the Contractor arrived at the location, how long it took to find the failure cause and which adjustments have been made on the prognosis. These loggings, starting from the time the failure notification reaching the Dispatcher and ending at the moment the function is definitely repaired, gives insight in cases that the disruption management has to deal with. Moreover, these loggings makes it possible to evaluate the function repairing process.

A research group at ProRail called OPOZ evaluates the disruption management process in cases with significant hindrance (labelled as hindrance-class 1). The reports they make give a detailed reconstruction of the disruption. OPOZ uses various data sources the analyse the quality of the process as good as possible. Each report concludes with some recommendations, for instance to adjust a protocol or extend the authority of a party. I will use these reports for simulating specific cases.

2.3 Agents

In artificial intelligence the agent paradigm is a well-known topic of research. Besides the extended literature one this subject, there is no definitive single definition of the term agent (1). According to Russell, “an agent is an entity that senses its environment and acts upon it ” (2). This is a very simple definition of an agent. A more extended definition is from Wooldridge (1995) which states that an agent is a computer system, situated in some environment and is capable of autonomous action in order to meet its design objectives (3). The definition I will use in this thesis states that an agent is “*anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators*" (4).

Within the field of agent-based modelling, the belief-desire-intention (BDI) (25) architecture has been widely used. The BDI paradigm approaches the reasoning of agents from a rational perspective. Rational in this sense means that an agent performs those actions that takes him a step closer to its goals, given information from its environment. Beliefs, in this architecture, are an internal representation of information about the environment of the agent. Desires are states that the agent wants to achieve. Lastly, intentions are actions that are part of a plan. Taken this three concepts together, agents perform actions, based on plans, to further their goals. An example of an implementation and a specific used methodology of developing such a multi-agent system can be found in (26).

2.3.1 Multi-agent system

MAS is the technique used in this thesis to approach the problem. A MAS is a collection of multiple autonomous agents, each acting towards its objectives while all interacting in a shared environment, being able to communicate and possibly coordinating their actions (5). A MAS could be defined as a network of problem solvers that work together to reach achievements that could not be reached individually (6). The essential characteristics of a MAS is the interaction between the agents. From this interaction, interesting behaviour could emerge (5). This interaction could take place by communication, norms and roles (7). The advantages of MAS have been mentioned by numerous authors. (8) mentions modularity, parallelism and scalability as some advantages. Others arguing the explanatory power or the metaphors for the way people conceptualize and implement many types of software (6). Most industrialist are interested in agents that tackle problems in planning, resource and decision making, diagnostics, control and real-time replanning and simulation and modelling (11). For an overview on MAS see (9, 12).

2.3.2 Simulation

Simulation has proven to be a valuable approach for various research purposes and for several issues at ProRail (33, 34). In an multi-agent based simulation a researcher explicitly describes the decision processes of simulated actors at the micro-level. Interaction among those agents and their environment results in the emergence of structures at macro level (10). Simulation in MAS is a suitable technique to study the effect of different scenarios on a given output, like different situations in demand and supply in the energy market (10).

More close to this thesis' topic on disruption management, MAS simulation has been applied to streamline emergency services in car accidents (13), in disaster management, in which the allocating of resources is simulated to handle the effects of, for instance, earthquakes and floods (14, 16) or simulating human decision making for evacuation scenarios (15). In the latter, the impact of several factors, like demographics, number of police officers and information sharing via speaker are tested on evacuation performances. This characterizes the power and possibilities of MAS simulation.

2.3.3 Prognosis and Forecasting

As described in the previous subsections, estimating the function recovery time at ProRail is an important issue. MAS could be applied for the purpose of generating prognosis or forecasting about future events. A common approach for making prognosis in a MAS is by using a Bayesian network. A MAS could be enriched with a Bayesian network to measure the probability of several hypothesis. The hypothesis with the highest probability is considered as the final prognosis (17). This general principle could be applied to many different purposes, like medical prognosis (18) or control prognosis in manufacturing organizations (19).

MAS are not usually regarded as forecasting tools. Although, MAS could be used to make forecasts about future events (20,). Forecasting in MAS simulations refers to the prediction of the value of a quantitative variable based on known past values of that variable or other related variables. In (20) a guideline for forecasting with MAS is presented. In this guideline, forecasting is realized by using sources of data on which simple statistical methods are perform that are integrated in the MAS. This approach, of using additional modules to enrich a MAS, is part of this study.

2.3.4 Developing Intelligent Agent Systems

The MAS paradigm is making its way from academia to industry more and more (23). Previous agent-orientated programming languages serve as an obstacle to develop MAS in industry. In the past decades, various programming languages and frameworks have been proposed to support the development of multi-agent systems. Some programming languages extended well-known standard programming technology such as Java (e.g. Jade and Jack). The languages are different in the sense that each language is specialized in some features of MAS. In (24) an initial Java library of object-oriented design patterns for MAS concepts and abstractions is presented. This library, called OO2APL, has the aim to implement autonomous agents and multi-agent systems directly in Java Programming language. OO2APL is suitable language to develop MAS in industry since it is an object-orientated library which uses the BDI framework to model multiple agents. For this reasons, I will use OO2APL to build and simulate the disruption management as a MAS.

I will describe some general components of OO2APL, which play a key role in the MAS that is developed for this thesis. To start with, the context. A context is anything that an agent uses to make decisions and act in its environment. The context of an agent is exposed to relevant methods for gathering information and executing actions. In agent terms, the context could be viewed as a belief base. Secondly, a plan scheme. A plan scheme specifies when a certain plan is relevant and applicable given a trigger and a context of the agent. A plan scheme could be a message plan scheme, which makes it possible for an agent to send a specific message to a receiving agent. Next, a trigger. A trigger is anything that can trigger a plan for an agent. Triggers makes it possible to start the right action at the right time. These three components form the most important building blocks of developing a MAS in OO2APL.

2.3.4.1 The Prometheus approach

Prometheus is a general purpose methodology for the development of software agent systems. Although several methodologies have been proposed, this is arguably the most mature. Prometheus is intended to be viewed as a set of guidelines, which should be interpreted by the user own common sense. The core of this methodology consist of three phases:

1. The *system specification phase* focuses on identifying the goals and basic functionalities of the system, along with inputs (percepts) and outputs (actions).
2. The *architectural design phase* uses the outputs from the previous phase to determine which agent types the system will contain and how they will interact.
3. The *detailed design phase* looks at the internals of each agent and how it will accomplish its tasks within the overall system.

In this thesis I will use these core ideas of Prometheus in combination with my own common sense and insights, to develop a MAS simulating the disruption management at ProRail.

2.4 Conclusion

In this chapter I have given an overview of the various aspects of the disruption management system. In line with these aspects, I discussed topics of the MAS paradigm that are in close relation to the research question. In the following chapters, I will go through the steps in answering the main question of this thesis.

3 Methodology

In the previous chapter I have described the working of the disruption management system. In this chapter I will clarify and motivated the process of modelling the disruption management system as a multi-agent system. According to *A Framework of effective modelling* (35), a model is an idealized, simplifying and with respect to certain aspects similar representation of an item, system or some other part of the world. The purpose of the model is to allow a better study of specific properties than using the original system. According to the Prometheus approach (30), the first phase of developing a MAS is looking at the system specification. For this thesis, I will give a specification of the system goals, specifying the agent types and the interactions among agents.

3.1 Goal of the disruption management system

The disruption management system at ProRail has the goal to minimize the time the infra is not available, due to disruptions. Therefore, the systems operates as fast and safe as possible.

To act fast and safe, the operation has to be reliable, accurate and creative, according to ProRail guideline for disruption management. In line with this way of operating, the participating parties have a clear defined role in the disruption system. Each role share a common sub-goal, namely, making sure that information is reliable and correct. Parties have to gained relevant information and properly share this among each other. Information, in this sense, could be anything sort of knowledge that is shared among the agents. For instance, facts about the situation, estimations about the time of arrival or the time a function will be solved. To achieve this, parties perform checks to be sure that information is reliable. Moreover, they have to reason about the situation to make sure that the estimations they make are as good as possible. In Table 1, there is a list of sub-goals that contributes to the main goal.

Main goal
Minimum hindrance of time infra is not available
Sub goals
Making RVO
Choosing priority level
Determining initial prognosis
Sharing estimated time of arrival
Approving BUTA
Finding failure cause
Brining equipment
Making prognosis
Repairing infra function

Table 1: The systems' main goal and sub goals.

These sub-goals are linked to the parties of the disruption management system. The system works on the basis of sharing information, coming from these different parties. Each parties plays a role in the whole disruption management and in estimating the function recovery time. Therefore, the disruption management system could be viewed as a distributed system. In the field of distributed problems, multi-agent systems is a suitable and effective approach for solving these kinds of problems (8).

3.1.1 Belief-Desire-Intentions Framework

In a distributed system, information is shared among parties, on which actions and decisions are based. Each party has its own sub-goal, related to its role in the disruption management. In the light of these goals, parties collaborate to solve the disruption as proper as possible and make the best possible function recovery time estimation. To accomplish their goals, they interact with each other. In this interaction, parties revise and adjust their beliefs in the disruption process. Towards a parties' goal and according to their beliefs, actors perform well-defined actions. In this mechanism, a party has some sort of mental state, which depends on their goals and interaction with other parties. The disruption management system behaviour fits the BDI framework. In the sense that each party has beliefs, which correspond to information that the parties has about the world. For instance, the information that is stored in the beliefbase of the Contractor by receiving the RVO. Desires represent states of affairs that the party would wish to be brought about, for instance, sending the ETA to the MKS. Finally, intentions represent actions that are part of a plan, which an agent performs. This way of framing a party is in line with the existing studies on BDI and support application described in section 2.3.2. Moreover, a BDI framework could be appropriately extended with algorithmic modules to enrich the model (18, 20).

3.2 From actor to agent

In this subsection I will descript the modelling of the parties involved in the disruption management system as agents. According to Prometheus, the next step in developing a MAS is specifying the agent types, what the agent should be and the tasks an agent should have. For this thesis, I will give a description of each agent and define the agents' relationships. Each party described in the background section can be modelled by an agent whose task is to perform certain actions. I will describe each agent separately.

3.2.1 Meldkamer Spoor agent (MKS)

The MKS can be modelled as an agent that is responsible for managing the function recovery process. The main goal of the MKS agent is to assist the other agents as appropriate as possible. According to this goal, the MKS agent has several tasks to perform. In the start of the disruption process the MKS agent is involved in making a RVO. The RVO is build up in cooperation with the Dispatcher. The priority level of a disruption is determined in collaboration with the Dispatcher. Depending on the prognosis, diagnosis, priority, time of the day and other factors, the MKS agent and Dispatcher agent decide whether the problem needs to be fixed immediately or at another time. This decision has on an effect on the estimated FRT (22). Furthermore, the MKS agent checks if other agents perform their tasks on time. So the MKS agent is specialised in giving deadlines and checking of those are achieve. Finally, the MKS has the important task to share information among different agents. The MKS agent sends the most recent information about the process to relevant agents. In Table 2 a descriptor of het MKS agent is presented.

Name: MKS agent.
Description: Makes RVO, sourcing information, manages process.
Lifetime: Instantiated on Dispatcher messages on making RVO. Demise at the time function is repaired.
Goals: Making RVO, determining priority level, collecting and sharing information, asking for FRT prognosis, checking deadlines.
Percepts responded to: Incoming RVO message, initial prognosis, knowing failure cause.

Table 2: Descriptor of the GL agent

3.2.2 General Leader agent (GL)

The GL can be modelled as an agent that is in charge during the operational process of repairing the function. The GL agent has the goal to let the function recovery process run as safe and smooth as possible. Therefore it closely communicates with the agents involved in the disruption management. In particular, the MKS agent and the Contractor agent. His first task is to send his ETA to the MKS agent. The GL agent and Contractor are both presented at the location of the failed infra object. In light of his goal, the GL agent is aiming to determine an estimation of the function recovery time. To make a reliable prognosis, the GL agent communicates closely with the Contractor. In Table 3 a descriptor of the GL agent is presented.

Name: GL agent.
Description: Manages operational process, adjusts prognosis.
Lifetime: Instantiated on request for sending ETA. Demise at the time function is repaired.
Goals: Sending ETA, approving initial prognosis, discussing prognosis, checking work Contractor
Percepts responded to: RVO with initial prognosis, information about available equipment.

Table 3: Descriptor of the GL agent.

3.2.3 Contractor agent

The Contractor agent has the goal to repair the failed object as soon as possible. To achieve this goal, several tasks have to be performed. After sending his ETA, the Contractor has to search for the cause of the failure. Quite often this takes some time. During the process, the Contractor agent determines in cooperation with the GL agent the estimated failure repair time. This FRT prediction depends on the availability of equipment that are required to repair the failure and de duration of the repairing process itself. In Table 4 a descriptor of the Contractor agent is presented.

Name: Contractor agent.	
Description: Repairs failed function, discusses prognosis.	
Lifetime: Instantiated on request for repair. Demise at the time function is repaired.	
Goals: Sending ETA, finding failure cause, bringing equipment, discussing prognosis, repairing failed function	
Percepts responded to: RVO, approved BUTA, request for (final) prognosis.	

Table 4: Descriptor of the Contractor agent.

3.2.4 Dispatcher agent

The Dispatcher agent is responsible for safety of train traffic in his/her own designated area. Therefore, the Dispatcher agent is able to regulate the rail infra objects. From this position, the Dispatcher agent could adjust the train traffic. For instance, let trains switch track. In the disruption management, the Dispatcher agent plays a role in making a RVO. Furthermore, the Dispatcher agent is in charge of approving BUTA, requested from the Contractor agent. In Table 5 a descriptor of the Dispatcher agent is presented.

Name: Dispatcher agent.	
Description: Makes RVO, approves BUTA.	
Lifetime: Instantiated on incoming failure notification. Demise at the time function is repaired.	
Goals: Making RVO, determining priority level, making and approving BUTA,	
Percepts responded to: Failure notification, ETA of Contractor.	

Table 5: Descriptor of the Dispatcher agent.

3.3 Relationships among agents

Once the agent types are decided, the next aspect of the architectural design is to specify the interaction between agents, capturing the dynamic aspects of the system (30). Since the role of the agents are clear, formulating their relationships is another import step in the modelling process. In the disruption process, the relations agents share can be quite broad. For the topic of this research, the focus lays on the relationships among agents that are relevant for the predicted FRT. Figure 2 below shows schematically the relationships among agents in the FRT predicting process. The figures contains the participating agents and their life lines. In the figure, all the relevant communication, information units and decision making are presented. Relevant in this sense means that the play a role in in estimating the function recovery time. A line with double arrows means that agents actively interact with each other. Two agents, namely the GL and the Contractor, interact in an iterative process, as could be seen in the Figure. The iterative process of asking for the current prognosis and sending the current prognosis terminates if the prognosis is shorter than 30 minutes. In de following, a concise description of the active interactions (e.g. double arrowed lines) will be given.

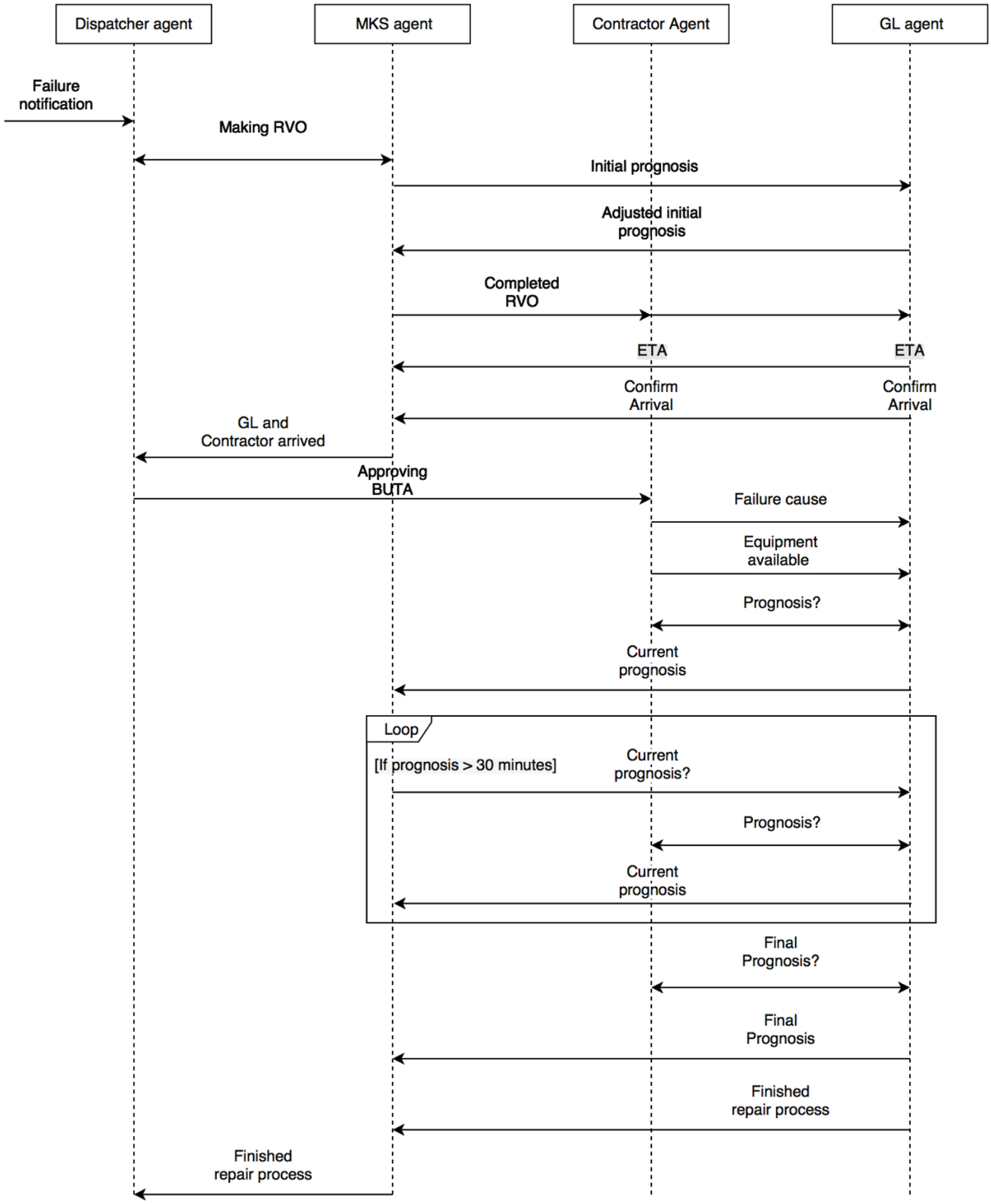


Figure 2: Overview of the aspects and their relations in the disruption management system.

3.3.1 Handling incoming disruption notification

The most important aspect of the disruption anamneses is to formulate an appropriate RVO with a proper priority, so the Contractor agent is well-informed, which benefits the recovering process (36). In this case, the question raises what an appropriate RVO is. In general, a RVO consists of the name and code of the failed object, a description of the disruption, generic location and a priority level. The

MKS agent is responsible for asking the right questions to the Dispatcher, to realise an appropriate RVO. For the scope of this thesis, the MKS agent is interested in two issues, see Figure 3.

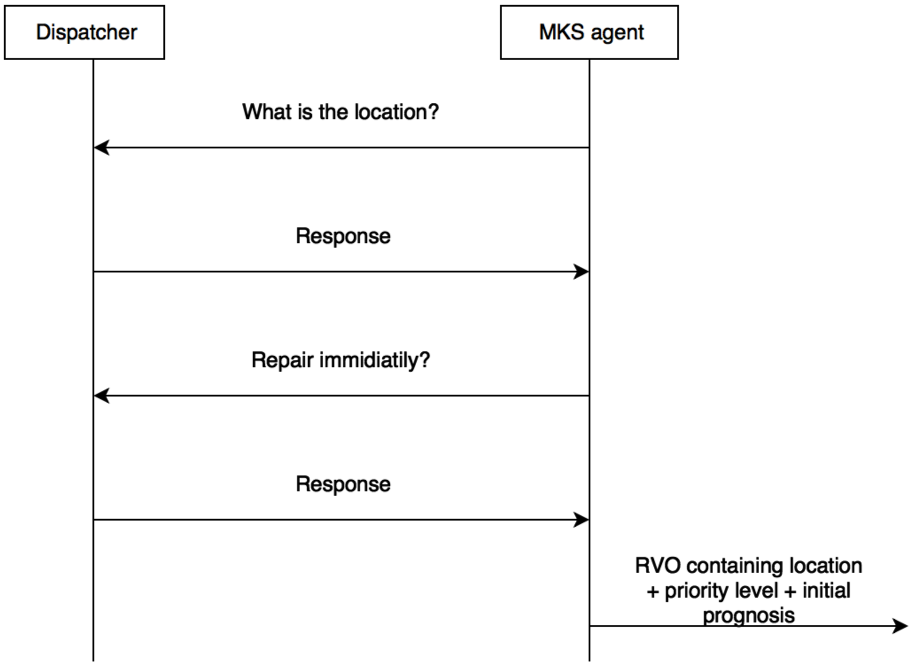


Figure 3: Interaction for making a RVO.

These two questions cover the most crucial information in relation to the recovery time. Namely, an exact location makes the ETA of the GL agent and Contractor agent more reliable. The answer on the second question covers the level of priority. If it is necessary to repair immediately, the priority level is set to level 2.

3.3.2 Prognosis after diagnosis

The second active interaction takes place right after finding the failure cause. The GL agent and the Contractor agent discuss which each other in what proportion the prognosis has to be adjusted. This is done by sending a message from the GL agent to the Contractor agent by asking which influence the specific cause type has on the length of the prognosis. Depending on the cause type, the prognosis is either made longer or shortened by the Contractor.

Onwards, during the repairing process, the GL sends requesting for the current prognosis to the Contractor. This process of requests and responses can iterate several times, according to the time is takes to repair the function and the amount of new information that is gained during the process. The iterations stops if the Contractor has enough information to be sure that the repairing process will take less than thirty minutes from this point, the current prognosis is transformed to a final prognosis, which results in finishing the repairing process.

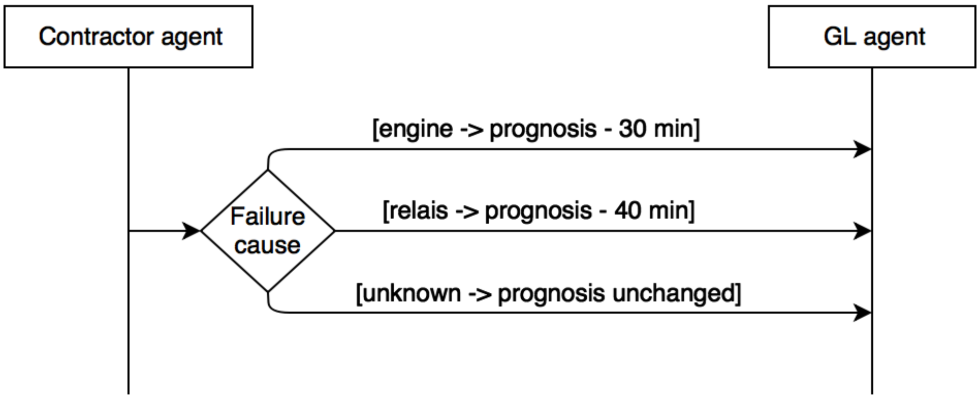


Figure 4: Decision making for failure cause and corresponding adjustment on prognosis.

3.3.3 Modelling failure repair time prediction

An important part of the systems behaviour is the process of predicting the function recovery time. As described in section 2.2 the initial prognosis has a value that is generated automatically. Onwards, this prognosis flows through the system and is adjusted based on reasoning with new information that becomes available during the process. Depending on the information and the decision that an agent makes, the current prognosis is either made longer or shorter. Factors of different kinds could result in an adjustment of the prognosis.

3.4 Improving a MAS

The first sub question of this thesis is about the way the disruption management could be modelled as a MAS. If this step is taken, the second sub question, on improving the MAS, becomes the next step. Extensions in MAS exists in various forms. In (27) a basic BDI architecture is improved by using a neural network, which results in a hybrid BDI agent model that could handle many different types of activities in a container terminal. In this model, agents have, beside simple plans, complex plans that could be viewed as a tuple and consist of many components. Using such complex plans in agents makes it possible to let them make decisions in more complex environments (27). Using data-driven algorithmic modules to improve the performance of a MAS is a common approach. Another common technique to enhance a MAS is data mining, for instance, using sequential pattern mining to derive prediction rules about what actions or situations might occur if certain predictions are satisfied (28). Moreover, to make decision-making in a MAS more human-like, models could be enhanced with several descriptive models of decision making from psychology (29). So, there are various methods for improving a MAS. The principle of using a baseline model and enriching it with a broad range of extensions, forms the basis of the approach I will use in this research.

3.4.1. List of extensions

Several studies at ProRail have been conducted with the aim to gain more insight in the effect of possible enhancements in the disruption management system (22, 36). These studies resulted in the development of algorithmic modules, adjusted communication protocols or recommendations to extend the actions a party has to perform. A list of extensions could be found in Table 6 below. I will briefly discuss each extensions and give a motivation why some extensions are being part of the MAS of this thesis and others don't. Especially overlap between extensions is a reason to pick one and not both. Moreover, I will clarify where an extensions will be integrated in MAS and how this could enhances decision making.

Type	Extensions	Profit
Data Driven	Delft PhD on Bayesian Network and FRT	More accurate prognosis given some characteristics of the failed infra object
	ProRail DataLab switch failure cause algorithm	Well-informed cause search performed by Contractor
	BUTA length/Type of Switch	The longer the BUTA, the longer the Contractor has to find the failure cause
	ETA calculated by Google Maps*	More accurate ETA that take the traffic situation into account
Semi Data Driven	Time of the day*	During rush hour the repairing process could be postponed which results in a later moment of repairing the function
	Extended RVO	More specific disruption location
Internal communication	GL checks and shares information in a more proper way	No decision made on misleading information
	GL confirms initial prognosis after have spoken with the MKS	Initial prognosis will be more accurate therefore train traffic will be able to drive right after function is repaired
External factors	Weather	In extreme weather condition, the repairing process is slowed down

*This extensions is not part of the enriched MAS

Table 6: List of extensions that aims to enrich the baseline MAS

3.4.1.1 Bayesian network

The first extensions is developed in a PhD project a TU Delft (31), which used a Bayesian Network to predicted the disruption length. The prediction of the disruption length is split in two components, namely the latency time, this is the time slot between the notification of the disruption and the moment the Contractor has arrived at the location of the disruption. The Bayesian network takes the location, weather and cause type of the switch into account regarding the latency time. The second component is the repair time, so the time it takes from the moment the Contractor arrived at the location and enters the rail track to repair the failed infra object. This algorithm is developed on a sufficient dataset and has proven to perform well (31). Therefore, this extension will be part of the MAS in this thesis.

This extension will be integrated in the beliefbase of the MKS. Namely, the MKS has all the information available that is needed for the Bayesian network to generate a prediction. In the MAS, the MKS has the information at hand and is capable of sending a request to the network. In return, this network, modelled as an artefact, responds to the request by sending a prediction of the function recovery time. This prediction for the Bayesian network becomes the initial prognosis that the MKS adjust based on other extensions, after which it is send to the GL. In this situation, the initial prognosis is still automatically generated, namely by an algorithm. Although automatically generated, this is an extension of the initial prognosis since the Bayesian network takes more factors into account than the current calculation. The current calculation for determining the initial prognosis only takes the average time of the disruption length of a specific infra object.

3.4.1.2 Predicted failure cause algorithm

A second extension is developed at the DataLab of ProRail. A team of data scientists has developed an algorithm that predicts the type cause of a switch failure. This is valuable information for a Contractor. If a Contractor knows the failure cause before entering the rail track, he is capable to search for the failure cause in a well-informed way. Having this information by forehand available will result in a faster cause search and benefits the time to repair the function. Due to this clear benefit of the algorithm I will use this to extend the baseline model.

This extension will be integrated in the belief base of the Contractor. The Contractor is responsible for discovering the failure cause by entering the rail track and start searching. The Contractor will be capable of using this extension by sending a request for the predicted failure cause. The switch failure cause algorithm, modelled as an artefact, will respond with the estimated cause of the failed switch. This additional information would result in sending an extra prognosis by the Contractor, namely right after receiving the predicted failure cause and before entering the rail track. This improves the model because a prognosis based on the failure cause is send earlier in the process. Knowing a correct prognosis earlier in the process increases the possibilities for participating agents to anticipated on this information, according OPOZ. I will not elaborate on the content of this extension, since this lays out of the scope of this thesis.

This extension would result in enhancing the function recovery time, so the process of repairing a disruption. Since the extended MAS has the aim to better predict the FRT, the influence of this extension of the FRT itself is predicted. Therefore, this extension is part of improving the prognosis in the extended MAS.

3.4.1.3 Type of switch failure algorithm

The third possible extension is about predicting the time a Contractor needs to repair a failed infra object corresponding to the specific type of this infra object. This difference from the cause type of the Bayesian network, that only takes this into account to predict the latency time. The switch failure algorithm takes the repair time into account. A master thesis at ProRail has made clear through data analysing that there exist a relation between recovering time and the type of failed switch (22). See Table 7 for the results.

Type of switch:	90 % repaired (min)
NIC - Regular	112
NIC - Diamond	87
Switch disturbed	114

Table 7: The time in minutes at which 90 % of specific type of switch failures are repaired

If a regular switch is NIC, after 112 minutes of repairing, 90 % of these types of disruptions will be repaired. For a diamond switch that is not in control and a disturbed switch, this is 87 minutes and 114 minutes respectively. This forms valuable information for making a decision about the prognosis after the type of switch is known.

The type of failed switch is known at the very beginning of the disruption process. So, the MKS could, besides the Bayesian network, takes the information about the type of switch into account to make a decision about the initial prognosis. This extension will be implemented by an artefact which could be requested by the MKS agent.

3.4.1.4 Determine ETA by route planner

The estimated driving time of the GL and Contractor could be predicted by using a route planner. Using a sufficient route planner that takes the current traffic situations into account would lead to an accurate estimated time of arrival. This accurate prediction benefits the disruption process in the sense that the first prognosis of the GL is partly based on this estimation. As discussed earlier, the first prognosis is an important concept in the MAS. Although, in real life most of the time the GL and the Contractor use a sufficient route planner already, so this makes the choice to use it as an extension less convincing. Therefore, this route planner will not be used as an extension.

3.4.1.5 Extended RVO

An adjustment in the process of making a RVO could improve the disruption management (36). In this master thesis at ProRail a fault tree and event tree analysis showed that more specific information in a RVO leads to a faster disruption process. In the light of predicting the function recovery time, applying this technique would assumingly result in a more accurate estimation in the MAS.

An extended RVO is established by asking more specific questions between the MKS and the Dispatcher. There is one key aspect that results from these additional questions, namely a more precise location of the disruption. Knowing the location more precisely, will result in a more accurate ETA of the Contractor and GL. Due to the chain-like structure of the system, this will result in a more accurate prognosis. Since the clear benefit of this extension and the feasibility of implementing it in the baseline model, this extension will be integrated in the baseline MAS.

This extension would result in enhancing the function recovery time, so the process of repairing a disruption. Since the extended MAS has the aim to better predict the FRT, the influence of this extension of the FRT itself is predicted. Therefore, this extension is part of improving the prognosis in the extended MAS.

3.4.1.6 Time of the day

During rush hour, the number of people travelling by train is maximum. If a disruption occurs in this period of time, the impact is higher compared to other moments of the day. Giving accurate prognosis in these moments of the day is favourable. Rush hour in the rail world means on the same time rush hour on the road. Busy traffic could hinderance the driving time of the GL and Contractor. So taking the rush hour into account in estimating the FRT is valuable. However, rush hour is also part of the Bayesian network extension discussed firstly. So the influence of rush hour on deciding the estimated function repair time is already taking into account in the first extension. Therefore, this extension, on its own, will not be part of the enriched MAS.

3.4.1.7 Extended tasks of GL

OPOZ, a research team at ProRail, recommended to improve the quality of GL by adjusting their training, tasks and responsibilities. By evaluating major impact disruptions, the research team has found that a GL would improve the process if he contacts the MKS after receiving the automatically generated initial prognosis. During this moment of communication, the GL could decide to adjust the prognosis. Assuming that an adjusted prognosis would be more accurate and has a beneficial effect on the system as a whole. For this reason, the communication between the MKS and GL, after sending the first initial prognosis, will be explicitly added to the baseline MAS.

A second extension for the responsibility of the GL, that is recommended by OPOZ, is performing checks to validate information. In the past, it turned out that parties sometimes understand shared information differently from each other, without being aware of it. As a consequence, decisions are made on misinterpreted information, which as well leads to mistakes in predicting function recovery time. This extension will be realised in the MAS by adding check-actions performed by the GL. These checks have the aim to be sure that agents understand and therefore work with the same information. This would lead to prognosis that are based on validated information.

This extension would result in enhancing the function recovery time, so the process of repairing a disruption. Since the extended MAS has the aim to better predict the FRT, the influence of this extension of the FRT itself is predicted. Therefore, this extension is part of improving the prognosis in the extended MAS.

3.4.1.8 Weather influence

Repairing a rail infra object is mostly done outside and therefore, has to deal with weather conditions. The weather type could influence the repairing process, performed by the Contractor. Especially heavy weather conditions, like extreme rain falling, snow, temperatures below zero degrees or above thirty degrees, will probably play a role in the time it takes to repair the function. Taking the influence of weather into account in determining the prognosis, will extend the baseline MAS.

This extension will be added to the decision making process of the Contractor, since this agent has to deal with the weather conditions the most. It is assumed that the extreme weather conditions just described, will delay the repairing process. So in determining the prognosis, the Contractor will in case of an extreme weather condition, increase the estimated function recovery time. Otherwise, the Contractor will leave the current prognosis unchanged.

3.5 Agent architecture

As mentioned earlier, the aim of the extensions is to improve the quality of the estimated FRT. In Figure 5, an overview is given of the extensions that adjust the prognosis. The extensions are queried from the belief base of the agents. In OO2APL terms, the extensions are queried from the context of the agents. In sections 4.2.4, the specific settings of each extension is described. It depends on the setting of the experiment which extension is queried and which not. The queried information from the extensions is used in the belief base of the agent to decide on the adjustment of the prognosis. The adjustment is a certain amount of minutes that is either added or subtracted from the current prognosis. The exact amount of minutes that forms the adjustment could be found in section 4.2.4. In the results section, the selection of used extensions is discussed in more detail.

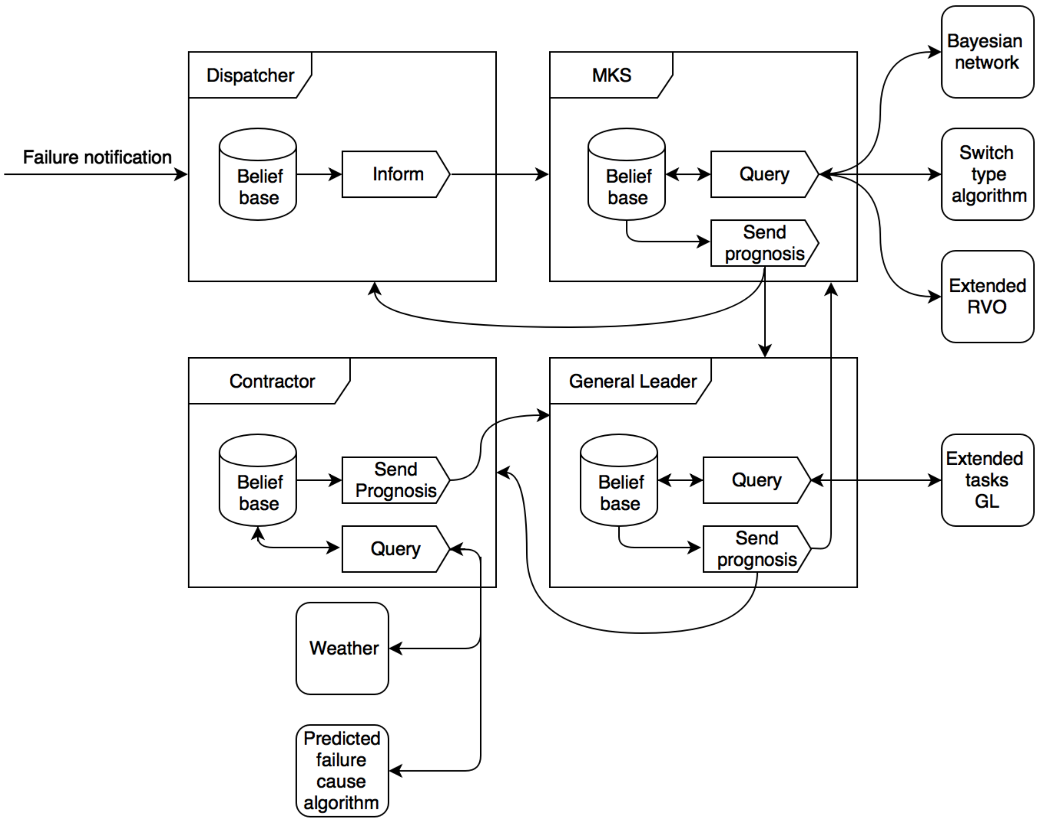


Figure 5: Overview of the queries that agents send to the extensions. The agents queries the extensions from their belief base. Within their belief bases, the agents adjust the prognosis accordingly. The agents are capable of sending the prognosis to each other.

3.6 Performance measurement

Running MAS simulations should result in some sort of performance. The output of a simulation in this research is expressed by the prognosis given in minutes. For this thesis, I will run several scenarios in the baseline model, that should result in a prognosis in minutes equal to the prognosis in real-life. The baseline model is in this sense a perfect simulation of real-life scenarios. For the extended model, the output is also a prognosis in minutes. The performance of the extended MAS is determined by calculating the difference between the extended MAS prognosis compared to an optimal prognosis. If this difference is smaller than the difference between the original prognosis compared to the optimal prognosis, the extended MAS prognosis is performing better. The difference could be calculated by using a statistical measure.

3.7 Conclusion

Approaching the problem of the disruption management process from a MAS perspective is done step by step. The Prometheus approach provides a useful method to build this steps on. All the necessary information about the system's specifications, agent characteristics and extensions have been determined. The chosen performance measurement provides solid ground to check to quality of the extended MAS prognosis.

4 Experiments and Results

In this chapter, I will motivate the steps to are involved in the MAS simulations experiments, the results from these experiments, consider validation issues and end with a discussion. Therefore, I will clarify the way I have developed the baseline model and the extended MAS. Moreover, I will present the simulation results and elaborate on them. Onwards, I will give an evaluation on the results and discuss the findings and research issues.

4.1 Introduction

Before going into the simulation experiments and corresponding results, I will briefly give a description of the steps involved in the simulation process. At the start, I have chosen five disruptions that have taken place in the past. These disruptions, which I call scenarios, form the input of the simulations. Onwards, a baseline model is built which simulates each scenario deterministically. In the next step, the baseline model is extended with modules that have the aim to improve decision making on the prognosis. Lastly, the output of the extended MAS simulation is compared with the baseline MAS simulation. The results of this comparison gives insight in the performance of the extended MAS.

4.2 Experimental setup

In this section I will describe the setup of the simulation experiments. Therefore, I will motivate which scenarios are part of the MAS simulation. Onwards, I will describe the baseline MAS and the extended MAS that are developed for this research.

4.2.1 Design of scenarios

For each disruption at ProRail, relevant information is collected and stored at a database. From this database full of historical disruptions, I have chosen five disruptions that forms the baseline in the simulations, called a scenario. A scenario consists of a set of information that contains all the decision making and information sharing that has taken place in the disruption management system for that specific case.

The five scenarios that are part of this study are all related to switch disruptions. Since one of the extensions, the predicted switch failure cause, is only applicable for switch disruptions. The scenarios have taken place on the Dutch railway. More specific, at Amsterdam, Dordrecht, Zevenaar and two at Utrecht. I have chosen these five scenario for several reasons. Firstly, the scenario has to contain enough information about decision making and sharing information as represented in Figure 2 in section 3.3. This is important, since a scenario has to simulate a real life case as proper as possible. For many disruptions, the data of ProRail lacks information to make it possible to construct a proper scenario. The data of disruptions is mainly stored manually by human operators. Which is a reason behind the fact that the disruption dataset that is largely incomplete. Since important information is missing in the data, searching for essential information to construct a scenario is a time consuming activity. In the light of this time consuming activity, I managed to construct five scenarios. Second, the time on which a prognosis is shared has to be clear. Having information about the time a prognosis is given and knowing the final recovery time, makes it possible to evaluate the quality of the prognosis. Not for all disruptions information about the time on which prognosis are given, is presented in the data. The five scenarios that are part of this study contain information about the time on which a prognosis and a final prognosis is given. Furthermore, according to expert knowledge, there exists a great variety in the quality of prognosis related to switch disruptions. In general terms, for switch disruptions, prognosis are sometimes pessimistic, somethings optimistic and in some cases relatively accurate. Therefore, I have chosen to pick scenarios in which the original prognosis was either too pessimistic, too optimistic or generally accurate. The five scenarios cover the variety of the quality of prognosis. Moreover, in the five chosen scenarios the parties followed a regular protocol. In this sense, the five scenarios are prototypically, since they represent the broad variety of the quality of prognosis and followed the regular protocol of switch disruptions.

Knowing the accuracy of a prognosis forms a relevant starting point to analyse the accuracy of the prognosis generated in the extended model. In section 4.3.1 the original prognosis related to each scenario will be presented.

4.2.3 Baseline MAS

In the previous sections I have discussed the system's specifications and the architectural design of the disruption management system. The next step consists of developing the baseline model, that takes all the specifications and properties discussed earlier into account. The MAS is programmed in OO2APL which uses the BDI framework to model agents.

In the broad sense, the baseline model is a representation of Fig 2 in section 3.3. The baseline model starts by bringing all four agents alive. Although the agents become alive at the same time, each agent becomes active at a different moment. The only external trigger in the model, the incoming failure notification, starts the system dynamics, such as the communication, action execution and decision making of the agents. The baseline model mimics the chronologically behaviour of the disruption management system, in the sense that the order of executing actions in the baseline model corresponds to the real life structure. This characterizes the system's sequence as described in section 2.1.2.

The prognosis is modelled as a variable that flows through the system and is adjusted based on decisions made by agents. The outcome of the decisions are equal to the value of the prognosis as could be found in the data. The prognosis is expressed in the number of minutes it will take to finish the repairing process. The baseline model is a simplified but complete automated version of the disruption management system and uses the prognosis data from the scenarios. The decision making, communication and prognosis of the baseline MAS corresponds one to one with the scenario.

Besides deterministically simulating the five scenarios, the baseline model has to be suitable to be enriched by the extensions listed in section 3.4.1. The baseline model is developed in such a way, that extensions of different kinds could easily be added to the MAS. Namely, the capabilities and belief base of agents are feasible to be extended through possible extensions. A capability is simply a collection of plans. Once a plan is triggered, it results in executing actions. In this collection of plans, a new plan, corresponding to an extension, could be added smoothly. This benefits the process of extending the baseline MAS with additional modules. The same principle goes up for the belief bases of agents.

4.2.4 Extended MAS

The extended model is an enriched version of the baseline model. Namely, those listed in section 3.4.1. In the following, I will explain how each extension influences decision making in estimating FRT. Since prognosis are adjusted in chronologically order in the disruption management system, I will give a concise description of the integration of the extensions in similar structure.

The Bayesian network is the first extension in line, see Figure 6. This algorithmic extension influences the adjustment of the automatically generated initial prognosis, if enough information is at hand. This precondition is formulated because in real life, the Bayesian network requires enough input variables to generate an alternative prognosis (31). If enough information is available, the Bayesian network replaces the initial prognosis with a prognosis of 140 minutes. This adjustment is roughly based on a four years research project of TU Delft (31). Otherwise, the initial prognosis is unchanged.

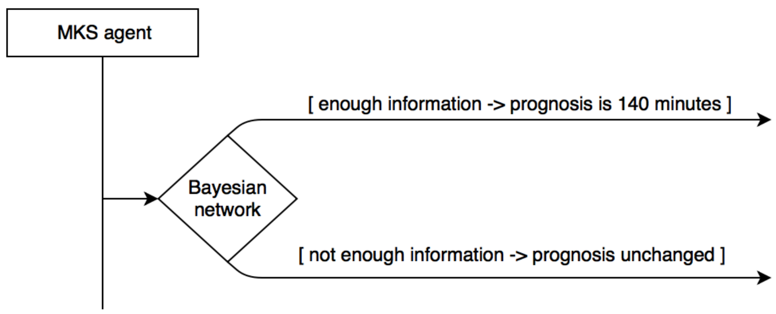


Figure 6: the Bayesian network extension

The second extension in the enriched model is the Switch type algorithm, see Figure 7. This extension is positioned at this part of the sequence, because information about the type of distributed switch is available at this point. The sooner an extension is taking into account, the better. Looking at the influence of this extension on the prognosis: based on a study at ProRail (22), the current prognosis is shortened with 20 minutes if either the disruption considers a regular switch that is not in control or if a switch is disturbed. Moreover, this extension could lower the prognosis with ten minutes if a diamond switch is not in control.

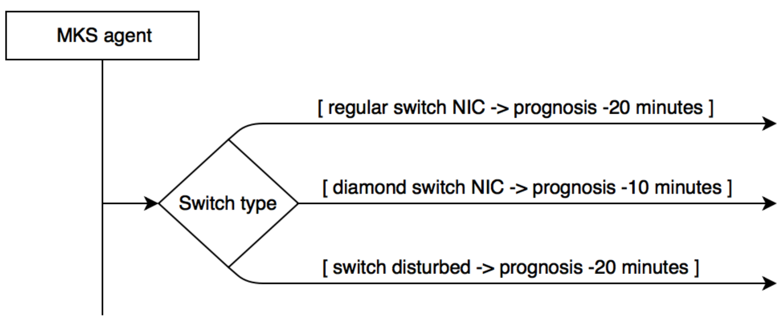


Figure 7: the Switch type extension

Thirdly, extending communication between the MKS agent and the Dispatcher agent enriches decision making on the estimated FRT, see Figure 8. Communication about the precision location of the disruption, results in a reduction of 40 minutes of the current prognosis. This reduction is roughly based on a research conducted at ProRail (22). It is assumed that knowing the precise location will result in an equal prognosis reduction for every scenario. After the third adjustment, the MKS agent sends the initial prognosis to the GL.

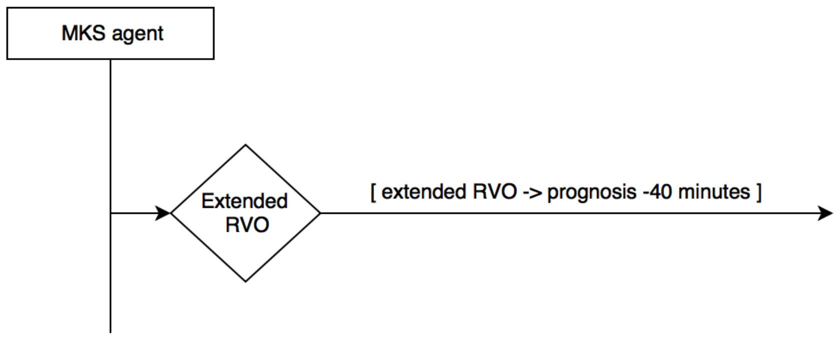


Figure 8: the extended RVO extension

An extension in the tasks of the GL forms the forth enrichment of the model, see Figure 9. Right after receiving the initial prognosis, the GL agent has the extra task to confirm the initial prognosis. Depending on the quality of the GL, a skilled GL either shortens the prognosis by 10 minutes and a unskilled GL increases the prognosis by 10 minutes. This value has been roughly estimated by a GL. Moreover, skilled in this sense means that the GL has recently passed a training which focusses on more communication with other agents. As described in 3.4.1.7 ProRail has plans to start with such trainings.

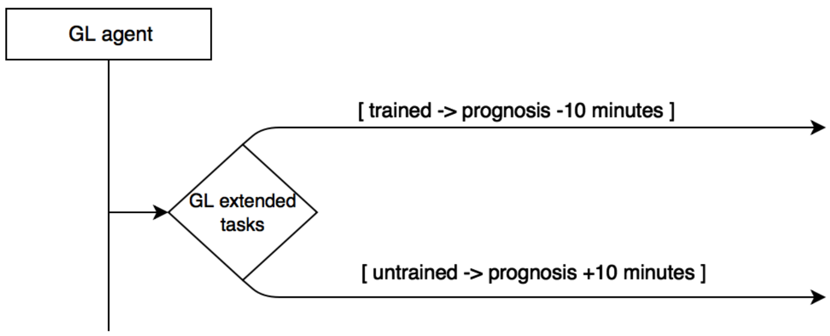


Figure 9: the extended tasks GL extension

The fifth extension that influences the current prognosis takes the weather conditions into account, see Figure 10. This extensions is positioned at the decision making of the Contractor. Since this agent has to make an estimation about the repairing time right before entering the rail track. The current prognosis is adjusted on the following way: if the weather is cold, then the current prognosis is incremented with 20 minutes. If the weather is sunny, the current prognosis is shortened with 10 minutes. Otherwise, the prognosis is left unchanged. The exact adjustment in minutes chosen here are roughly based on the PhD study described previously (31).

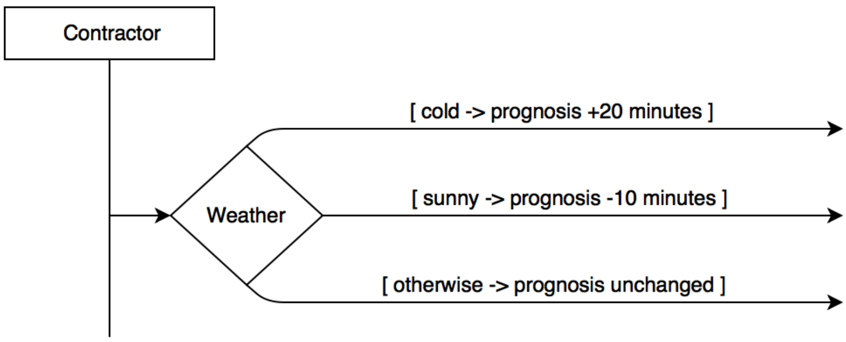


Figure 10: the weather extension

The last extension consist of the predicted failure cause, see Figure 11. Having a prediction of the switch failure cause, would result in a shorter search time, as described in section 3.4.1.2. Since this algorithm have never been used in practice, it is hard to determine the precise adjustment in minutes that this extension brings about. Therefore, an expert is consulted. Based on his expertise I have chosen to shorten the prognosis with 30 minutes if the engine is the predicted failure cause. If a relay is the predicted disruption cause, the prognoses is shortened with 40 minutes. Otherwise, the prognosis is left unchanged.

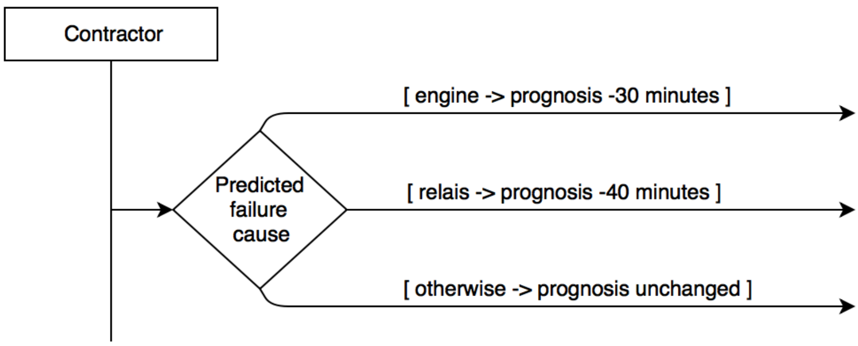


Figure 11: the predicted failure cause extension

4.2.5 Simulation experiments

In the previous section each extension for the MAS is described. The next step is running simulations experiments in the MAS in which all the extensions are integrated. Every simulation starts with an incoming disruption notification. This triggers the agents to communicate, share information and make decisions about the estimated function recovery time. The process of sharing information and making decisions is enriched by the extensions, which will lead to adjustments in the prognosis.

Each simulation has a starting prognosis that is equal to the initial prognosis of the original scenario. The prognosis is defined as the number of minutes it will take to recover the function, starting from that moment of time. The simulation experiment is setup in such a way, that every possible selection of extensions is applied to the MAS. In a simulation, the prognosis is received, adjusted and shared a number of times, depending on the number of extensions that is applied. At the moment the prognosis is less than 30 minutes, the current prognosis is defined as the final prognosis, and the system terminates.

In the experiments, three extensions are fixed per scenario. Namely, the switch type, weather conditions and predicted failure cause, since these are known by forehand and match the real situation of a scenario. In Table 8, for each scenario the settings for these three extensions are presented.

Scenario	Switch type	Weather	Predicted failure cause
1	Regular NIC	Normal	Engine
2	Regular NIC	Normal	Unknown
3	Regular NIC	Normal	Engine
4	Switch disturbed	Sunny	Relay
5	Regular NIC	Sunny	Relay

Table 8: The settings of the three extensions that are fixed in each scenario.

4.3 Results

Before going into the results of the MAS simulations, three notable concepts will be further explained. Namely the original prognosis, the extended MAS prognosis and the optimal prognosis. These concepts forms an important base for discussing the results.

4.3.1 Three types of prognosis

Firstly, the original prognosis corresponds to the estimations of the function recovery time as stored in the data for each scenario. This prognosis is equal to the prognosis generated in the baseline MAS. The extended MAS prognosis is generated during the MAS simulation experiments as discussed in section 4.4. The optimal prognosis is based on the recovery time it truly has taken to repair the function. For each scenario, the true recovery time is known. The optimal prognosis has a starting value that is equal to the complete recovery time and decrease over time. See Figure 12 for an example of the optimal prognosis. The optimal prognosis will be used in the next paragraph

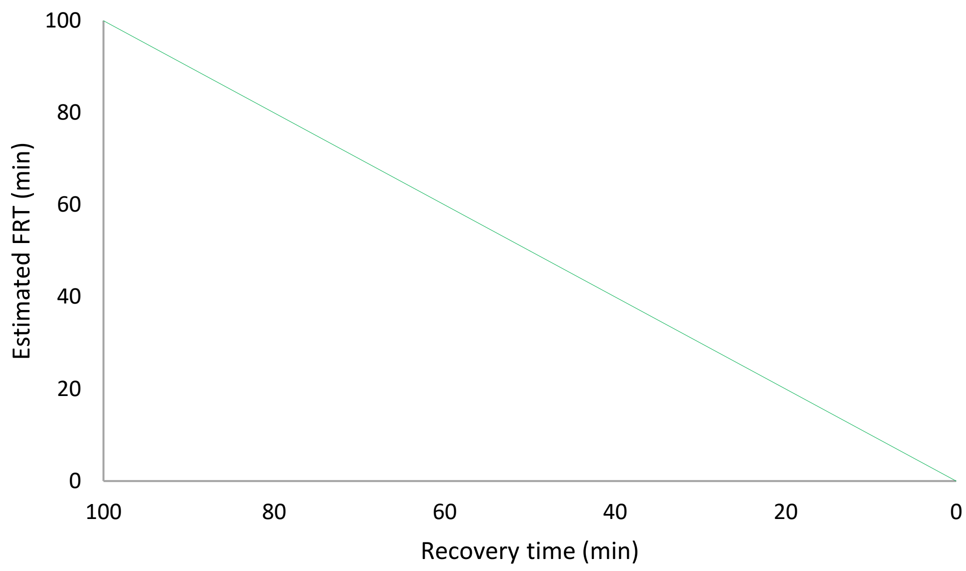


Figure 12: An example of an optimal prognosis. The estimated FRT equals the true recovery time perfectly.

The optimal prognosis is useful since it offers a measurement for the quality of the original and extended MAS prognosis. Before presenting the precise outcomes of these measurements, the original and optimal prognosis for each scenario will be plotted.

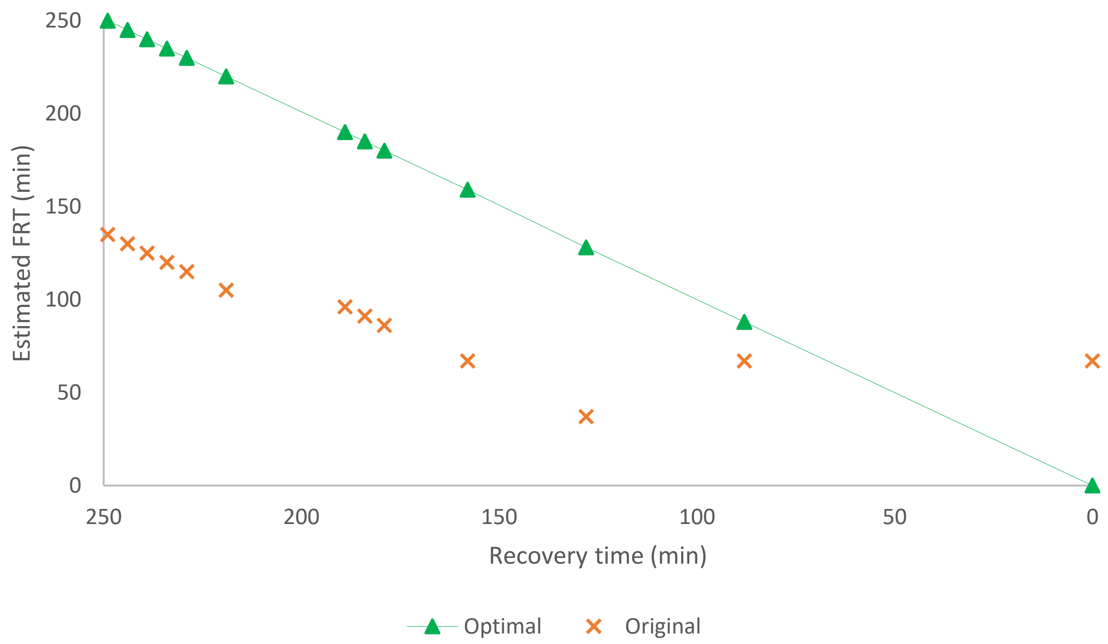


Figure 13: Scenario 1, Utrecht 4-10-2017.

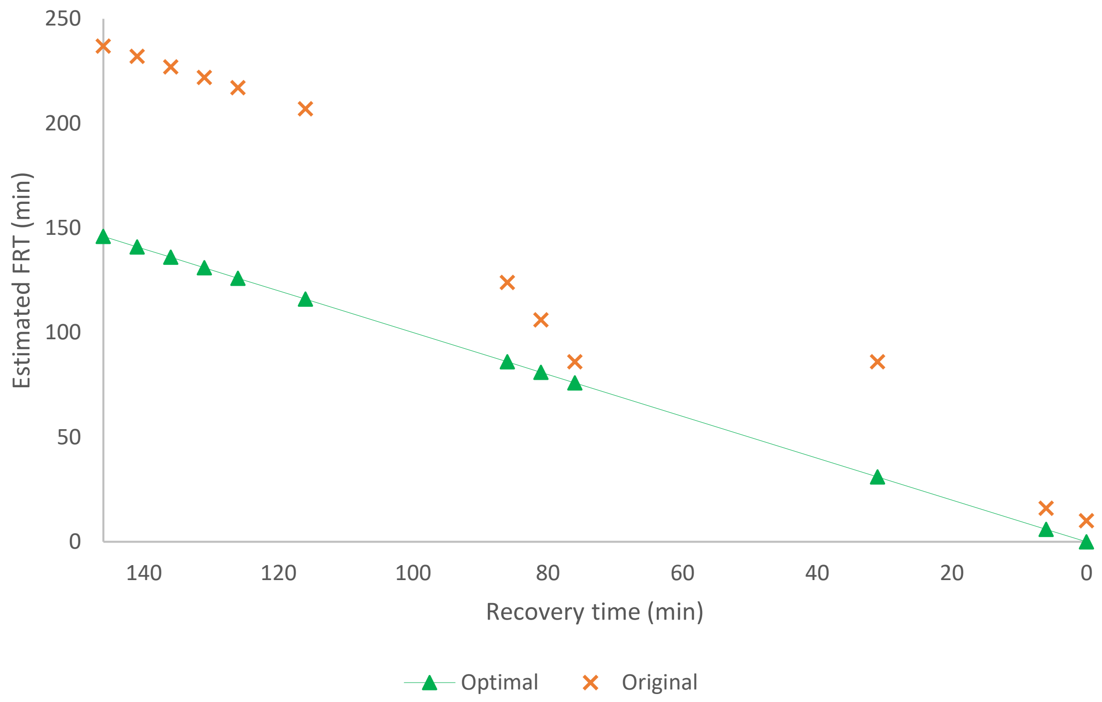


Figure 14: Scenario 2, Zevenaar 29-09-2017.

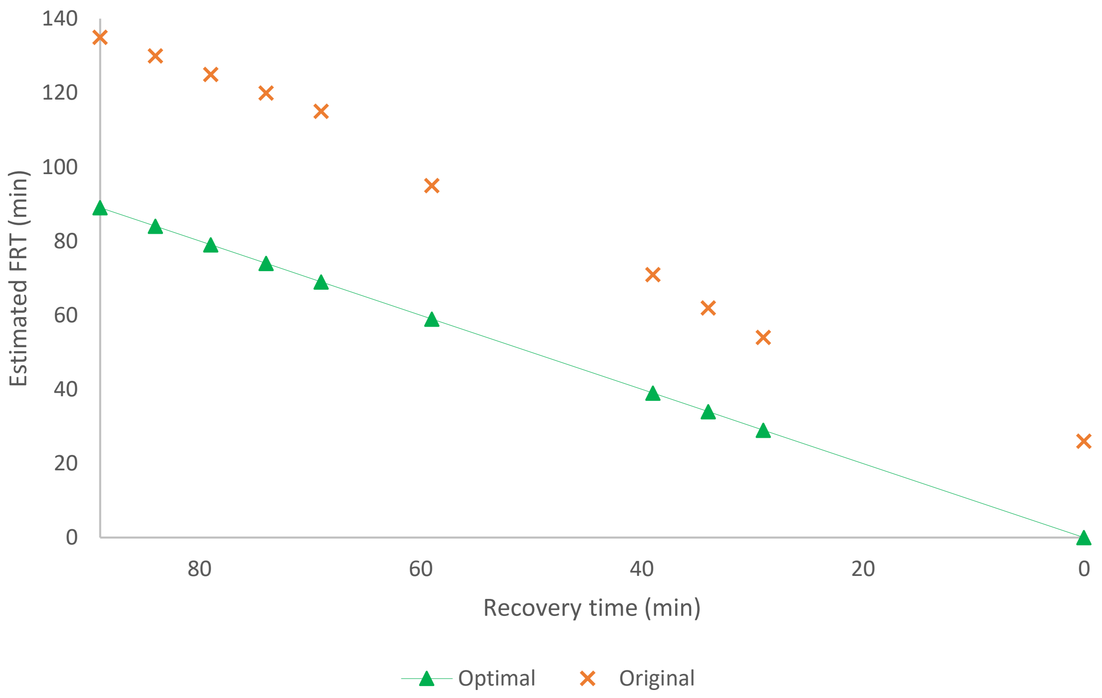


Figure 15: Scenario 3, Amsterdam 7-09-2017

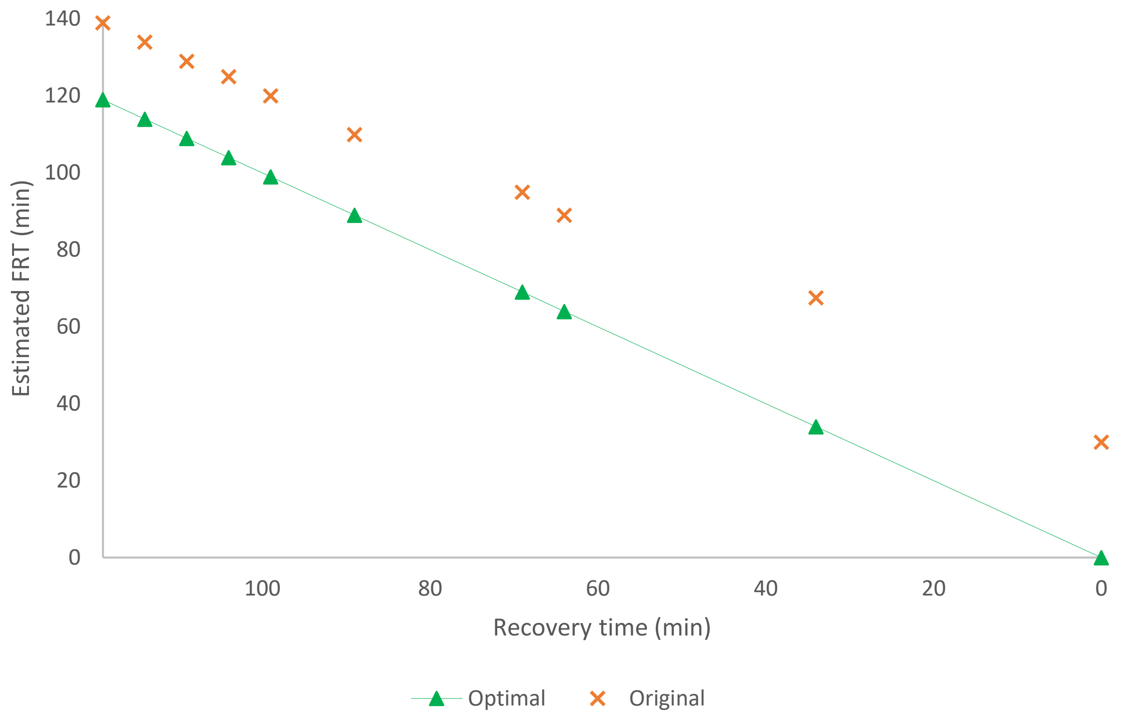


Figure 16: Scenario 4, Utrecht 15-09-2017

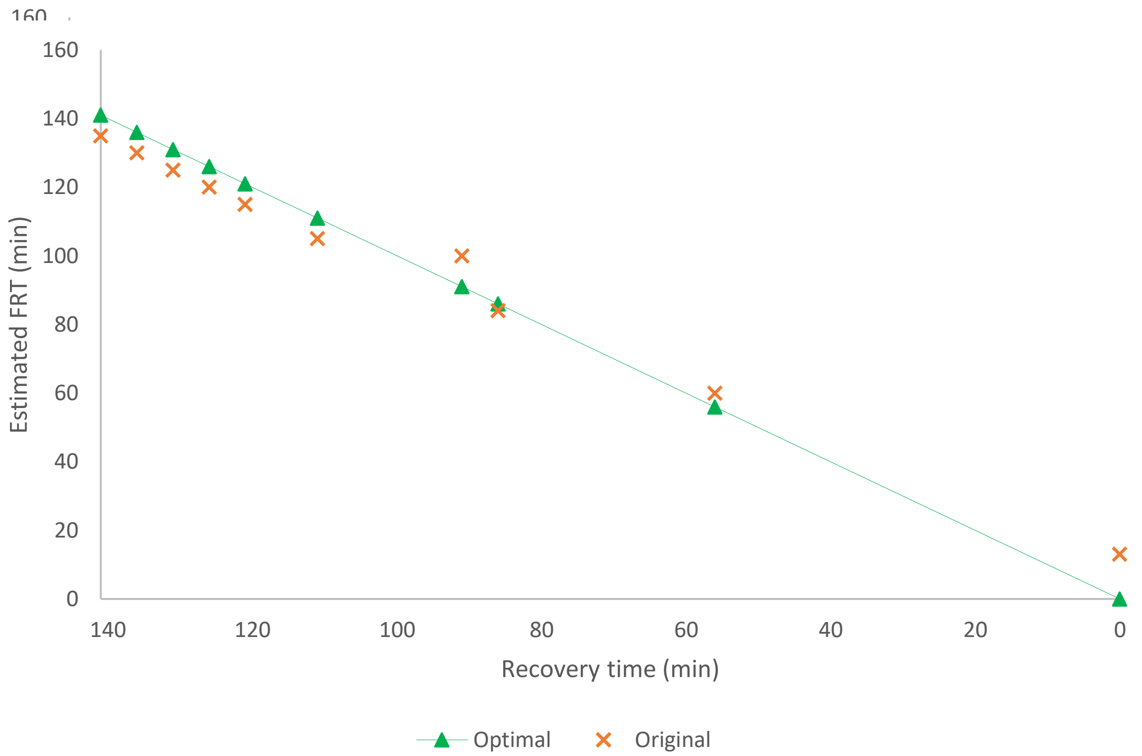


Figure 17: Scenario 5, Dordrecht 31-08-2017.

The previous five graphs gives an insight in the quality of the original prognosis of each scenario. The better the original prognosis fits the optimal prognosis, the better the original prognosis. For instance, in scenario 1, in the beginning the original prognosis is too optimistic and too pessimistic in the end. However, the original prognosis in scenario 5 is pretty much in line with the optimal prognosis.

From this point, every original prognosis could be compared with the optimal one, just by looking at the graphs and calculating the precise error rate. This forms the baseline on which the extended MAS prognosis will be compared. The next paragraph contains results of the extended MAS simulation experiments.

4.3.2 Extended MAS

The aim of the extended MAS is to improve the quality of the estimated function recovery time. Therefore, the behaviour of the extended MAS has to be studied. To gain more insight in the behaviour of this MAS, the results from simulation experiments have to be correctly presented. A feasible way of presenting, is by plotting the twelve prognosis made by agents in one scenario.

An example of such a graph is Figure 18. This figure relates to the twelve extended MAS simulations for scenario 4. Similar figures of the other four scenarios are part of Appendix A. The green line with green triangles represents the optimal prognosis. The orange crosses represents the original prognosis. The blue dots represents the prognosis made by the agents in the extended MAS simulation. Each blue dot corresponds to the outcome of decisions made by agents according to the estimated FRT. The lines between the blue dots do not represent prognosis, since the prognosis are only given at specific moments (i.e. the blue dots). These blue lines make it more clear how an initial prognosis is adjusted by agents during the process. It represents the decision path of agents, which explains that at some points, branches occur. The branches occur because agents try every setting of the Bayesian network, extended RVO and extended tasks of the GL (corresponding to the numbers 1, 3 and 5 in Figure 18 respectively). Every setting of these non-fixed extensions are part of the simulation, since these scenarios do not contain information to choose just one setting of the extensions. So, from the twelve simulations in Figure 18, six times the agents have requested the Bayesian Network and six times this extension was not requested. The same counts for requesting the extended RVO half of the time. Lastly, the agents have three choices for the extended tasks of the GL, namely the GL was trained or the GL was not trained or the extension was not queried. These possible number of combinations of these non-fixed extensions resulted in twelve simulation for scenario 2. The other three extension (see Table 8) differ from the non-fixed extensions. For the switch type algorithm, weather and predicted failure cause algorithm, only one setting is applicable for each extension, namely the one that matches the real condition of a scenario.

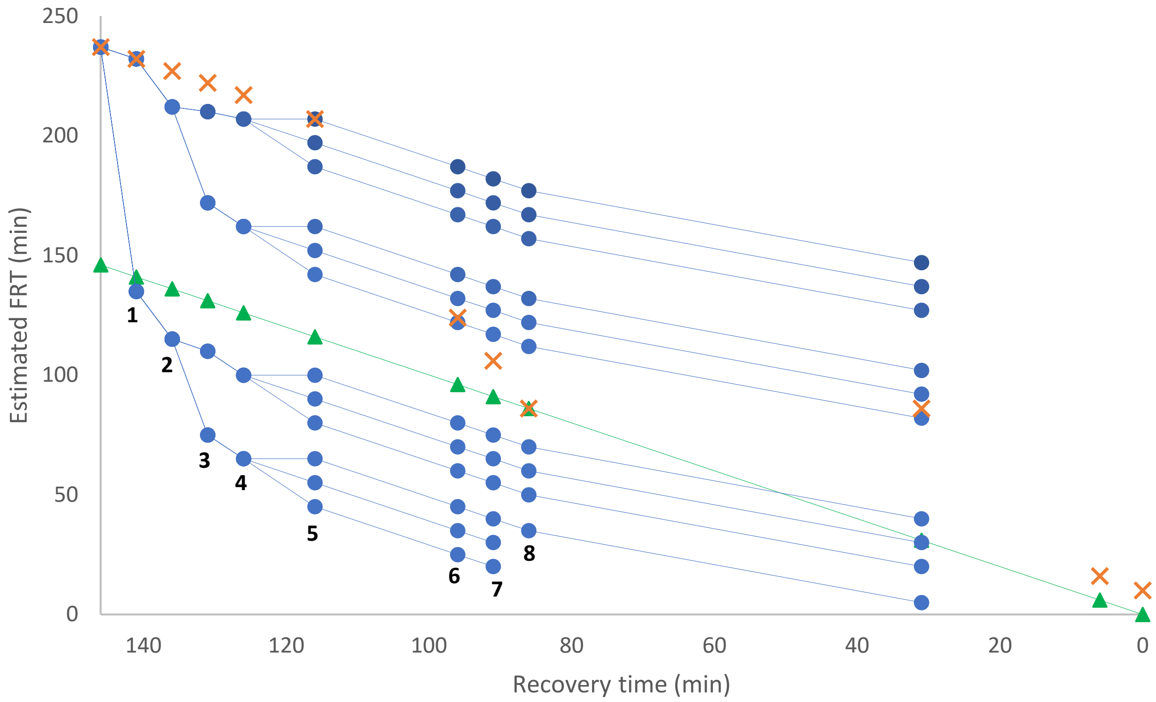


Figure 18: The prognosis for scenario 2. The green line represents the optimal prognosis. The orange crosses represents the original prognosis. The blue dots represent the prognosis made by agents in the extended MAS. The number correspond to the extensions that adjust the prognosis in that moment in time. 1: Bayesian network. 2: Switch type algorithm. 3: Extended RVO. 4: No extension, just prognosis 1 confirmed. 5: Extended tasks GL. 6: Weather. 7: Predicted failure cause algorithm. 8: Prognosis 2. Disclaimer: the blue lines do not represent prognosis. These lines only support the readability of the figure, by visualizing the decision paths.

Although Figure 18 visualizes the prognosis of the extended MAS simulation, it lacks clarity to draw specific conclusions from it. However, there exists a statistical measurement to compare the original and extended MAS prognosis with the optimal one. This statistical measurement is known as the root-mean-square error (RMSE).

The RMSE is a statistical measurement for determining the difference between values predicted by a model and the values actually observed (32). The RMSE is an absolute error measure which states that the scale of the RMSE equals the scale of the used values. The lower the RMSE the better. For this thesis, the RMSE is used to calculate the difference between the optimal prognosis and the original and extended MAS prognosis. So the RMSE provides a measurement to determine in which quantity the prognosis difference from the optimal prognosis. In Table 9 below, for every scenario the RMSE of the original prognosis and the RMSEs of the extended MAS prognosis is presented. These results are generated by the MAS that applied all extensions in predicting the FRT.

Prognose	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
Original	102	74	40	23	6
Extended 1	107	94	35	16	27
Extended 2	128	89	32	24	33
Extended 3	134	83	32	24	41
Extended 4	150	65	28	33	59
Extended 5	149	61	28	35	64
Extended 6	153	57	29	36	61
Extended 7	108	32	39	20	31
Extended 8	116	34	35	21	36
Extended 9	120	36	35	24	42
Extended 10	139	49	29	32	55
Extended 11	146	55	30	36	60
Extended 12	148	60	30	38	65

Table 9: The RMSE scores for the original prognosis and the twelve extended MAS prognosis per scenario. RMSE scores which are lower than the original RMSE, are marked grey.

For scenario, 2, 3 and 4, the agents in the extended MAS making prognosis that are in some cases closer to the optimal one than the original prognosis. Especially, for scenario 2 and 3 the extended MAS making estimations on the FRT that are closer to the optimal prognosis. The RMSE score of scenario 2 in Table 9 correspond to the prognosis presented in Figure 18. In which Extended 1 of Table 9 corresponds to the most upper prognosis in Figure 18. The prognosis right below this prognosis corresponds to Extended 2, and onwards.

In the previous results, each extension is part of the MAS prognosis, except from the dynamic extensions that are switched either on or off. However, in the MAS simulation experiment, every combination of extensions is applied to each scenario. The five prognosis with the lowest RMSE are presented in Table 10. The letters in superscript at the RMSE scores correspond to the extensions that were applied accordingly. See the table description for the meaning of the superscript letters. These results give insight in which combination of extensions contributed to prognosis with a low RMSE.

The RMSE scores of scenario 1 in Table 10 are not lower than the original prognosis for this scenario. However, in general, the RMSE scores are lower compared to the scores in Table 9. So applying some extensions or only one extension in the MAS for predicting the FRT, results in prognosis that fits the optimal prognosis more closely.

Prognose	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
Original	102	74	40	23	6
Rank 1	105^{A}	6^{B}	$4^{C,D}$	1^{B}	1^{A}
Rank 2	$120^{A,B}$	$12^{A,D}$	6^{C}	$7^{C,D}$	$11^{A,B}$
Rank 3	$123^{A,D}$	$17^{A,D,E,F}$	$16^{B,C,D}$	$16^{C,D,E}$	$15^{A,B,D}$
Rank 4	$127^{A,B,D}$	$19^{A,B}$	$17^{B,C,D,E,F}$	$18^{C,E}$	$15^{A,B,E}$
Rank 5	130^{B}	$23^{A,C,E,F}$	$19^{B,C}$	$18^{B,E}$	$17^{A,B,D,E}$

Table 10: The five lowest RMSE of the extended MAS prognosis that only used a specific selection of extensions. The letters in superscript correspond to the extensions that were applied in the MAS simulation accordingly. A: Bayesian network. B: Switch type. C: Extended RVO. D: Extended tasks GL. E: Weather. F: Predicted failure cause. RMSE scores which are lower than the original RMSE, are marked grey.

For each scenario, the prognosis with the lowest RMSE applied only one or two extensions for estimating the FRT. Taken these five RMSE scores per scenario into account, it seems that some extensions contributed to prognosis that fits the optimal one closely. I will evaluate these results in the next chapter.

4.3.3 Results conclusion

In this chapter, I have motivated the steps in developing the baseline MAS and the extended MAS. The baseline MAS simulates the original process of estimating the FRT perfectly. This forms a feasible base to develop the extended MAS on. The extensions of the extended MAS has the aim to positively adjust the estimation of the FRT. Although the influence of the extensions are a rough estimation on previous research, expert knowledge and common sense, the extended MAS generated more prognosis that are closer to the optimal prognosis in four of the five scenarios, compared to the original prognosis in these scenarios.

4.4 Evaluation

In the following, I will give an evaluation of the results from the simulation experiments. Therefore, I will dive into the extensions that influence agent's decision making and elaborate on which combination of extensions resulted in well performing prognosis.

4.4.1 Extended MAS

The extended MAS as described in section 4.3.2, is a system of agents that uses additional information for decision making on the estimated FRT. The aim of these extra sources of information is to let the agents benefit from it, in the process of making prognosis. For scenario 2 and 3, I will dive into some prognosis that fit the optimal prognosis better than the original one. I will start with given reasons for the fact that the extended MAS generates better prognosis in scenario 2, 3 and partly in 4, but performs worse in scenario 1 and 5.

In Figure 18 most extended MAS prognosis perform better than the original one. However, the RMSEs for scenario 5 makes clear that every extended MAS prognosis fits the optimal prognosis worse, compared to the original one.

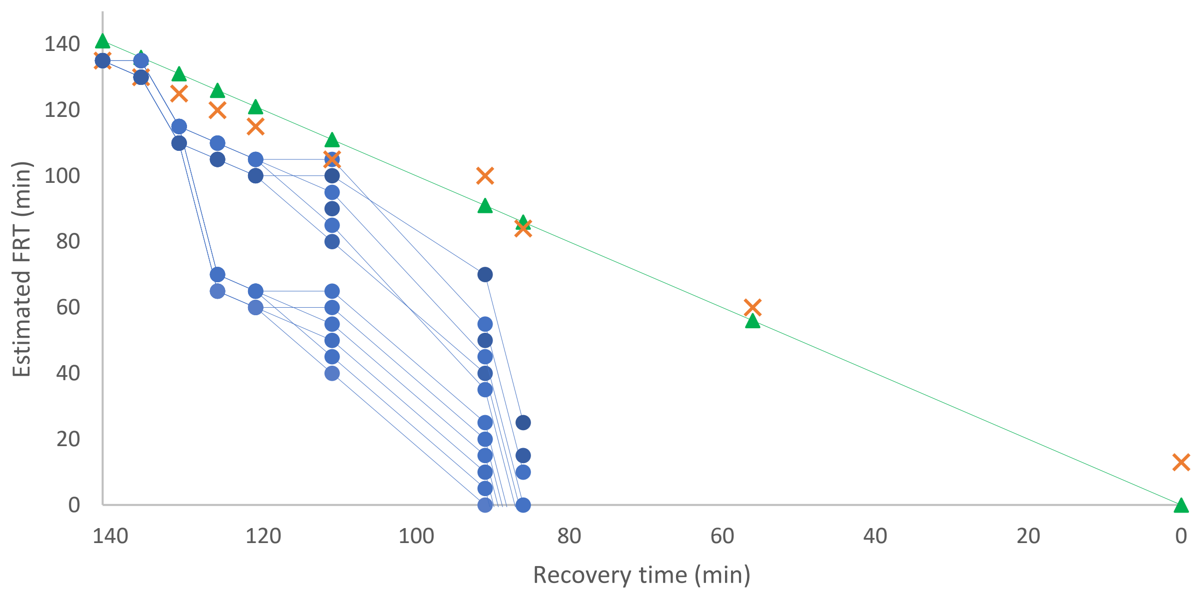


Figure 19: The prognosis for scenario 5. The green line represents the optimal prognosis. The orange crosses represents the original prognosis. The blue dots represent the prognosis made by agents in the extended MAS. Disclaimer: the blue lines do not represent prognosis. These lines only support the readability of the figure, by visualizing the decision paths.

Figure 19 shows the optimal, original and extended MAS prognosis for this scenario. The original prognosis performance very well, as also could be seen in Figure 17. Focussing on the extended MAS prognosis, the pattern has similarities with the patterns in Figure 18 In both cases, agents decide to shorten the prognosis during the process. For instance, in Figure 18, the two far most right blue dots correspond to the weather and the predicted failure cause extension. Based on the scenario as stated in Table 8, the agents decide to shorten the prognosis with 10 minutes (weather type: sunny) and 40 minutes (predicted failure cause: engine) respectively. These decisions make the extended MAS prognosis in this scenario too optimistic. This optimistic behaviour of the extended MAS explains the fact that the MAS is performing well in scenario 2 and 3. Since in these scenarios, the original prognosis is too pessimistic, see Figure 14 and 15. With this in mind, in the following, I will evaluate an extended MAS prognosis that outperform the original one.

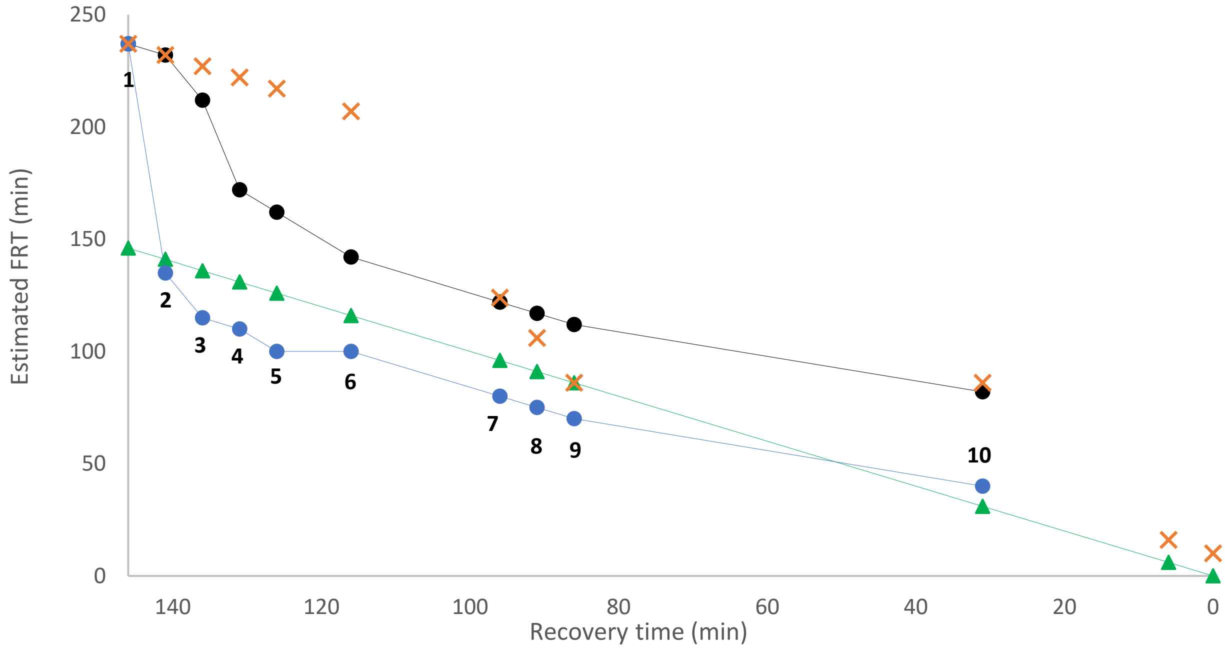


Figure 20: The low blue dots represent the best performing (RMSE 32) extended MAS prognosis 7 in scenario 2. The black dots represent extended prognosis 6 (RMSE 57). The green line is the optimal line. The orange crosses are the original prognosis. The numbers correspond to the extensions that influenced the adjustment of the prognosis. 1: Initial prognosis. 2: Bayesian network. 3: Switch type. 4: Extended RVO. 5: No extension, just prognosis 1 confirmed. 6: extended tasks GL. 7: Weather. 8: Predicted failure cause. 9: Prognosis 2. 10: Prognosis 3.

In Table 9, all the RMSEs of the original and extended MAS prognosis are presented. In scenario 2, extended 7 performs the best (RMSE of 32). In Figure 20 this extended MAS prognosis is presented. The labels in the figure correspond to the extensions that played a role in adjusting the prognosis. In the beginning of the process, the agents in the extended MAS have to deal with the initial prognosis. In this simulation, the MKS agent requests the Bayesian network, marked as number 2 in the figure, on which he changes the prognosis to 140 minutes. Due to the switch type, number 3, the prognosis is shortened with 20 minutes. For extension 6 in this case, the GL agent does not perform extended tasks, and therefore the prognosis is incremented with 10 minutes, which results in a better fit with the optimal line. So not performing extra tasks of the GL agent, results in a better prognosis, in this scenario. If the GL agent did perform extra tasks, then the prognosis would become worse in this scenario. This contradicts the aim of this extension. However, in extended prognosis MAS 6, represented by the black dots in Figure 20, the GL agents performs extended tasks, which results in a better fit with the optimal prognosis. It turns out that extending the tasks of the GL agents only results in a better prognosis if the previous prognosis was to long (i.e. above the optimal line).

This principle, that the quality of an extension depends on the initial prognosis and adjustments of previous extensions, gives difficulties in making general statements about the performance of each extension on its own.

4.4.2 Best performing prognosis

However, in the MAS simulation experiment, every combination of extensions has been applied in predicting FRT, to gain more insight in the performance of the extensions. In Table 10 the five lowest RMSE scores are presented. Focussing on the lowest RMSE in each scenario, it turns out that only applying one or two extensions, results in the best extended MAS prognosis. For scenario 2 and 4, only applying the switch type extension resulted in prognosis with a RMSE of 6 and 1 respectively. For scenario 5, only applying the Bayesian network extension resulted in a prognosis with a RMSE of 1, which is better than the original prognosis. In this scenario, the Bayesian network extension is part of all five best performing extended MAS prognosis. For scenario 4, applying both the extended RVO and the extended tasks of the GL in the MAS, resulted in the prognosis with the lowest RMSE. Combining these two extensions, resulted in scenario 4 the second best performing prognosis. Another relation of extensions between scenario could be found in scenario 1, 2, and 5. In these scenarios, the Bayesian network is part of almost every top five prognosis. In general, between the scenarios, there seems to be no further consistent relation among the extensions that contribute to well performing prognosis.

So far, the extensions that contribute to a well performing prognosis have been discussed. On the other hand, one extension does not played a sufficient role in the RMSE of well performing prognosis, as listed in Table 10. Namely, the predicted failure cause extension. This extension is only applied in three of the top five performing prognosis. Apparently, the predicted failure cause does not enrich the quality of the prognosis, as simulated in the MAS of this study .

4.4.3 Evaluation conclusion

In this previous sub-section I presented an evaluation of the results from the extended MAS simulation. Some extended MAS prognosis perform better than the original prognosis. In general, the extended MAS made better prognosis in scenarios in which the original prognosis was too pessimistic. Applying only one extension or a specific combination of extensions in the MAS, resulted in better prognosis in four of the five scenarios.

4.5 Discussion

In the following section I will elaborate on the meaning of the results, limitations and applicability of this research. The section begins with the validation of this study.

4.5.1 Validation

The validity of the research design has multiple aspects. To start with the internal validity, namely the design of the simulations. To determine the effect of the extensions on the quality of the prognosis, this research is designed to conduct simulation experiments. In this experiments, the results of the extended MAS is compared to the baseline MAS. The baseline simulates original decision making in the scenarios deterministically. Extending this baseline model and comparing the performance with the original scenario is a suitable design to measure the quality of the prognosis. This is a valid design to measure effect of the extensions.

In terms of external validity, this research provides insight in the usage of validated additional sources, to better predicted FRT. The results make clear that in some scenarios, the extended MAS does make better prognosis, based on validated extra information. To check the overall external validity, realisation data is acquired to check the validity of the extensions that are currently applicable to the scenario (see Table 8). According to the extensions that has an impact on the FRT, the parameters of these extension are based on expert judgement. Since there is no realisation data at hand, conducting experiments to test the impact of these extensions is a recommended next step.

4.5.2 Results elaboration

The prognosis generated by the extended MAS simulation is discussed in more detail here. I will argue how valuable the results are. By taken the performance of each extension and the extended MAS prognosis into account.

First of all, take into account that the context is shaped by the optimal prognosis. The optimal prognosis is an unrealistically well performing prognosis, that only serves a feasible base on which the quality of the other prognosis are calculated. So, performing less than the optimal prognosis is reasonable. The aspect to focus on is the comparison between the extended MAS prognosis and the original prognosis.

The results in Table 9 make clear that the extended MAS generates better prognosis in two of the five scenarios. For these prognosis, all the extensions are combined. For scenario 2 and 3, fifty-fifth applying the dynamic extensions (i.e. the Bayesian network, the extended RVO and the extended GL tasks), in combination with the fixed extensions, resulted in better prognosis. In these two scenarios, an extension could adjust the prognosis negatively, although overall, still performing better than the original prognosis. These findings are valuable in the sense that for some scenarios, if the agents use a combination of all extensions, the estimated FRT is improved. However, in the other three scenarios, the quality of the estimated FRT is worse compared to the original one.

Besides these results, the extended MAS simulation generated prognosis in which only one extension or a combination of extensions was applied. It turned out that from these prognosis, in four of the five scenarios, the extended MAS made better prognosis compared to the original one, see Table 10. Only for scenario 1, the extended MAS has not made a better prognosis. For scenario 5, only applying the Bayesian network resulted in a better prognosis compared to the original one. Overall, the best performing prognosis in each scenario only used one or two extensions (see Table 10). For these scenarios, it turned out that less is more.

In an ideal situation, the extended MAS generated improved prognosis is every kind of scenario. Only for scenario 1 the extended MAS has not generated a better prognosis compared to the original one. For the other four scenarios, the agents used a specific combination of extensions that improved decision making on the predicted FRT.

4.5.3 Limitations

I will give reasons for some limitations of this research and explain the impact that each of them has.

Firstly, despite using existing research and expert knowledge, the parameters for the extensions are not as dynamic as they could be. Therefore, it is currently unknown how the extended MAS will perform in practice. This study requires further research when generalizing for additional scenarios, but appeared sufficient for the initial scenarios.

Secondly, the extended MAS performed better than the original prognosis for several of the scenarios. However, to classify these performance as significantly better, additional measurements are required. This is an option for possible future research.

Lastly, the performance of the extended MAS is measured in five scenarios. This small size of scenarios makes it hard to generalize the performance of the extended MAS. Due to time issues this study contains a small set of scenarios. However, this thesis aims to provide a proof of concept, a goal that is achieved even using a few scenarios.

4.5.4 Applicability

It is yet to be investigated to what extent the extended MAS is applicable in practice. Since the MAS has only been tested in five scenarios, the question remains in how well the MAS should perform in other scenarios. However, in terms of applicability, the validity of each extension is relevant. For the extensions that influence the FRT itself, it would be recommended to test the influence they have in practice. This is a method to judge the validity of these extensions. For the other extensions, that have no influence on the operation, validation could be performed by using realisation data. These methods for validating the extensions and therefore the extended MAS itself, is a recommended approach to state more about the applicability of this research.

4.6 Conclusion

The present study shows the process of modelling the disruption management system as a MAS, and enrich the MAS with algorithmic modules and different communication lines, to better predict the FRT. The influences of the extensions are roughly estimated on existing research and expert judgement. In general, the developed MAS is able to improve its performance by using information in a smart way, which overall is a proof of concept. A proof of concept that makes clear that the impact of extensions, that improve FRT and the prediction of it, can be simulated in a MAS and tested accordingly. Keeping the validity issues, of the extensions that influence the FRT process itself, in mind.

5 Conclusion

In this chapter, I will give answers to the research question and related sub-questions as stated at the beginning of this thesis. Besides giving answers, I will come up with topics for future research.

5.1 Findings

The developed MAS, results and conclusion form the previous chapters can be taken together to answer the main research question and related sub-questions. The main research question was:

- Can the disruption management system of ProRail be modelled as a multi-agent system to make the usage of information smarter through better information sharing among the participating agents to improve the quality of the estimated failure recovery time?

From this main question, three sub-questions were divided:

- How can the disruption management system be modelled as a multi-agent system?
- How can the multi-agent system use information in a smarter way, i.e. how to make the information sharing protocol smarter?
- How to evaluate the modified multi-agent system to check if it improves the quality of the estimated function recovery time?

The answer to the first question was acquired through defining the system's specifications, so it's main goal and sub-goals. These goals form the base for agent descriptors. These descriptors contained all the essential information for each of the four agents. Next step was specifying the relationship among agents. This approach of modelling the disruption management system as a MAS has been inspired by the Prometheus methodology. This approach resulted in a step by step modelling process. By modelling the disruption management system in such a structurally manner, the most important characteristics and components of the system arose. In particular, aspects according to decision making on determining prognosis. Having clarified the agent's communication, tasks and decision making related to estimating the FRT, resulted in a functional disruption management MAS.

In order to answer the second sub-question, possible weak spots of the current disruption management system and additional modules that could enrich the system have been studied. According to studying current aspects of the system that could be improved, evaluation reports at ProRail were consulted. These reports and opinions of experts, made it clear that the role of the GL could be improved by changing its tasks. In the short sense, these changes in the tasks of the GL related to actively confirming the initial prognosis and communicating more on the validity of information.

Moreover, the information sharing protocol could become smarter by using additional modules for decision making. Some specific characteristics of the disruption management system have been topic of previous research. These previous research projects gained insight in factors that influence the function recovery time and developed algorithms that benefits the understanding of some characterises of the disruption management system. To shortly sum up these additional modules: a Bayesian network for generating an initial prognosis. The type of switch is a relevant aspect to take into account. Knowing the precise location of the disruption benefits the function recovery time. Taking weather conditions into account in the prognosis that is being shared. Lastly, having a prediction about the failure cause, shortens the search time on the track and therefore benefits the recovering process. These extension have been integrated in the right place of the baseline MAS. Overall, these list of additional modules add value in making the information sharing protocol smarter.

The last sub-question was aimed to check if the additional modules and the adjustments in communications, improves the quality of the estimated FRT. To get an answer, a MAS has been developed that simulates the disruption management system and outputs the estimated FRT. This MAS simulates the estimated FRT of the five original scenarios perfectly. This forms the baseline MAS. Onwards, the posted extensions for the information sharing protocol are added to the baseline MAS, which results in an extended MAS. This extended MAS also outputs the estimated FRT. The next step consisted of conducting simulation experiments to gain insight in the performance of the extended MAS. To measure the performance of the extended MAS prognosis, an optimal prognosis was made for each scenario. This optimal prognosis served as a base to determine the difference of the original and extended MAS prognosis, by calculating the RMSE. The lower the difference, the better the quality of the prognosis. Lastly, the performance of each extensions could be determined in a similar fashion.

Each answer of the sub-questions contributes in answering the main research question. The goal of the research question was to come up with a MAS solution for better predicting the FRT in the disruption management system. Approaching this problem from a MAS perspective is something that has never been done before. In the light of the main goal, I managed to develop a MAS that simulates the disruption management system. In this MAS, all the communication lines relevant for predicting the FRT are incorporated. The communication takes place between agents, that represent the parties that are part of the disruption management system. The agents have capabilities, which is a collection of actions they can perform, like sending requests and make decisions. This network of agents that perform actions, communicate and make decisions simulates the mechanism of the disruption management system. In this system, making reliable prognosis early in the process is a difficult thing to do. Therefore, I added modules and changed communication lines in the original MAS, with the aim to enrich the process and improve the quality of the prognosis. These modules consists of existing algorithms and conclusions form data analysis that could support decision making. Extending the original MAS with these modules resulted in a modified MAS. With this modified MAS, simulation experiment were conducted. Therefore, the process of making prognosis was simulated in the modified MAS, which used all the possible combinations of extensions. It turned out that, if all extensions were applied, in two of the five scenarios, the modified MAS generated better prognosis than the original prognosis. Focussing on applying only one or a specific combination of extensions to the MAS, the agents predicted the FRT better in four of the five scenarios. So the modules improves the estimated FRT in some situations.

If a closer look is taken at the performance of the modified MAS, then it turns out that the quality of the prognosis depends on the scenario it is part of. This dependence makes it difficult to generalize the performance of the modified MAS to other scenarios. Moreover, for some modules, the influence they have on decision making is based on a rough estimation. However, the modified MAS is capable of using additional information and acting smartly on it, which results in better prognosis.

The modified MAS could also be viewed from a different perspective. Namely, a MAS that is capable of judging the quality of modules, which are aimed to improve the estimated FRT. So, some sort of judging system that checks the effect of on extension on the estimated FRT in the disruption management system. Since ProRail is working on improving the prognosis and optimizing the disruption management in general, the MAS of this thesis could also be useful as an evaluation tool.

5.2 Future subjects

In this study, I demonstrated the use of multi-agent systems for better estimating the FRT in the disruption management system. A model can always be improved. In this case, for the modified MAS, the model could be improved by making it capable to process several incoming disruptions in parallel. The current modified MAS can only process one disruption at the time. However, at ProRail several disruption could occur on the same time. For instance, at the 25th of January, a heavy storm caused 344 incidents at ProRail's territory. Although this is a unique situation, improving the modified MAS so it is able to handle disruptions parallelly, is a valuable topic for future research at ProRail.

The MAS of this research could also in another way be improved. By making the MAS capable of automatically use information from scenario, to test its performance on. Since this study only used five scenarios, automating processing scenarios would make it possible to test the performance of the MAS on a larger scale.

Another subject for future research is modelling the disruption process as a normative MAS. Normative MAS has been a well-study topic considering human and artificial agent cooperation and co-ordination, group decision making, secure multiagent systems, and so on. Approaching the behaviour of the disruption management system with typical normative concepts as norm violation, permission or prohibition and normative deadlines, to name a few, would shine a new academic light on a familiar problem.

Lastly, some agents that are part of the disruption management system could be assisted by an intelligent agent. For instance, the Dispatcher agent has difficulties in making an optimal RVO in cooperation with the MKS. Moreover, the Dispatcher is challenged in getting a clear picture of the details of a disruption. Using voice-to-text techniques, to grasp information that is shared among the parties, could be caught by an intelligent assistant agent. Who is supporting the Dispatcher in asking the right questions, prioritizing tasks and making correct decisions. This intelligent agent could be implemented besides the programmes that the Dispatcher is currently using.

5.3 Conclusion

This study has investigated the possibilities of using an improved MAS to better predicted the FRT during the disruption management process. Although the performance of the extended MAS depends on the scenario it is part of, the results made clear that in four of the five scenarios, the agents better predict the FRT, when using an extension or a combination of extensions. Overall, the developed MAS and it performance shows that the concept works in some situations. In this light, this study is a proof of concept which opens doors for future research.

Bibliography

1) Stan Franklin and Art Graesser. Is it an agent, or just a program?: A taxonomy for autonomous agents. In *Intelligent agents III agent theories, architectures, and languages*, pages 21–35. Springer, 1997.
2) Russell,S.J. Rationality and intelligence. Artificial Intelligence, Vol. 94, 1997. p.57-77.
3) M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. The Knowledge Engineering Review, 10(2):115–152, 1995.
4) Stuart Russell and Peter Norvig. A modern approach. 1995.
5) Weiss, Gerhard, ed. *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT press, 1999.
6) Nicholas R Jennings, Katia Sycara, and Michael Wooldridge. A roadmap of agent research and development. Autonomous agents andmulti-agent systems, 1(1):7–38, 1998.
7) Rafael H Bordini, Lars Braubach, Mehdi Dastani, Amal El Fallah- Seghrouchni, Jorge J Gomez-Sanz, Joao Leite, Gregory MP O'Hare, Alexander Pokahr, and Alessandro Ricci. A survey of programming languages and platforms for multi-agent systems. *Informatica (Slovenia)*, 30(1):33–44, 2006.
8) Peter Stone and Manuela Veloso. Multiagent systems: A survey from a machine learning perspective. Autonomous Robots, 8(3):345–383, 2000.
9) Rafael H Bordini, Mehdi Dastani, and Michael Winikoff. Current issues in multi-agent systems development. In Engineering Societies in the Agents World VII, pages 38–61. Springer, 2007.
10) Siebers, Peer-Olaf, and Uwe Aickelin. "Introduction to multi-agent simulation." (2008).
11) McArthur, Stephen DJ, et al. "Multi-agent systems for power engineering applications—Part I: Concepts, approaches, and technical challenges." IEEE Transactions on Power systems 22.4 (2007): 1743-1752.
12) Jennings, Nicholas R., Katia Sycara, and Michael Wooldridge. "A roadmap of agent research and development." *Autonomous agents and multi-agent systems* 1.1 (1998): 7-38.
13) Lundberg, Jenny, and Anne Håkansson. "An Approach towards using Agent in Multi-Agent Systems to streamline emergency services." (2008).
14) Basak, Swati, Neelam Modanwal, and Bireshwar Dass Mazumdar. "Multi-agent based disaster management system: a review." *International Journal of Computer Science & Technology* 2.2 (2011): 343-348.

15) Lee, Seungho, Young-Jun Son, and Judy Jin. "An integrated human decision making model for evacuation scenarios under a BDI framework." *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 20.4 (2010): 23.
16) Adams, N. M., et al. "The aladdin project: intelligent agents for disaster management." *IARP/EURON Workshop on Robotics for Risky Interventions and Environmental Surveillance (RISE*. 2008.
17) Carrera, Álvaro, et al. "A real-life application of multi-agent systems for fault diagnosis in the provision of an Internet business service." *Journal of Network and Computer Applications* 37 (2014): 146-154.
18) Ciampolini, Ann, P. Mello, and S. Storari. "Distributed medical diagnosis with abductive logic agents." *Proceedings of BIXMAS 2002 workshop*. 2002.
19) Sharma, Somesh Kumar, Swatantra Vishwakarma, and Nishant Jha. "Prognosis agent technology: influence on manufacturing organizations." *The International Journal of Advanced Manufacturing Technology* (2017): 1-12.
20) Hassan, Samer, et al. "Asking the oracle: Introducing forecasting principles into agent-based modelling." *Journal of Artificial Societies and Social Simulation*16.3 (2013): 13.
21) Rogoza, W., and M. Zabłocki. "A Weather Forecasting System Using Intelligent BDI Multiagent-Based Group Method of Data Handling." *International Multi-Conference on Advanced Computer Systems*. Springer International Publishing, 2016.
22) Wit, N. *Development of a Reliable Prediction Method for Urgent Infra-Failure Recovery Times at ProRail BV*. MS thesis. University of Twente, 2016.
23) Pěchouček, Michal, and Vladimír Mařík. "Industrial deployment of multi-agent technologies: review and selected case studies." *Autonomous Agents and Multi-Agent Systems* 17.3 (2008): 397-431.
24) Dastani, Mehdi, and Bas Testerink. "Design patterns for multi-agent programming." *International Journal of Agent-Oriented Software Engineering* 5.2-3 (2016): 167-202.
25) Anand S Rao,Michael P Georgeff, et al. Bdi agents: From theory to practice. In ICMAS, volume 95, pages 312–319, 1995.
26) David Kinny, Michael Georgeff, and Anand Rao. A methodology and modelling technique for systems of bdi agents. In Agents breaking away, pages 56–71. Springer, 1996.
27) Lokuge, Prasanna, and Damminda Alahakoon. "Improving the adaptability in automated vessel scheduling in container ports using intelligent software agents." *European Journal of Operational Research* 177.3 (2007): 1985-2015.

28) A.D. Lattner et al., “Sequential Pattern Mining for Situation and Behavior Prediction in Simulated Robotic Soccer,” Proc. 9th Robot Soccer World Cup Conf. (RoboCup 05), Springer, 2005, pp. 118–129.

29) Norling, Emma, Liz Sonenberg, and Ralph Rönnquist. "Enhancing multi-agent based simulation with human-like decision making strategies." *International Workshop on Multi-Agent Systems and Agent-Based Simulation*. Springer, Berlin, Heidelberg, 2000.

30) Padgham, Lin, and Michael Winikoff. "Prometheus: A methodology for developing intelligent agents." *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*. ACM, 2002.

31) Zilko, AURELIUS ARMANDO. "Mixed Discrete-Continuous Railway Disruption-Length Models with Copulas." (2017).

32) Doniec, Arnaud, et al. "Non-normative behaviour in multi-agent system: Some experiments in traffic simulation." *Intelligent Agent Technology, 2006. IAT'06. IEEE/WIC/ACM International Conference on*. IEEE, 2006.

33) Tielman, W. L. *An Agent-based Approach to Simulating Train Driver Behaviour*. MS thesis. 2015.

34) Van Beek, R. *A Statistical Model of Traffic Control.* MS thesis. 2013.

35) Siegfried, Robert. *Modeling and simulation of complex systems: A framework for efficient agent-based modeling and simulation*. Springer, 2014.

36) Leistra, B. *Identification and cause isolation of Dutch rail dispatching system failure notifications.* MS thesis. 2017.

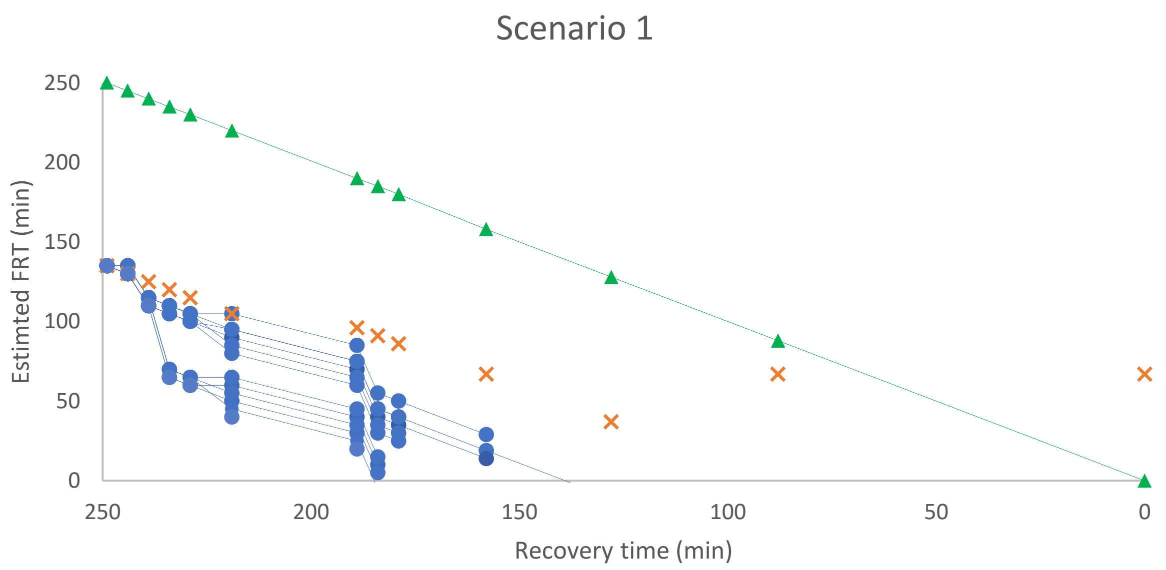
Glossary

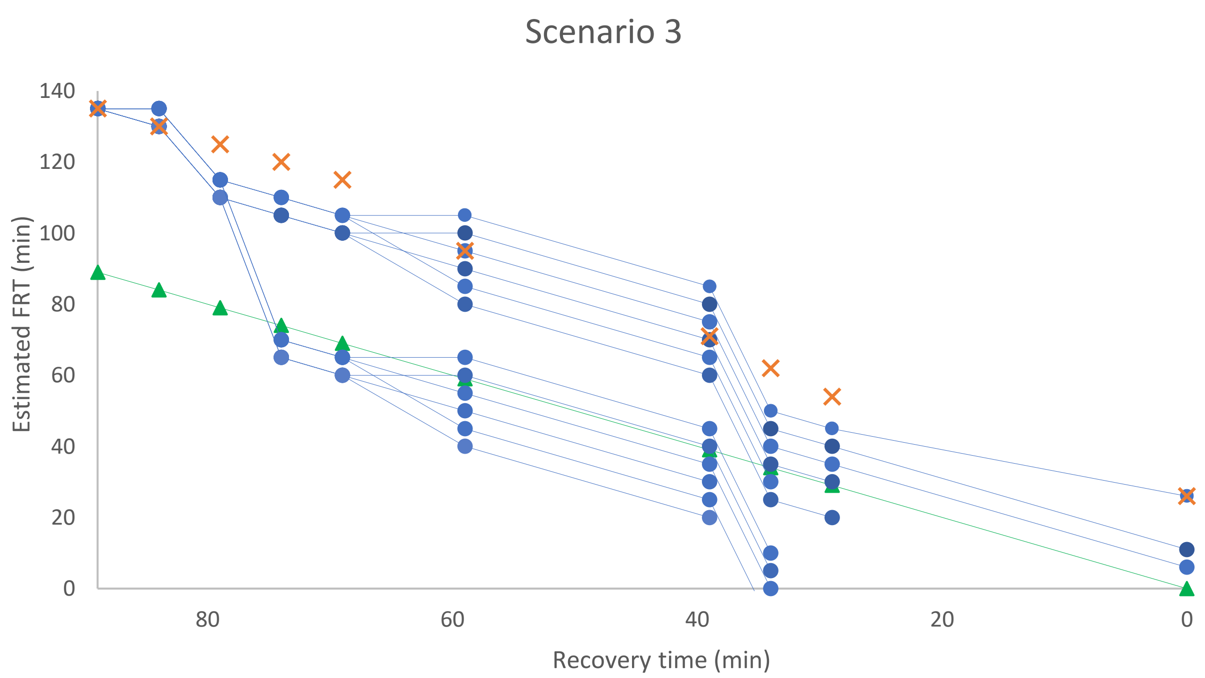
Delay	*Vertraging*	The time difference between the actual activity and the planned activity
Disruption	*Storing*	A collection of irregularities, with the consequence that the functionality of an infra object is not available
Function Recovery Time (FRT)	*Functie herstel tijd*	Time it takes to recover the function of a failed infra object
Prognosis	*Prognose*	Prediction of the time it will take to recover the function of a disrupted infra object
Railway control room	*Meldkamer spoor (MKS)*	The failure-handeling organisation of ProRail's Asset Management within the Operational Control Centre Rail
Report of irregularity (RVO)	*Rapport van onregelmatigheid*	A report that contains all the relevant information, such as location, type of infra object and priority level, for the disruption management process
Traffic control	*Verkeersleiding*	The department of ProRail that is tasked with directing train traffic through controlling switches and signals where necessary

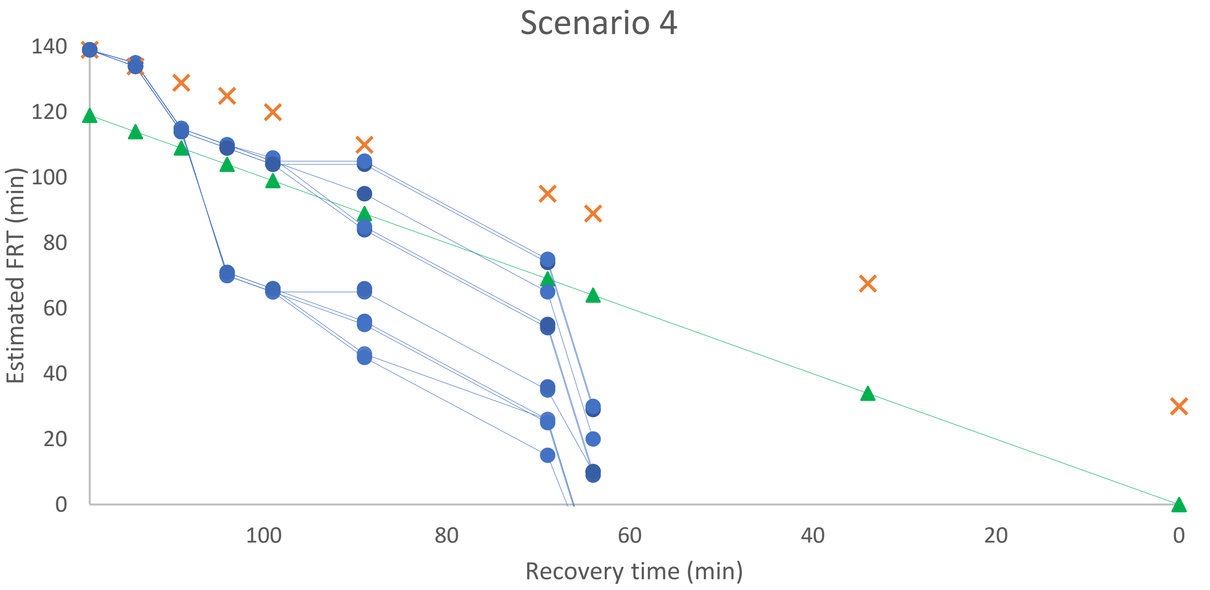
Appendix

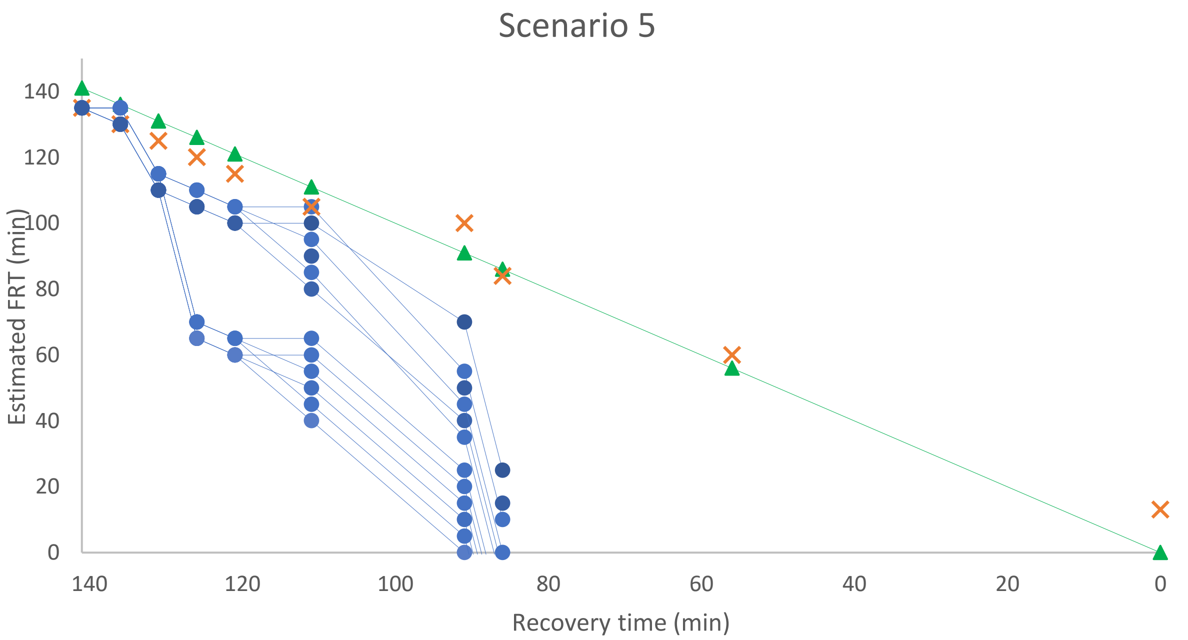
A Prognosis figures

For the figures, the green line represents the optimal prognosis. The orange crosses represents the original prognosis. The blue dots represent the prognosis made by agents in the extended MAS. Disclaimer: the blue lines do not represent prognosis. These lines only support the readability of the figure, by visualizing the decision paths.






Scenario 4
140
120
100
80
60
40
20
0
Estimated FRT (min)
100
80
60
40
20
0
Recovery time (min)


Scenario 5
140
120
100
80
60
40
20
0
Estimated FRT (min)
140
120
100
80
60
40
20
0
Recovery time (min)

B Agent's capabilities

The capability of each agent. The capability contains all the actions an agents can perform in the extended MAS.

Dispatcher agent
• Start making RVO • SendPreciseFailureLocation • ApprovingBUTA

GL agent
• HandleInitialPrognosis • SaveAgentAdresses • SendETA • HandlePrognosisRequest • DetermineCurrentPrognosis • HandleUpdateAvailableEquipment • HandlePrognosisResults • HandleAskForPrognosisUpdate • HandleDiscussedUpdatePrognosisResult • HandleFinalPrognosisRequest • LogFinishedRepairingProcess

MKS agent
• HandleRequestForMakingRVO • SendRVO • HandleAcceptedRVO • ShareInitialPrognosis • HandleConfrimedInitialPrognosis • HandleGL_ETA • HandleContractorETA • AskForCurrentPrognosis • HandlePrognoseReply • HandleFailureCause • HandleReplyForUpdatedPrognosis • HandleFinalPrognosisReply

Contractor agent
• HandleRVO • HandleInitialPrognosis • HandleRequestForSendingETA • DiagnosingFailureCause • BringEquipment • HandleDiscussedPrognosis • HandleApprovedBUTA • HandleDiscussedUpdatedPrognosis