MASTER THESIS

MULTI-AGENT SYSTEM DESIGN FOR AUTOMATED DOCKING OF SEMI-TRAILERS BY MEANS OF AUTONOMOUS VEHICLES

Berry Gerrits

INDUSTRIAL ENGINEERING AND BUSINESS INFORMATION SYSTEMS

EXAMINATION COMMITTEE
DR. P.C. SCHUUR
DR. IR. M.R.K. MES

EXTERNAL ADVISOR
ING. L.R. BUNING MSC MBA

Research supported by the Dutch organization for scientific research (NWO) of the Ministry of Education, Culture and Science through the RAAK-PRO project: "Intelligent Truck Applications in Logistics" (INTRALOG).

12-04-2016

UNIVERSITY OF TWENTE.

MANAGEMENT SUMMARY

In this thesis we develop a generic automated planning and control system based on agent technology for the pick-up and docking of semi-trailers by means of Automated Guided Vehicles (AGVs) in a collision- and conflict free environment**.** To design a comprehensive Multi-Agent System (MAS) we (i) decompose the system into a functional specification, (ii) design agents based on the functional specification, (iii) develop the required interactions between the agents and the environment and (iv) develop the decision capabilities of the agents. We then put our design to work using a case study provided by logistics service provider Rotra through the RAAK-PRO project Intelligent Truck Applications in Logistics (INTRALOG). The case study provides a pilot location where a new cross-dock will be built in the near future. We focus on this pilot location where the handling of semi-trailers should be done autonomously where we use yard tractors (YTs) as AGVs. We validate our design by building a discrete-event simulation model tailored to the pilot location.

This thesis contributes to both theory and practice by identifying the potential of automatic driving within the logistics domain by working together with both business and scientific partners within the consortium of INTRALOG. The relevance of this research and the added benefit of autonomous vehicles is driven by three factors: (i) 'Active Safety' (reducing accidents caused by human errors), (ii) 'Efficiency' (the increase of transport system efficiency) and (iii) 'Comfort' (freedom of the user when automated systems are active) ("HTSM Automotive Roadmap 2016-2020," 2015). Especially the latter is an interesting business opportunity for logistic service providers, such as Rotra to decrease waiting time and maneuvering time at distribution centers and thus to overcome problems with the driver's tachograph as they have to comply with European Law regarding driving times and rest periods.

We approach the design of the MAS by first defining an agent as "a computer system that is *situated* in some *environment* and that is capable of *autonomous action* in this environment in order to meet its design objectives." (Wooldridge, 2009). We employ the Prometheus methodology to guide our MAS design as it is a comprehensive, practical and easily implementable methodology specifically built for designing agent systems. The Prometheus method consists of (i) a system specification phase, (ii) an architectural design phase and (iii) a detailed design phase. From the first phase we obtain the functional specification by decomposing the system. In the architectural design phase we build upon this work by defining the agents within the MAS by grouping the functionalities and assessing the cohesion and coupling of the agents. From this analysis we found seven different agent types. We summarize our findings on the agent types below:

- **Demand Manager**
  The Demand Manager agent is responsible for retrieving and logging all cargo arrival and departure data, including all external systems (e.g. TMS and on-board computers) and brings this into the MAS.

- **Parking Manager**
  The Parking Manager assigns parking slots to all arriving and departing cargo and also for the AGVs when idling. Assigning parking slots to empty trailers on the terrain itself is also part of the functionality of this agent.

- **Vehicle Scheduling Agent**
  The Vehicle Scheduling agent assigns AGVs to transportation requests.

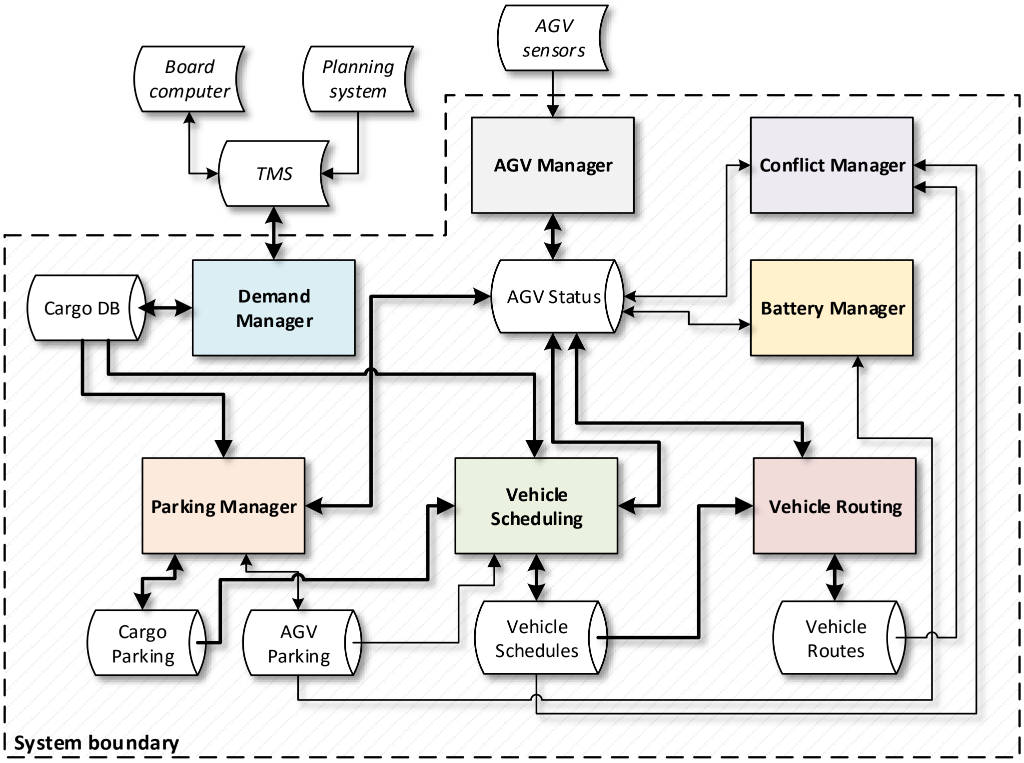
- **Vehicle Routing Agent**
  The Vehicle Routing agent determines the routing for all AGVs.

- **Battery Manager**
  The Battery Manager is responsible for effective charging schedules for all AGVs.

- **Conflict Manager**
  The Conflict Manager resolves all possible conflicts between AGVs and maintains a conflict-free environment by making stop-and-go decisions.

- **AGV Manager**
  The AGV Manager processes all data to and from the AGV controller and thus maintains the AGV status during system operation.

We summarize the overall design of the MAS, the coupling between the agents and the connections with the environment in the figure below.



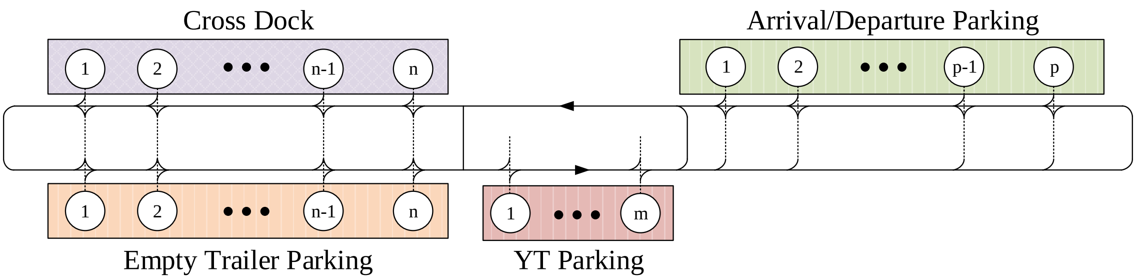
In the final design phase of the Prometheus Method we build upon the interactions as well as the agent descriptors from the previous design phase to further specify the behavior of agents. First of all the capabilities that are needed for all agents to fulfill their functionalities are determined. After this, all agents are narrowly described. This results in an agent overview which gives the most detailed view of an agent. It includes its capabilities, plans, all external databases connected, percepts responded to as well as the interaction with other agents using messages. All capabilities are further specified in capability overviews.

Building upon these capabilities we use the AGV framework of Le-Anh & De Koster (2006) to define all agent intelligence and to connect the design of the MAS with the physical design of the stackyard of the cross-dock at our pilot location. Using this framework we first determine the system requirements such as the stackyard layout, the pick-up and drop-off locations and the material flows. The most important requirements and assumptions of the pilot location are summarized on the next page.

Summary of the pilot location:

- 15.000 m^2 cross-dock (300x50x10 m);
- 150 loading docks;
- 100 parking slots;
- The cargo consists only of standard-size semi-trailers (2- and 3-axle);
- Integration of container movements of the container terminal at the pilot location is subject to further research;
- All cargo is (un)loaded from the rear;
- Forecasted vehicle movements: 400/day;
- A clear physical boundary between AGVs and manual operations is necessary;
- No humans should be involved in the handling process;
- The AGVs are based on yard tractors made by Terberg Benschop;
- We assume that the doors of the semi-trailers can be opened from the inside of the cross-dock;
- We assume that the yard tractors can autonomously (de)couple semi-trailers and are charged wirelessly and autonomously.

Within the AGV framework we first propose a design for the guide-paths on which the AGVs are allowed to drive and implement this at the pilot location. The figure below exemplifies how the guide-paths fit within the layout of the pilot location.



To summarize the physical design: at the Arrival/Departure Parking, truck drivers drop-off and pick-up semi-trailers where *p* parking slots are available. From this point on the autonomous YTs take over the handling at the terminal and use the guide-paths to dock, park and move around semi-trailers. The cross-dock has *n* docks and the parking opposite to it also has *n* parking slots. The YT parking is used for charging YTs or when YTs are idling, and is strategically positioned between the main areas to decrease the response time of a YT. The YT parking has *m* parking/charging slots and is equal to the number of YTs deployed.

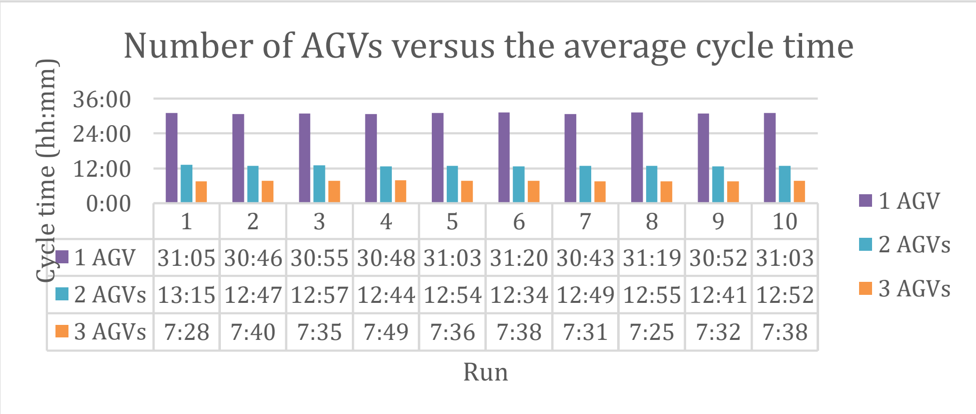
Based on the decision capabilities of every agent we develop the intelligence every agent should have in order to fulfill these capabilities. We summarize our findings on these capabilities per agent below:

- **Demand Manager**
  This agent does not require complex algorithms, but instead provides the link between external systems (e.g. TMS) and the MAS. It is however of utmost importance that all data required is readily available and up-to-date.
- **Parking Manager**
  We use a *nearest-available* rule for all parking areas. Arriving and departing semi-trailers are assigned to the first available (i.e. not occupied) parking slot. This results in an ordered priority list where parking slot 1 is more favorable to parking slot 2 and so forth.
- **Vehicle Scheduling Agent**
  A very common way of scheduling within MASs is using an auction mechanism where AGVs compete for orders. The Vehicle Scheduling agent initiates a proposal

when a new job enters the system via the Demand Manager (e.g., pick-up at cross-dock). All AGV agents evaluate this proposal and send back a bid. Based on a bid evaluation function the winner is announced. We use the auction mechanism as described in Mes, van der Heijden, & van Harten (2007).

- **Vehicle Routing Agent**
  The routing agent determines which route the AGV should take given the pick-up and drop-off location of the job. We use the shortest-path method and determine *a priori* the shortest paths between all nodes using the guide-path defined above.
- **Battery Manager**
  The Battery Manager uses an opportunity charging strategy as defined by McHaney (1995). Whenever an AGV is idling it is send to the AGV parking. The Battery Manager also monitors the battery level of all AGVs and asks the Vehicle Scheduling agent to schedule a charging job whenever the battery is below a certain threshold.
- **Conflict Manager**
  This agent is responsible to avoid collisions, congestion and deadlocks. We use a priority list to make stop-and-go decisions based on the current status of the AGV. When two or more AGVs want to use the same arc at the same time, the Conflict Manager evaluates which AGV has priority based on the current jobs the AGVs are processing and stops the AGV with the lowest priority.
- **AGV Manager**
  The AGV Manager has an important role within the MAS as it feeds information about the AGV to the system (e.g. current position and battery level). It furthermore uses the bid calculating function as described above to respond to the proposals of the Vehicle Scheduling agent.

From the design of the MAS we build a conceptual simulation model in which we outline the problem situation, the modelling objectives, the model inputs and outputs, the scope and level of detail of the model content (i.e., the MAS) and identify all assumptions and simplifications. From this conceptual model we build a simulation model using Technomatix Plant Simulation. We build a flexible simulation model to be able to quickly tailor the model to different case studies. For example, we can easily adjust the number of docks, the number of YTs, the number of parking slots and the dimensions of the guide-paths. We verify our model using (i) code debugging, (ii) model reviewing, (iii) providing demonstrations to the stakeholders, (iv) comparing the assignment of the YT to a semi-trailer with the actual YT picking it up and (v) watching the animation of the semi-trailers moving along the guide-paths. From this analysis we conclude that the simulation is an accurate translation of the conceptual model. We validate our model using white-box validation (i.e., to validate the behavior of every agent individually) and black-box validation (i.e., to validate the overall behavior of the model). The figure below shows the results of one of the validation techniques with ten simulation runs. From this figure we see a clear decrease in cycle time when the number of AGVs increases and conclude that our simulation model is valid and that a minimum of three AGVs is required for an acceptable cycle time.



This research lays down the fundamentals of the MAS for the planning and control of the autonomous vehicles at our case study within INTRALOG. Due to the broad nature of this foundation, we make several simplifying assumptions and demarcations along the way and thus this research has its limitations. We therefore recommend the following extensions of this research, both from a theoretical as a practical point-of-view:

Theoretical

1. Having built a valid simulation model, the experimental design of the simulation model still needs to be done. The analysis of the output data should provide answers to the number of AGVs needed and assess the impact on Key Performance Indicators (KPIs) for the system, such as costs, utilization, throughput time, travel time and waiting time using various scenarios.
2. We have focused on the cross dock of the pilot location in our case study. This pilot location will also feature a container terminal and the integration of the container movements with the semi-trailer movements is an interesting and promising extension of this research.
3. The MAS design should be applied to more case studies to validate the genericity of the MAS.
4. More research is required on which vehicle guidance and orientation system is suitable for our automated Material Handling System.
5. Extension of the simulation model by assessing the impact of different vehicle parking strategies on system performance.
6. Extension of the simulation model by assessing impact of different vehicle scheduling strategies on system performance.
7. Extension of the simulation model by assessing impact of different battery charging alternatives on system performance.
8. Extension of the simulation model by assessing impact of different vehicle routing strategies on system performance.
9. Extension of the simulation model by incorporating uncertainty of arrival times, driving times and handling times.
10. Extension of the simulation model by incorporating multi-stage bidding of the vehicle scheduling agent.

Practical

1. The design of an autonomous coupling device between the AGV and semi-trailer.
2. The design of an autonomous battery charging system.
3. The design of a dock door and corresponding apron space to facilitate automatic docking.
4. The design of an IT infrastructure for the MAS and the interfaces between external systems.
5. Further research is required on how to mature the MAS from conceptual/simulation to pilot testing/implementation, including the identification of all practical requirements.

PREFACE

The thesis that lies in front of you is the result of a seven year knowledge-driven journey through the corridors of the University of Twente where I studied Industrial Engineering and Management on both bachelor and master level. This dissertation has been written to finalize my master program and to receive the degree of Master of Science. During the better part of a year I was happy to meet so many different people within the INTRALOG project while writing my thesis and happy to work so closely with them. Numerous people have put their faith in me and helped me to get this thesis to a level which I alone would never have reached. To this extent I would first like to thank my lead supervisor Peter Schuur from the University of Twente for putting his trust in me and helping me to get involved into INTRALOG from the very beginning. I have fond memories of our many meetings, company visits and discussions. I would also like to thank my second supervisor Martijn Mes from the University of Twente for taking this thesis to a higher level and helping me with the simulation study. Furthermore a work of thanks goes out to the employees of the Automotive department of the HAN University of Applied Sciences and particularly to Lejo Buning who provided a workspace for me at the HAN and guided me throughout my master assignment as project leader of INTRALOG. I would also like to express my gratitude to all the partners involved in INTRALOG with whom I've met, discussed and laughed. They all helped me with my research, in particular Karel Kural of the University of Eindhoven with whom I've discussed regularly, Frank Rieck of the Rotterdam University of Applied Sciences who organized a guided tour at the APM Terminals at the Maasvlakte 2 and Abhishek Kumar of Terberg Benschop who provided information on their yard tractors. A special thanks goes out to the employees of Rotra, in particular to Pleun Nagtegaal, Gerard Roelofsen and Roy Gerritsen, who all contributed to this work by providing very valuable insights in their current practices and supporting me by sharing their views on the future of logistics. Last but not least I would like to thank my family, friends and fellow students for their input, support and appreciation. Many of them have unwittingly helped my research by lending an ear in often late-night discussions on semi-trailers, multi-agent systems and yard tractors.

Berry Gerrits

Enschede, April 2016

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

A/D	Arrival/Departure
AGT	Automated Guided Truck
AGV	Automated Guided Vehicle
BM	Battery Management
CFP	Call for Proposal
CNP	Contract Net Protocol
CR	Conflict Resolution
DB	Database
DC	Distribution center
DM	Demand Management
FIPA	Foundation for Intelligent Physical Agents
GUI	Guided User Interface
HAN	University of applied sciences Arnhem & Nijmegen
INTRALOG	Intelligent Truck Applications in Logistics
KPI	Key Performance Indicator
LHV	Longer Heavier Vehicle
LSP	Logistic Service Provider
MAS	Multi-agent System
MHS	Material Handling System
P/D	Pick-up and drop-off
PDT	Prometheus Design Tool
PM	Park Management
TCO	Total Cost of Ownership
TMS	Transport Management System
UT	University of Twente
VR	Vehicle Routing
VS	Vehicle Scheduling

LIST OF FIGURES

1 INTRODUCTION

This chapter gives an introduction to this thesis. Section 1.1 outlines our research motivation. Section 1.2 describes our research design. Section 1.3 describes our pilot location. Section 1.4 gives a summary and the thesis outline.

1.1 Research motivation

The logistic sector, being one of the largest economic sectors in the Netherlands, is continually looking for new innovations to increase operational efficiency, reduce fuel consumption and emissions and to address social aspects like (road)safety and comfort for all people involved in logistics. Ambitious projects across the globe (e.g. Google's Driverless Car and Ford's ParkAssist) have taken off to gain insight and knowledge in automatic driving, to explore business cases, as well as to address societal (willingness to relinquish control of the car) and political (legislation) challenges. Also the political climate in the Netherlands is changing to assist those who wish to experiment with automatic driving (Autonieuws, 2015)

To identify the potential of automatic driving within the logistics domain, the Intelligent Truck Applications in Logistics (INTRALOG) project has been started. Within this consortium, different business partners (DAF, Port of Rotterdam, Rotra and Terberg Benschop), universities (Eindhoven University of Technology, University of Twente), universities of applied sciences (Rotterdam, Arnhem & Nijmegen) as well as branch partners (CarrossieNL and Automotive Centre of Expertise) are joining forces to show the advantage of automated guided vehicles (AGVs) docking semi-trailers at distribution centers (DCs) and automated intermodal traffic. An overview of all consortium partners is shown in Figure 1-1. The goal of this four year project is to contribute to the opportunities of autonomous driving in the commercial transport sector.

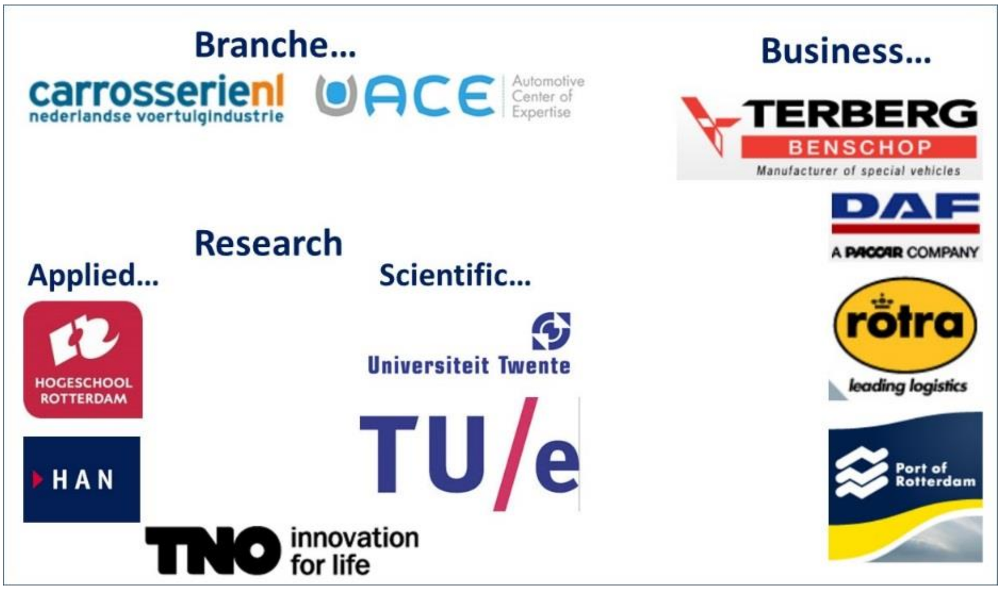


FIGURE 1-1 CONSORTIUM PARTNERS INTRALOG

This project is therefore directly in line with the recently published HTSM Automotive Roadmap 2016-2020 where Smart Mobility is mentioned as one of the key challenges. Three relevant main drivers for achieving automated driving are 'Active Safety' (reducing accidents caused by human errors), 'Efficiency' (the increase of transport system efficiency) and 'Comfort' (freedom of the user when automated systems are active) ("HTSM Automotive Roadmap 2016-2020," 2015).

Especially the latter is an interesting business opportunity for logistic service providers, such as Rotra to decrease waiting time and maneuvering time at DCs and thus to overcome problems with the driver's tachograph as they have to comply with European Law regarding driving times and rest periods.

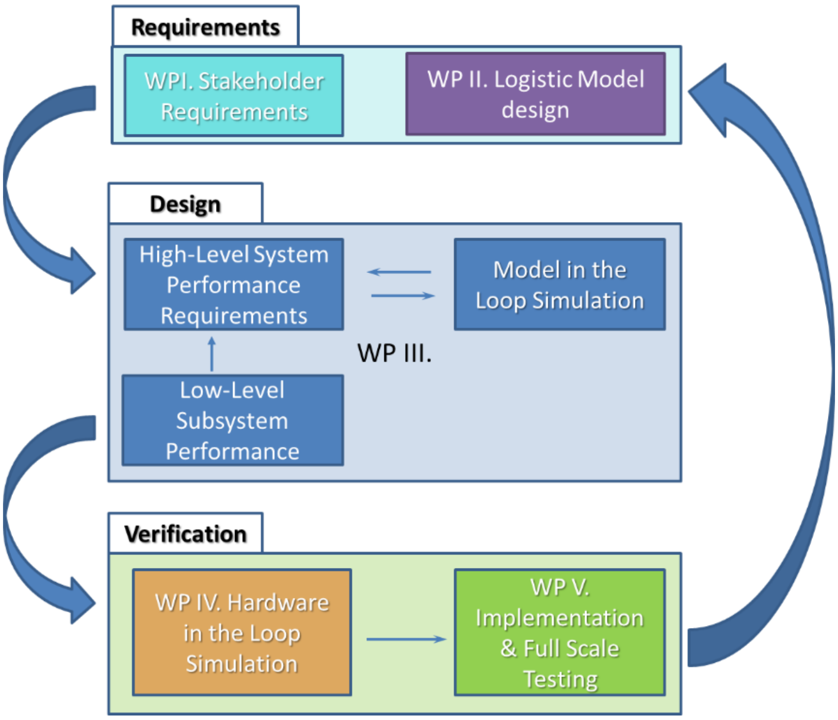


FIGURE 1-2 INTRALOG WORK PACKAGE BREAKDOWN

1.2 Research design

The design of this research has multiple aspects. We first position our research in Section 1.2.1 and define our objectives and research questions in Section 1.2.2. Section 1.2.3 describes our research approach and Section 1.2.4 outlines our contribution to the literature. We conclude with our demarcation and assumptions in Section 1.2.5.

1.2.1 Research setting

The INTRALOG project has been broken down into different work packages as shown in Figure 1-2. One of the crucial work packages is the design of a logistic model to facilitate AGV operations and the interaction with its environment. The University of Twente (UT) is the leading partner in this work package and is supported by the HAN University of Applied Sciences, Port of Rotterdam and Rotra. This work package has to deliver a stable and robust planning and control system using a multi-agent system (MAS). A MAS consists of several independent and autonomous control units (agents) which pursue their own interests and interact with other agents in the environment. One of the key challenges is the configuration of the agents such that their self-interested behavior yields a near-optimal solution for the network as a whole.

The network is defined as a closed transportation network consisting of a fixed number of pick-up and drop-off locations. Automated Guided Vehicles (AGVs) transport the goods (e.g. semi-trailers) between these locations using a certain track layout. The network is closed as no AGV can enter or leave the system, even when idling. Similarly to Ebben (2001) this *Automated Transportation Network* is defined as a fully automated system for the transportation, loading and unloading of goods using AGVs supported by a MAS which functions as a planning and control system. Figure 1-3 shows such a network and its system boundaries.

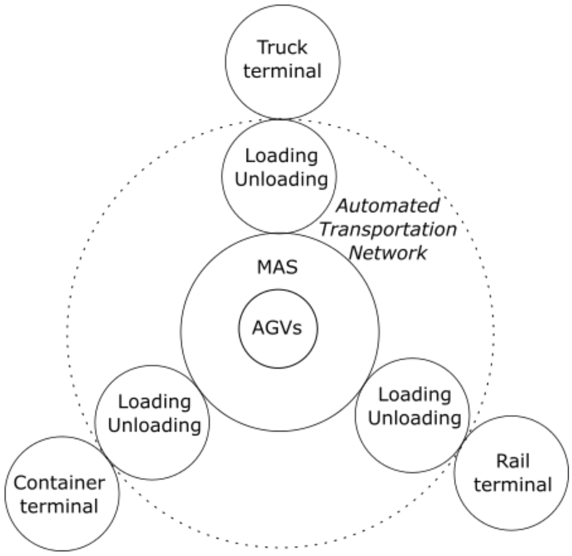


FIGURE 1-3 THE SYSTEM BOUNDARIES OF THE AUTOMATED TRANSPORTATION NETWORK

The boundary of the system is at the terminal level. The internal operations of a terminal such as workforce scheduling, goods inspection, internal transportation and warehousing require complex systems and would tremendously increase the complexity of the system as a whole. In this thesis the system ends at the loading/unloading of goods at the terminal. In the case of a truck terminal this would be the docking and undocking of a semi-trailer, which is the focus of this thesis. The succeeding operations at the terminal fall beyond the system boundary. Although this representation shows only a truck,- container,- and rail terminal, it could also be extended to for example airport terminals or multimodal terminals.

The purpose of this thesis is to develop a *generic* system in the sense that is suitable for different kinds of terminals, varying in size or layout, using parametrization or extensions. The loading/unloading operations at the system boundary would vary in practice. For example at a truck terminal, picking up a semi-trailer and rear-ward docking it is a complex task which require a well-designed control systems. The control system would be different (or even simpler) at a container- or rail terminal where other physical activities are needed. Although different in nature, these are considered as equal loading and unloading operations in this thesis. In the rest of this work we define *cargo* as freight using any mode of transport which can be decoupled from the transporting entity. For example, semi-trailers and ISO-containers fall within this definition, but rigid trucks do not, as the cargo cannot be decoupled from the truck. Automated Guided Trucks (AGTs) need to be used to automate box trucks (see Section 1.2.5). The AGVs within the system perform the following main actions:

1. **Pick-up**
   Arriving cargo is decoupled from the truck and picked-up by an AGV at a predetermined location (see Figure 1-4).
2. **Move**
   The AGV moves the cargo to a predetermined drop-off location.
3. **Docking**
   The cargo is docked rearwards in to the loading dock.
4. **Undocking**
   After the terminal has finished unloading/loading the cargo, the AGV picks up the cargo.
5. **Drop-off**
   The cargo is dropped off at a predetermined location such that a truck is able to pick-up the cargo.

1.2.2 Research objectives and research questions

The research goal of this thesis, in compliance with the INTRALOG project, is as follows:

> ***The development of a generic automated planning and control system based on agent technology for the pick-up and docking of semi-trailers by means of AGVs in a collision- and conflict free environment in such a way that it yields a cost-effective, near-optimal solution.***

Research questions

To accomplish our research goal we have defined a number of research questions. For each research question a small description is provided including the planned approach how to answer the research question.

1. **How should the system be decomposed into functional specifications?**
   First of all the system decomposition has to be done. This has important consequences for the rest of the project, as this gives a bound to the rest of the design. It is important to define the system as generic as possible. In this way the project can start out small and in a later stage can be extended to fit other applications (e.g. intermodal traffic hubs) or larger scale systems. The system should then be decomposed in functional specifications. This will be done in collaboration with the lead supervisor from a theoretical point-of-view and with business partners from a practical point-of-view to ensure the system is as close to reality as possible and as time permits. Next to that, we conduct a literature study on multi-agent systems and AGVs, and more specifically will look for and study related cases.

2. **How should the agents be designed in terms of functionalities?**
   When the system functionalities are defined, the next step is to design the agents such that they can carry out these functionalities. Decisions have to be made on how many

different kinds of agents the system should have and how many of each should be implemented. Again a literature study on agent design has to be conducted.

3. **How should the agents interact with each other and the environment?**
When the agents are designed, the interaction protocol is the next important step. Agents have to communicate and cooperate with other agents in the environment for the system to work. Therefore the information exchange has to be investigated and the question which information should be exchanged by whom at which moments in time has to be answered. Again literature and case studies have to be analyzed and lessons learned from these can be of valuable input to this project.

4. **Which decision capabilities should the different types of agents have?**
As agents are self-interested goal-driven entities they have to make decisions to ensure that their goal is reached. This can lead to discrepancies between the goals of different agents and thus some mechanism of group decision making, bargaining and arguing has to be designed. This has to be designed in such a way that it is autonomous and yields a near optimal solution for the system as a whole. Different kinds of decision making protocols have to be studied and the most suitable has to be chosen. Also the decision making capabilities of each agent has to be researched in order to create an intelligent and well-balanced balanced system.

5. **How to build a valid simulation model for the multi-agent system?**
When the MAS has been fully designed, the model has to be analyzed by building a simulation model. This way it can be shown that the model is stable and robust. Preferably real-life data is used from one of the consortium partners to validate the model and to establish credibility among all partners. The simulation should eventually provide answers to the number of AGVs needed and assess the impact on Key Performance Indicators (KPIs) for the system, such as costs, utilization, throughput time, travel time and waiting time using various scenarios. The preferred type of simulation is a discrete-event simulation study with agent technology embedded.

This research thus has the following deliverables:
1. Functionality specification;
2. Allocation of functions to agents;
3. Establishing interaction protocols between agents;
4. Decision-making capabilities of the agents;
5. A multi-agent discrete event simulation model;
6. Verification and validation of the simulation model;
7. Report containing all parts of the multi-agent system.

1.2.3 Research approach

The research questions above and the order in which they are stated are in compliance with the research that is carried out within work package II of INTRALOG. We chose to incorporate all these research questions to build a solid basis comprehending all relevant parts of the multi-agent system, instead of focusing only on a single aspect of the MAS. With this approach we lay down the ground foundation of work package II and encourage other researchers to complement this work. The order of the research questions reflects the natural order of a design study. First we translate the system to a functional specification and then we design agents to reflect these functionalities. We then design the environment for the agents and how they interact with each other. From there we define the decision capabilities of all agents to incorporate a certain amount of intelligence in every agent. Finally we put our MAS to work in a discrete-event simulation study to verify and validate the MAS design. To perform a structured design we first conduct a literature review on multi-agent systems with a specific focus on design methodology. The functionality of the MAS should correspond to a realistic scenario for which we use a case study provided by INTRALOG. This case study focuses on a new to be build cross-dock of consortium partner Rotra. We actively involve different consortium partners, especially Rotra and Terberg Benschop, to incorporate the expertise of practitioners and to fully comprehend the functionalities the MAS should have, but also to address issues arising from practice. Their knowledge is also a valuable asset when building and parametrizing our simulation model with which we validate our design of the MAS.

1.2.4 Research motivation

The novelty of this research is the design and realization of a full-size, large-scale AGV system capable of handling heavy and large cargo, particularly semi-trailers. Compared to already existing AGV systems at for example container terminals, this research does not solely focuses on the transshipment of goods, but uses a cross-dock for the consolidation of cargo either in a one-to-many or many-to-one fashion. This research could be extended to incorporate multimodal cross-docks and thus provides a generic solution to AGV systems for the handling of large-scale and heavy freight at logistic hubs. To our best knowledge this particular topic has not been researched in the literature to this date. The use of industry partners provides a different point-of-view to address issues arising from practice.

1.2.5 Demarcation and assumptions

To cope with the complexity of a large-scale system including many practical considerations it is important to narrowly define the scope of this research. This includes demarcation and defining the assumptions including their validity.

Demarcation

1. Although the AGV is part of the automated transport system as shown in Figure 1-3, the AGV controller is part of work package III and thus not falls within the scope of this research. The AGV is the boundary between WPII and WPIII as it is an agent within WPII from a logistic point-of-view and an agent from a control point-of-view within WPIII. Therefore its functionality (maneuvering, acceleration, turning, etc.) is not part of this research, but the planning and control is.
2. The system only involves the outside operations at a logistic terminal. Therefore any internal operations such as loading/unloading is not part of the system. It is assumed that these operations take a certain amount of time, which is input to the MAS.
3. In the pilot study the only cargo considered are standard-sized semi-trailers.

Assumptions

1. An Automated Guided Truck (AGT) is assumed to be a real-life physical truck, which drives automatically and remains connected to the cargo. An Automated Guided Vehicle (AGV) is assumed to be a separate, stand-alone automated vehicle, which transports decoupled cargo. In contrast to an AGV, an AGT can leave the system. This thesis focuses on AGVs.
2. We assume that the layout and structure of the terminals is given. Although some design issues can be addressed, this is bounded by the space of a given layout.
3. A simplified behavior of the AGV maneuvering is assumed. The driving time of an AGV depends on the distance travelled and the docking maneuver is assumed to have a fixed duration depending on the cargo.
4. It is assumed that all information necessary for the real-time operation of the MAS is available and accessible. This includes the expected arrival and departure time of cargo.
5. The loading/unloading duration is assumed to be stochastic variable with a yet to determine probability density function.

6. The number of semi-trailers needed to be unloaded before a departing cargo can be loaded (transshipment) is assumed to be a stochastic variable with a yet to determine probability density function.
7. We assume that all AGVs are identical and are always functional (i.e., they do not break down).
8. Recharging or refueling locations are identical to AGV parking locations.
9. Maintenance is carried out during system down-time.
10. The doors of the semi-trailer can be opened from the inside of the DC.
11. It is assumed that an AGV can automatically couple and decouple a semi-trailer, which takes a predetermined amount of time.

1.3 Pilot location

Consortium partner Rotra has plans to build a new multimodal cross-dock near Velp, The Netherlands. Rotra owns all the property (around 13 ha) necessary to build a new logistics center which is strategically positioned between the A348 motorway and IJssel river. Rotra is planning to build a multimodal logistics center where road and water transport meet in a 15.000 m^2 cross-dock with 150 docks. The location is close (within 10 km) to the current cross-dock and container terminal in Doesburg, but due to space restrictions and continuous growth, Rotra wants a new and bigger location. The container terminal at Velp is scheduled to be built together with the cross-dock and should support inland vessels up to CEMT-Class Va (110 m x 11,40 m). Rotra has assigned this location to be the pilot location for INTRALOG. The location is a perfect pilot location as there are no public roads (no regulatory restrictions), there is enough space (no capacity restrictions), the possibility to extent the system to multimodal (road and water) and there are no design restrictions (the terminal has yet to be built). In this way it is possible to design the terminal such that it fits seamlessly with automated transport. A first sketch of the location is shown in Figure 1-5. The sketch is projected on the current infrastructure and terrain. On the top, the A348 motorway is shown and on the bottom the IJssel river. The green line represents the cadastral boundaries. The terrain features a multimodal cross-dock (300 m x 50 m) with 150 loading docks, a container terminal, a LNG fueling station, a garage for up to six vehicles, 200 regular parking lots (shown in red) and 100 truck parking lots (shown in black).

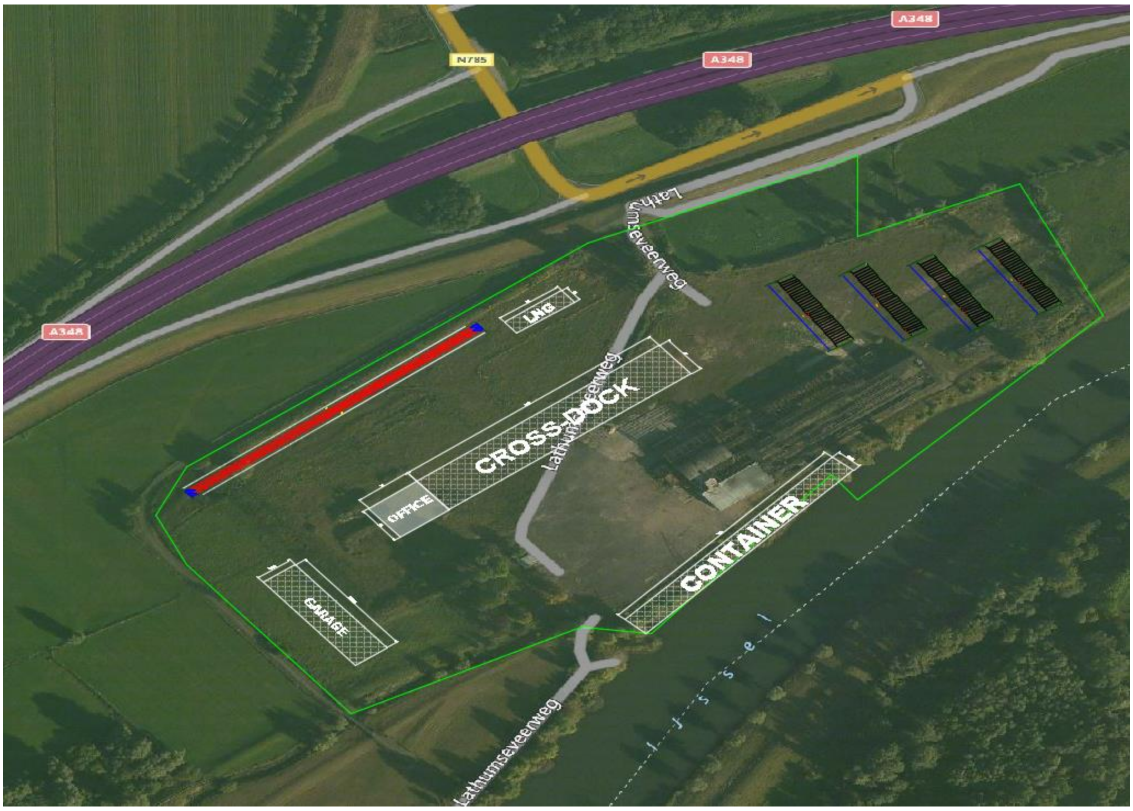


FIGURE 1-5 SKETCH OF PILOT LOCATION VELP

1.4 Summary and thesis outline

This thesis focuses on the design of a multi-agent system for the planning and control of autonomous vehicles for the automatic handling of semi-trailers at a cross-dock. We first decompose the system into a functional specification from which we define agents, how they interact and how the environment in which they are situated is shaped. We put our MAS design to work using a case study for which we build a simulation model. The remainder of this thesis is structured as follows. In Chapter 2 we introduce Multi-Agent Systems. In Chapter 3 we describe the design of the MAS. In Chapter 4 we introduce our case study. Chapter 5 presents an AGV framework which we link to the MAS. In Chapter 6 we present a conceptual simulation model and validate this model in Chapter 7. Chapter 8 contains our conclusions and recommendations for future research.

2 INTRODUCING MULTI-AGENT SYSTEMS

This chapter provides background material on Multi-Agent Systems (MASs). Section 2.1 describes the origin and history of Multi-Agent Systems. Section 2.2 provides definitions and characteristics of MASs. Section 2.3 highlights some key applications of MASs. We conclude with a summary in Section 2.4.

2.1 History

The concept of Multi-Agent Systems (MASs) is not new. Literature about MASs have been around since the mid-1990s and have in the recent years gained popularity in scientific research as well as in the industry. Wooldridge (2009) mentions five important trends in the history of computing:

- Ubiquity
- Interconnection
- Intelligence
- Delegation
- Human-orientation

Ubiquity is nowadays still a continuing trend and means the constant increase of computing power against lowering costs. This has led to increased processing power in many applications and devices which would have been unimaginable or uneconomic in the past. This omnipresence of technology and smart devices has transitioned our society and businesses from closed and ill-informed to open and well-informed.

The *interconnection* of technology has led to large, complex distributed systems which typically are more powerful than closed, stand-alone systems. For example, smartphones enable us to share and process information at anytime, anywhere with anyone across the globe. But also business delegate their processing power across the globe using high-speed connections to solve large-scale problems which simply would not be possible to solve alone.

Increasing *intelligence* of all systems also provides new opportunities for many people and businesses. For example, navigation systems are able to pro-actively alter current routes to find faster routes to the users destination based on current traffic information or road closures. This proactiveness makes systems more intelligent than solely active or reactive systems.

Another important trend is the continuing *delegation* of control to software systems. Nowadays we trust software to perform better than humans in certain safety-critical tasks. Take for example an automatic pilot on an aircraft or all advanced electronics in passenger cars to assists us or take control of our driving (e.g. adaptive cruise-control).

The final trend is the *human-oriented* view of programming and software. Increasingly software systems are user-oriented with user-friendly Guided User Interfaces (GUIs) and real-time interaction with the user. A good example is Google’s “OK Google” or Apple’s “Siri” which processes voice input and displays information based on the user’ input using intelligent algorithms to determine *what* the user is saying and what the user is *trying to say*.

These five trends pose challenges to software developers as systems become more and more intelligent, interconnected and need to act on behalf of the user. To resolve these problems, the field of multi-agent systems has emerged.

2.2 Definition and characteristics

The concept of multi-agent systems is relatively easy to grasp. The main idea is that an agent is a computer system which can act *autonomously* on behalf of its user to reach its design objectives. This means an agent can determine its own course of action instead of following pre-determined executable code. A multi-agent system has multiple agents, most likely different

types of agents, with different kind of goals and objectives, that are able to interact with each other, typically using certain protocols or messing passing within a computer system. In order to interact, agents must be able to cooperate, coordinate and negotiate with each other, similarly to our everyday lives (Wooldridge, 2009).
Although there exist different kinds of definitions of agents, where the lowest common denominator would be "proactive objects" (Parunak, 1998), the most commonly adapted definition throughout the literature is the following:

"An agent is a computer system that is *situated* in some *environment* and that is capable of *autonomous action* in this environment in order to meet its design objectives." (Wooldridge, 2009)

From this definition the following characteristics can be derived (Farahvash & Boucher, 2004; Padgham & Winikoff, 2005):

- *Situated*: Agents are situated in an environment which is most often dynamic, unpredictable and unreliable.
- *Autonomous*: Agents can work independently of other agents as well as without interference of human intelligence.
- *Interactive*: Agents affect other agents or may be affected by other agents. Communication and interaction is key in the design of a MAS.
- *Intelligent*: Some level of intelligence must be present in an agent to guide decision-making in order to fulfill its design objectives in an efficient and effective way.
- *Flexible*: Agent systems should consider different environmental states and configurations and act properly to different situations.
- *Proactive:* Agents should continue to pursue their goal(s) over time and these goals are persistent.
- *Robust:* An agent must be able to overcome failure by continuing to achieve a goal despite of previous failed attempts.

2.3 Applications

MASs are applied in many different domains due to the universal applicability of agent technology as many systems resemble one or more of the characteristics shown above. Typical applications are intelligent manufacturing systems, work flow management, e-commerce, autonomous control systems such as air traffic control and traffic telematics (Oliveira, Fischer, & Stepankova, 1999). MASs are also used for entertainment and as learning tools such as the multi-player soccer game TOS (freely available at https://sourceforge.net/projects/soccer/). MASs are also used in the field of logistics for planning and scheduling (Gath, Herzog, & Edelkamp, 2014) and agents in general are in useful within this field as they can represent many parts of a supply chain, such as vehicles, machines, distribution centers or packages (Becker, Syyed, Wenning, & Görg, 2007). In this thesis we apply a MAS to the handling of cargo at terminals. Figure 2-1 shows the primary processes at a distribution center where semi-trailers are handled. It is easily verified that these primary processes are also applicable to other types of terminals by modifying the transport mode specific terminology.

2.4 Summary

This chapter has shown the origin of multi-agent technology based on five software engineering trends and has presented a definition of a multi-agent system from which we derived the characteristics of a MAS. We conclude with a discussion on the applications of MASs.

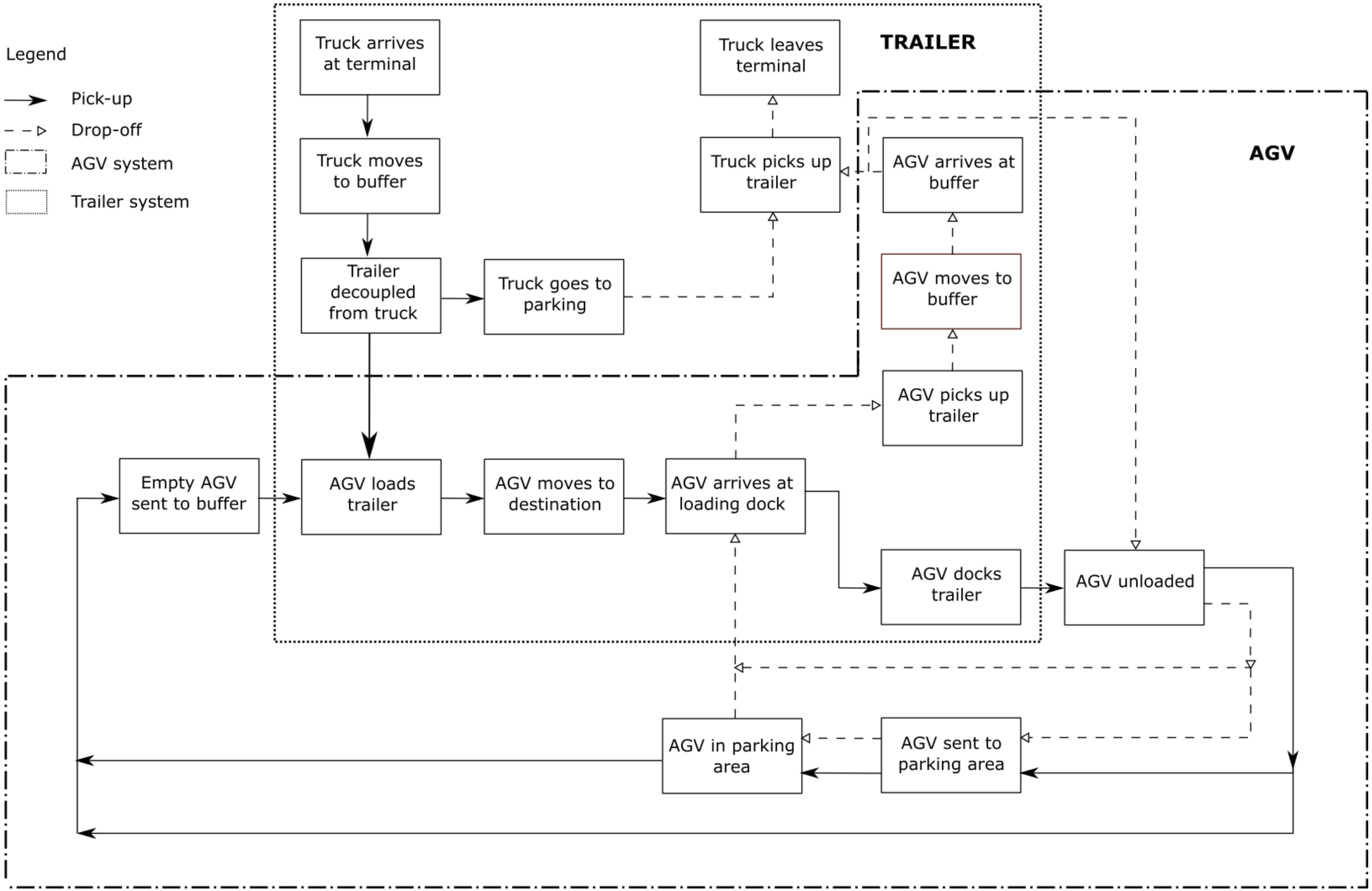


FIGURE 2-1 PRIMARY PROCESSES OF AN AUTOMATED GUIDED TRANSPORTATION NETWORK

3 MULTI-AGENT SYSTEM DESIGN

This chapter contains the design of the Multi-Agent System. In Section 3.1 we provide an introduction to MAS design. Section 3.2 provides a literature overview. In Section 3.3 we provide a design methodology and put this to work in Section 3.4, 3.5 and 3.6. Section 3.7 gives an overview of the design.

3.1 Introduction

Building multi-agent systems (MASs) is a complicate and iterative process. Large-scale systems can contain agents in the hundreds operating at real-runtime, perceiving, communicating, negotiating, decision-making and executing. This complex system has to be designed properly to not only include all functionalities necessary to keep the system running and accurate, but also to make the system scalable and generic (i.e., applicable to a wide range of situations), where a situation may vary between any forms of logistic hubs. The key challenge is to design a generic system such that it can be configured (e.g. adjust parameters or introduce new instances of generic building blocks) to represent a new situation and thus not to design a *one-off* system which is only applicable to a certain architecture or testbed. In order to systematically address this challenge, a comprehensive and detailed methodology should be used to guide us through the process of breaking down the high-level objective of *'building an intelligent multi-agent system'* into smaller, easier to grasp chunks such that the system can be fully understood and then designed. This should not only include high-level steps such as 'specify the system boundaries', but also mid- and lower level steps providing enough detailed guidelines to design and implement a software package based on intelligent agents.

3.2 Literature overview

Agent-Oriented Design and Analysis is intended to gain understanding of a system and to guide the process of designing these systems. The methodologies for this analysis and design can be roughly categorized into two groups (Wooldridge, 2009):

- Based on object-oriented (OO) development, by either extending or adapting existing OO methodologies to suit agent-oriented software.
- Adaptations of knowledge engineering or other techniques.

The comprehensive book of (Wooldridge, 2009) provides an overview of multiple methodologies, including AAII, Gaia, Tropos, Prometheus and Agent UML. It is not the intension of this thesis to fully review and compare all available methodologies. For the interested reader, more information, including references to the original developers of all methodologies can be found in (Wooldridge, 2009) and (Padgham & Winikoff, 2005). However, as noted above, in this work a methodology is needed that fully comprehends the design and analysis ranging from high- to low level steps. This is where for example Agent UML falls short as it is merely a language protocol for object-oriented modelling. Although standard object-oriented models are able to design a MAS, there seems to be lacking some essential parts related to the *nature* of agents (e.g. agents can be proactive). The Prometheus methodology, however, is specifically designed to include agent-technology and aims to be complete, providing everything necessary for specifying and designing agent systems (Winikoff & Padgham, 2004). Also throughout the literature Prometheus is used to design multi-agent systems, including systems which feature AGVs (e.g. Erol, Sahin, Baykasoglu, & Kaplanoglu, 2012; Xing et al., 2012). Another important practical aspect is that the Prometheus Design Tool (PDT) is freely available, speeding up the design process and has built-in completeness and consistency checks. Prometheus is also based on industry standards like case scenarios, UML sequence diagrams, AUML (itself an extension of UML) and the Rational Unified Process (RUP) (Padgham & Winikoff, 2005).

It has been argued that Prometheus is a comprehensive, practical and easy implementable methodology specifically built for designing agent systems. As such, the Prometheus method has been used in this thesis to guide the process of designing the multi-agent system. The Prometheus method is introduced in the following section.

3.3 The Prometheus Method

The Prometheus Method consist of three main phases: system specification, architectural design and detailed design. More specifically the phases are defined as follows:

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* phase looks at the internals of each agent and how it will accomplish its task within the overall system." (Padgham & Winikoff, 2005)

Figure 3-1 depicts the phases of the Prometheus methodology including all artifacts and interactions.

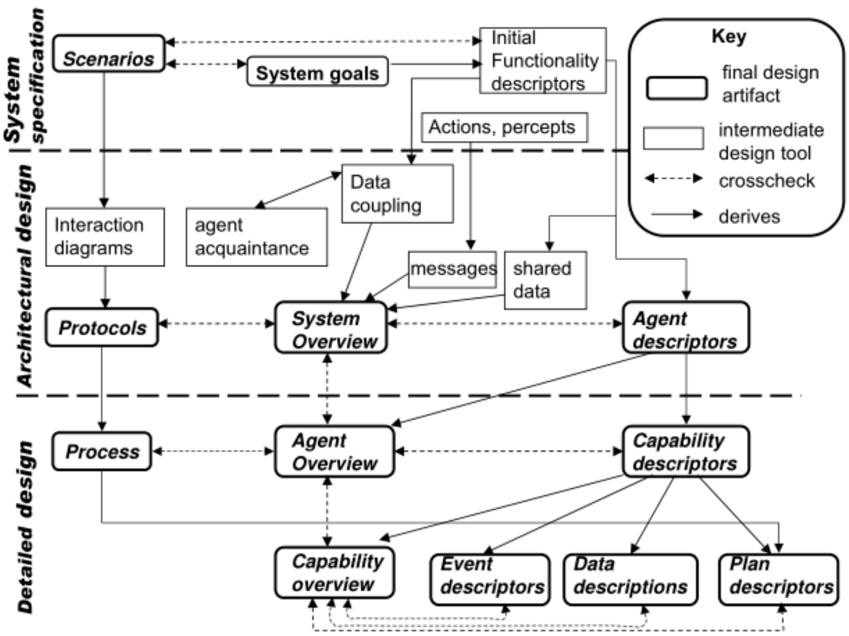


FIGURE 3-1 THE PHASES OF THE PROMETHEUS METHODOLOGY (ADAPTED FROM (PADGHAM & WINIKOFF, 2005)

The remainder of this chapter follows the Prometheus methodology. The three main phases are distinguished in the next few paragraphs and all the design decisions within the phases are covered in their respective sub-paragraphs. The realization of all elements is supported by the practical guide book from Padgham & Winikoff (2005) on developing intelligent agent systems.

3.4 System specification

This paragraph outlines the system specification phase of the Prometheus methodology. Section 3.4.1 describes the goals of the system and Section 3.4.2 continues with the system functionality. Section 3.4.3 contains the development of scenarios. Section 3.4.4 outlines alternative scenarios and we conclude with an interface description in Section 3.4.5.

3.4.1 System goals

The first step is to indicate the *goals* of the system – what should the system do. This is the starting point of the system specification phase and forms a basis for the rest of the design. In the case of INTRALOG, the following description captures the system:

> *A **generic automated planning and control system** based on agent technology for the **pick-up and delivery (docking)** of semi-trailers or containers by means of **AGVs** in a **collision- and conflict free environment** in such a way that it yields a **cost-effective, near-optimal solution.***

From this initial description we can define (and later refine) a list of system goals. The bold parts in the above description highlight the system main goals. To accomplish these goals, several sub-goals have been defined. After rearranging and grouping similar (sub)goals, a goal decomposition diagram (goal overview) has been made using the PDT, shown in Figure 3-2.

Note that *generic* does not mean that the MAS should be applicable to all planning and control systems. It is merely to state that the MAS should be applicable or extendable to a number of deviations in the scope of this project. That is to say, different configurations of DC's, with more or fewer loading docks or a different lay-out. Or easily configured such that the system is also applicable to container terminals or other similar pick-up and delivery systems involving (multimodal) cargo. It is therefore not intended to design a system which could for example be used to control AGVs for warehouse operations or production lines.

3.4.2 System functionalities

After specifying the system goals, the functionalities of the system have been defined. Functionalities are chunks of behavior that are needed to achieve the system goals. This includes the grouping of goals, but also percepts, actions and data related to the behavior of the functionality (Padgham & Winikoff, 2005). Defining functionalities and system goals are intertwined and is an iterative process. When specifying functionalities, it may seem logical to add, group or delete goals. Also functionalities can be split when they encompasses too much behavior. A functionality descriptor should not be more than one or two coherent sentences to describe the entire behavior of the functionality. The following (generic) functionalities have been defined (see also Figure 3-3 and Appendix 1):

1. **Demand Management (DM)**
   This functionality monitors inbound/outbound cargo, obtaining information about expected arrival/dispatch time and cargo description.
2. **Park Management (PM)**
   This functionality assigns pick-up and drop-off locations to all cargo and AGVs.
3. **Vehicle Scheduling (VS)**
   This functionality decides when, where and which AGV should pick-up and drop-off cargo or empty trailers.
4. **Vehicle Routing (VR)**
   This functionality determines the route such that the AGV can pick-up and drop-off cargo or empty trailers
5. **Conflict Resolution (CR)**
   This functionality monitors all AGV movements and makes sure there are no collisions and resolves conflicts.
6. **Battery Management (BM)**
   This functionality determines when and where AGVs should be refilled or recharged.

Note: Any functionality related to the actual movement of the AGV belongs to the AGV controller and is not part of the MAS.

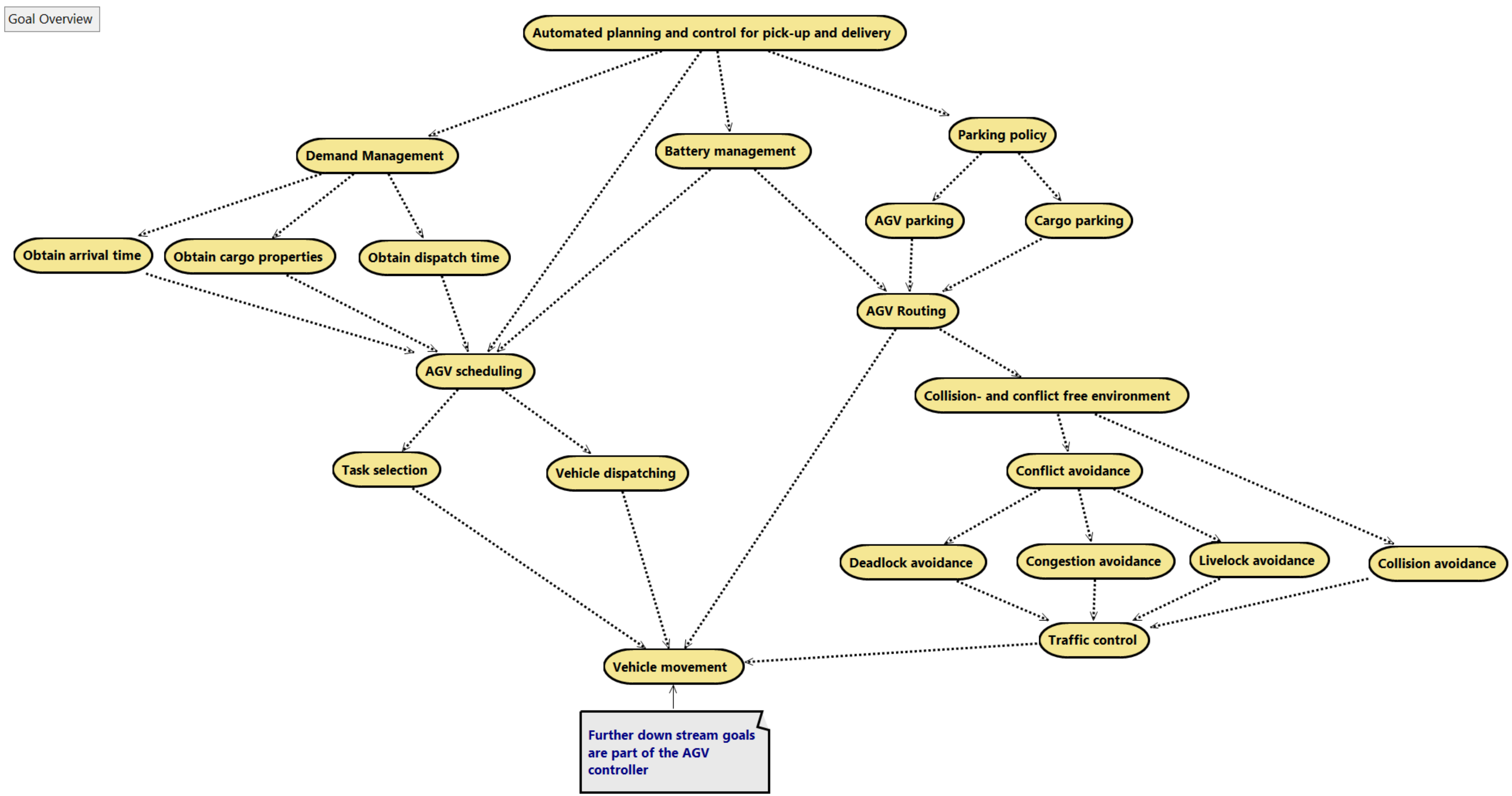


FIGURE 3-2 SYSTEM GOAL OVERVIEW

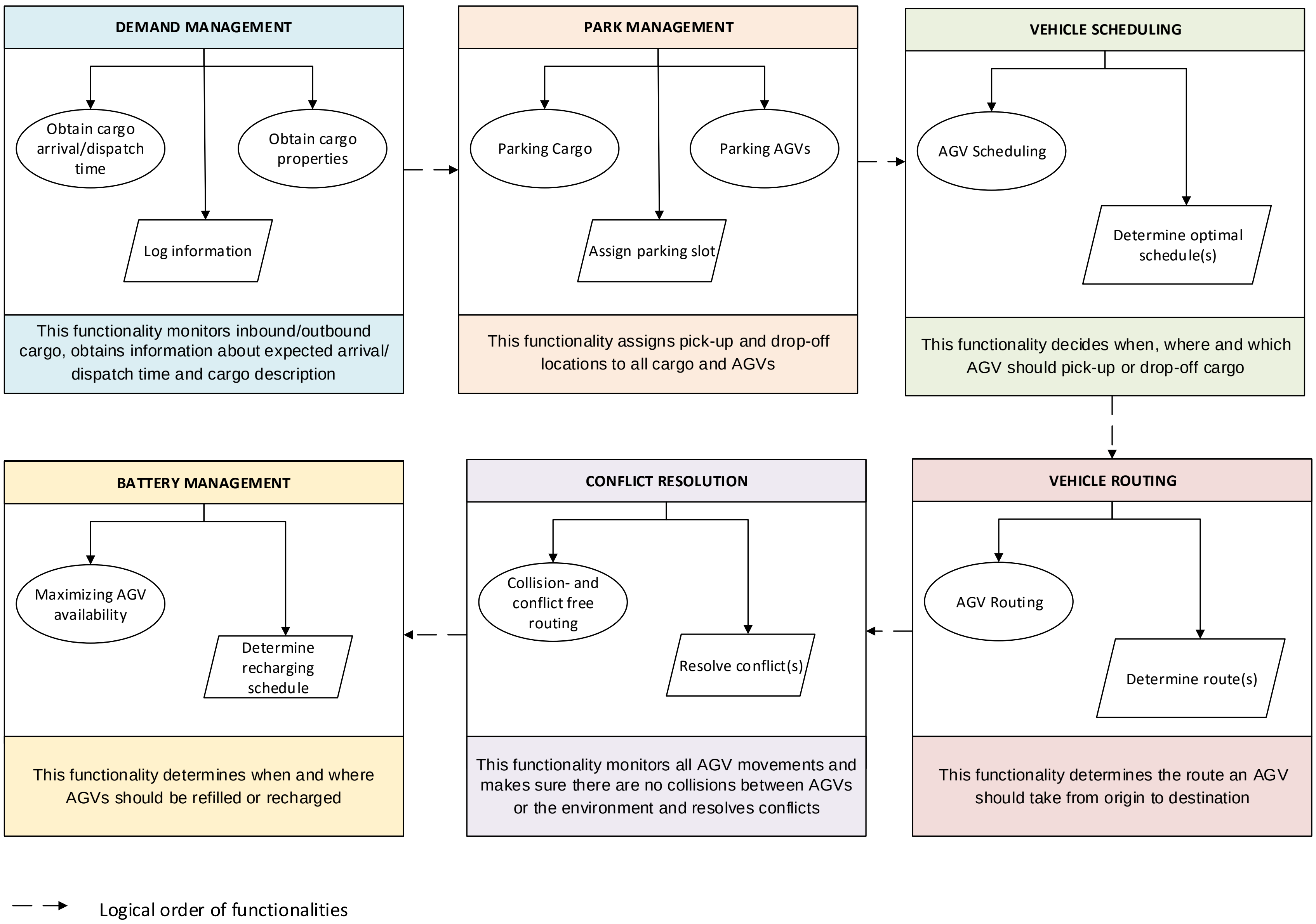


FIGURE 3-3 FUNCTIONALITY OVERVIEW

3.4.3 Scenario development

The goals and the functionalities of the system have been defined and one of the final parts of the system specification phase is to develop scenarios. Scenarios are states the system can be in and describe *which* functionality does *what* when this scenario occurs. Scenarios thus include goals, actions, percepts and others (e.g. waiting for something to happen). Linked to this scenario development is the data used and produced by all functionalities. The following scenarios have been defined:

- Cargo arrival
- Cargo departure
- AGV idle
- AGV low battery
- AGV conflict

The first scenario assumes *inbound* cargo and the second *outbound* cargo. The third scenario occurs when an AGV is idling and thus has to be parked somewhere. The fourth scenario occurs when an AGV has a low battery and thus has to be recharged at the appropriate location. Note that recharging can also be replaced by refueling (in case of a combustion engine). The final scenario is one that can happen anytime during system run-time and thus also during other scenarios, namely AGVs who are in conflict. This could for example happen when two or more AGVs want to take the same route at the same time. Appropriate measures have to be taken when this occurs such that congestion and system dead- or livelocks are avoided. Also during the scenario development phase an iterative process has been used between system goals, functionalities and scenarios. Table 3-1 shows the scenario of a cargo arrival. The other scenarios can be found in Appendix 2.

The scenario starts with a percept which perceives any inbound cargo close to the terminal. The boundaries of the proximity of the cargo can be variable and depends on the location of the terminal and the mode of transport. The scenario could for example start when cargo is 1 hour away from the terminal. The Demand Management (DM) functionality then obtains the expected arrival time of the cargo (from the board computer of the transporting entity) and also obtains the properties of the cargo (type of cargo, identification number, content, origin, destination, etc.). An important property of the cargo is the destination within the terminal (e.g. dock number 13 or stack yard 1, row 4). This information originates from the planning system of the terminal and thus falls beyond the scope of the MAS. We assume that this property is known up-front, or at least before any scheduling activity takes place. When the system knowns which cargo arrives and when, it requests a parking slot (Park Management). A parking slot will then be assigned to the cargo and it will then determine when the cargo will arrive at this parking slot and will thus ready to be processed further. Note that a parking slot can also be a location in a stack yard.

At the same time this information is passed on to Vehicle Scheduling (VS), which determines which AGV should pick-up the cargo. VS thus uses the information on the assigned parking slot, the time the cargo will be ready for pick-up, the status of all AGVs, which includes information on the current location, whether is it busy, idle or charging and in the latter two cases, the expected time it will be available again. When an AGV is selected, Vehicle Routing (VR) takes over by determining the route the AGV should take. One of the inputs is the guide-path design (tracks AGVs are allowed to take). VR determines the route between the current location of the AGV and the parking slot of the cargo and then the route between the parking slot and the cargo destination. The Conflict Resolution (CR) functionality then updates the AGV status (from idle to busy). This status update is incorporated in the CR functionality as this functionality monitors all AGV movements and prevents collisions and congestion. Finally VR sends all relevant information to the AGV controller, which then moves the AGV at the right time to the right location. As noted before the AGV controller falls beyond the boundaries of the MAS, but to be complete it is part of this scenario. One possible interfering scenario is the AGV conflict scenario. Suppose two AGVs want to be on the same place at the same time, the CR functionality then decides which AGV has priority and gives stop and go information to the AGV controller. Any stop decision has of course impact on the total processing time of the AGV as waiting time is introduced. Depending on the size of the terminal, this waiting time can be marginal compared to the total driving and handling time. In the AGV conflict scenario, the AGV status is updated when waiting time is introduced such that the rest of the system is informed. At some point in time the AGV reaches its destination and all succeeding operations are completed. The AGV controller gives back a signal to the MAS. The AGV Status is updated (from busy to idle) and the cargo properties are updated (e.g. being unloaded).

TABLE 3-1 CARGO ARRIVAL SCENARIO

Scenario: *Cargo arrival*

Key for functionality and data abbreviations

DM	Demand Management
PM	Parking Management
VS	Vehicle Scheduling
VR	Vehicle Routing
CR	Conflict Resolution
BM	Battery Management
AG	AGV Controller
Car. Arr. T.	Cargo Arrival Time
Car. Prop.	Cargo Properties
Car. Park	Cargo Parking Slot
Park. Avail.	Parking Availability
AGV Stat.	AGV Status
AGV Sched.	AGV Schedule
G.P. Des.	Guide-Path Design

	Step type	**Step**	**Functionality**	**Data used and produced**
1	Percept:	*New inbound cargo*		
2	Goal:	Obtain arrival time	DM	Car. Arr. T.
3	Goal:	Obtain properties	DM	Car. Prop.
4	Action:	Request parking slot	PM	Car. Arr. T. Car. Prop.
5	Goal:	Assign parking slot	PM	Car. Arr. T. Car. Prop. Park. Avail. Car. Park. Park. Avail.
6	Goal:	Determine parking arrival time	PM	Car. Arr. T. G.P. Des. Car. Arr. T.
7	Action:	Request pick-up	VS	Car. Arr. T. Car. Prop. Car. Park.
8	Goal:	Determine AGV schedule	VS	Car. Park. AGV Stat. AGV Sched.
9	Action:	Request route	VR	AGV Sched.
10	Goal:	Determine AGV Route	VR	G.P. Des. AGV Sched. AGV Stat. AGV Route
11	Goal:	Update AGV Status	CR	AGV Stat. AGV Stat.

12	Action:	Send info to AGV Controller	VR	AGV Sched. AGV Route
13	Other:	Wait for AGV to be done		
14	Scenario:	*AGV conflict*		AGV Routing AGV Stat.
15	Percept:	*AGV is done*		
16	Goal:	Update AGV Status	CR	AGV Stat. AGV Stat.
17	Goal:	Update Cargo Properties	DM	Car. Prop. Car. Prop.

The cargo departure scenario and the cargo arrival scenario are very similar and only the first step is different. It can be changed to the following percept such that it is applicable to departing cargo. The rest of the scenario remains the same.

TABLE 3-2 ALTERATION FOR DEPARTURE SCENARIO

	Step type	**Step**	**Functionality**	**Data used and produced**
1	Percept:	*Cargo ready to leave*		

Having defined the scenarios in the way we did, we have established generic building blocks for the MAS capable of controlling all kinds of cargo at any type of terminal whether the cargo is arriving or leaving. The remainder of the scenarios can be bound in Appendix 2.

3.4.4 Alternative scenarios

The scenarios mentioned above all describe a single sequence of steps. As the MAS should be generic as possible, minor deviations from these steps are needed depending on the configuration of the MAS (truck terminal, container terminal, …). For example the parking slot assignment (steps 4-6) can be skipped in a container terminal if the cargo is directly loaded onto the AGV from the barge. When cargo is first stored in a stack yard before further processing, steps 4-6 remain applicable, but should be rephrased to *stack yard slot* which fits better to the situation. Also the *AGV Idle* scenario checks whether it is smart to send the AGV to the parking lot, depending the AGV schedule. Sometimes it may be more efficient to keep the AGV idling if in the near future a new task awaits instead of driving back and forth to the parking lot. In this modification the scenario ends at the first few steps and there is no need to further carry out the steps this particular scenario.

Similarly many deviations can occur depending on the environment the MAS is operating in. As shown it is easy to modify or delete the steps in a scenario to make it suitable for a wide variety of cases. We don't intend to capture all these modifications here, but it is worth noting that by adding/deleting or modifying scenarios, the MAS can be tailored to any specific case.

3.4.5 Interface description

The MAS is situated in a dynamic and stochastic environment, which affect or is affected by the MAS. As such it is important to define all the relations the MAS has with its environment and vice versa. This could either be input from the environment (percepts) or output from the MAS to affect the environment (actions). Directly in line with the MAS-environment interaction is the question which data should be exchanged. In the system specification phase it is important to recognize all the data necessary, including any data external to the agent system.

Data related to the expected arrival/departure time of cargo is probably the most important external data to the system. Any expected arrival triggers the system to handle the cargo, including decisions on parking, AGV assignment, scheduling and routing. On the other hand, expected departures require similar actions but instead of inbound cargo, it handles outbound cargo. The expected arrival time of a piece of cargo should come from some sort of board computer of the transporting entity. Although this information is readily available in practice, often this information is not coupled to other software systems, such as planning software. This forms a major challenge to not only extract relevant data from the board computer, but also incorporating it in such a way that it is usable for the MAS. The same goes for departing cargo, which could depart at a fixed point in time or when all the cargo is loaded. In both cases some kind of trigger should be given to the MAS to notify that the cargo is ready to leave. The feasibility of the system depends largely on the availability and usability of the external data necessary for the system to function properly. In lower level design steps these issues are more explicitly handled.

3.5 Architectural design

In this phase, the agents of the system are defined and also how they interact with each other in order for the system to function properly. This is done by first grouping similar functionalities as defined in the system specification phase to develop cohesive and loosely coupled agents. This is done to find a balance between the degree of dependency between agents and the complexity of a single agent. In the following sub-sections, agent types, interaction protocols and the overall system structure are defined.

3.5.1 Grouping functionalities

The first step is to group functionalities into agent types. Agent types are lay-outs of how an agent should look like whereas an agent instance is the actual run-time agent. There are multiple ways of grouping functionalities, depending on the situation at hand, but in this case the grouping is largely based on the functionality overview presented in the previous chapter. This overview already groups some functionalities based on the goals specified in the system specification phase. One or more of the following considerations have been taken into account when grouping functionalities:

- The functionalities are related and common sense dictates that these belong to each other;
- The functionalities require the same data sources. This minimizes message passing between agents;
- Functionalities require the same hardware platform;
- The cardinality of the functionalities is the same within an agent. For example at run-time one agent instance is needed per active AGV whereas the AGV routing is required to be a single agent within the system.

The standard software engineering criteria of *coupling* and *cohesion* are also used to assess proper grouping of functionalities to agents. *Coupling* is defined as how much communication exists between agents and *cohesion* is defined as to which extent the goals of an agent are closely related (Padgham & Winikoff, 2005).

3.5.2 The boundary between the MAS and the AGV controller

An important design question is the incorporation of the AGV agent. Although we specified that the functionality of the AGV is part of the AGV controller (and thus omitted from the functionality overview), it is an important part of the MAS. This is exactly the boundary between the MAS and the AGV controller as all its status related data is necessary for the MAS to function properly but the controller of the agent is not. Therefore we design the AGV agent in such a way that it contains all relevant information needed for the MAS and thus also receives instructions from other functionalities (e.g. the route it should take), but the controlling part of the AGV is viewed as an external data source. In other words, the MAS tells the AGV what to do and the AGV tells the MAS what it is doing, but not telling how.

3.5.3 Data coupling

According to Padgham & Winikoff (2005) an important part of grouping functionalities and assessing its validity is using data coupling diagrams. These diagrams show which data is coupled to which functionality based on the functionality descriptors defined in the previous design phase. Directed links are shown between functionalities and data. Arrows pointing towards data indicate that data is written by the functionality and the other way around indicates the usage of data. Double-headed arrows indicate the usage and writing of data by the functionality. Bold arrows indicate important connections within the system. All data used and produced by the system must be in the diagram somewhere including any external data sources. This also functions as a consistency check between the first two design phases. To cope with the complexity of the diagram we do not show the data sources *within* a functionality (e.g. priority mechanisms or scheduling algorithms). The data coupling diagram is shown in Figure 3-4.

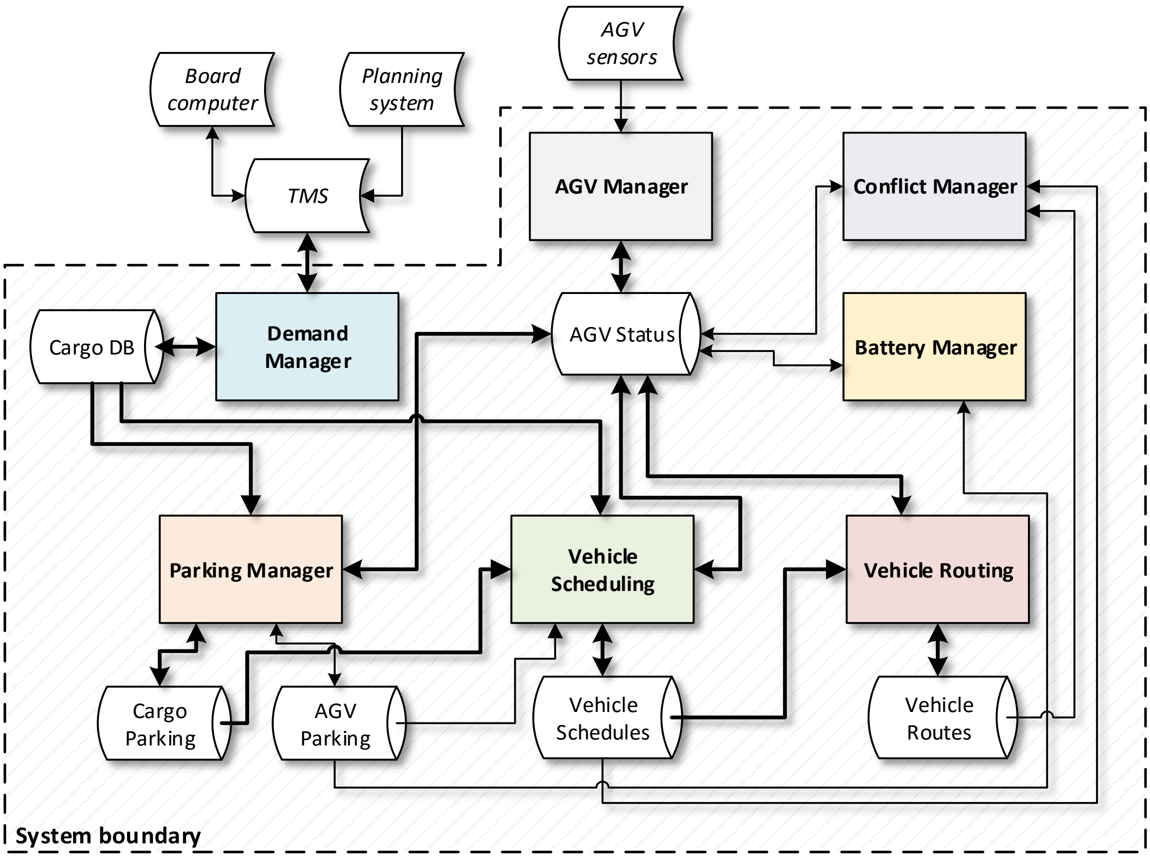


FIGURE 3-4 DATA COUPLING DIAGRAM

The main external database required for the MAS is the Transport Management System (TMS). This TMS contains information about inbound and outbound cargo, cargo properties (e.g. consolidation information), where the cargo should go at the terminal (originating from the planning system) as well as client-specific information. The expected arrival time of inbound cargo is obtained using the board computer of the transporting entity. A double-headed arrow is shown between the board computer, the TMS and the Demand Manager as it could be possible to send information to the driver from the MAS (e.g. which parking slot is assigned to a truck driver). Also the AGV sensors play an important external role as these sensors provide the data for the AGV status. Many agents communicate with the AGV agent to pass information on where to go, at what time and other status related information (e.g. on-route, loaded/unloaded and charging). All links originate from one or more scenarios as defined in Section 3.4.3. From this diagram we can clearly see seven agents emerging, similarly to the previously defined functionalities but in addition an AGV manager is also implemented.

3.5.4 Agent acquaintance

To further assess the coupling of functionalities, the agent acquaintance diagram can be used. This diagram shows all the agent types with links representing interactions with other agents, including cardinality information. We use this diagram to assess potential bottlenecks in the design by counting the maximum number of links an agent has. Also the cardinality of the agent types is important to evaluate possible run-time bottlenecks. If *n* agents have to communicate with *n* other agents, this could be a bottleneck. The degree of coupling is also determined using this diagram by comparing the maximum number of links with the actual number used. If *n* denotes the number of agents in the system, then $n*(n-1)/2$ is the maximum number of non-directional links between the agents. The link density is then defined by the division of the number of links and the maximum number of links. Figure 3-5 shows the agent acquaintance diagram.

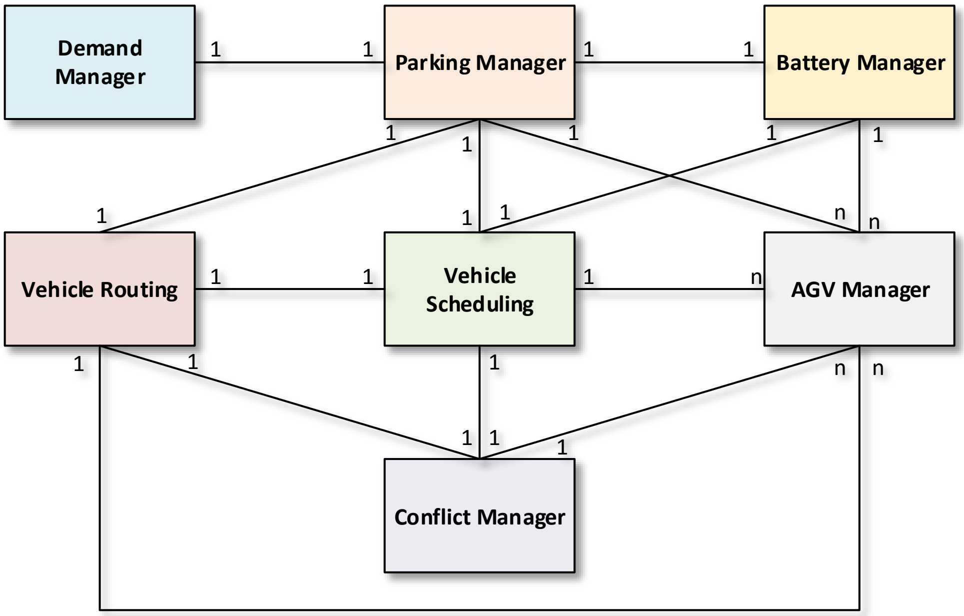


FIGURE 3-5 AGENT ACQUAINTENCE DIAGRAM

This configuration is moderately coupled, with a link density of 57% (12 divided by 21). Fortunately 7 out of these 12 have a cardinality of one both ways, indicating that communication only takes place between two agent instances, decreasing possible run-time bottlenecks compared to cardinalities of *n*. The AGV agent however has a lot of connections with the other agents and thus could lead to potential run-time bottlenecks. This is of course not preferred, but in all the cases only *one* of the other agent types communicates with *n* AGV agents. Most information flows in the system are directed *towards* the AGV agent, so most data is used by the AGV agent instead of produced by it. Also not all links are necessary for every scenario. For example the Conflict Resolution agent only needs to communicate with the AGV agent if there is some sort of conflict. We therefore decide to stick with this design, although moderately coupled, also keeping in mind that we do not expect that *n* will be an excessively large number in most practical cases.

In this design some decisions have been made about the cardinality of the agents. For example all in- and outbound cargo is represented by a single demand manager. It can be argued that in some applications an agent per *job* is more suitable for auctioning protocols. We have chosen to make these decisions on a somewhat more centralized level, incorporated in the demand manager. This is not only done to lower the coupling of the system, but also to keep the system more generic. For example the added benefit of an agent per container on a ship, competing to be serviced first could be counter-productive as it more likely that a centralized *container agent* is able to make smarter decisions on which container to service first as it has information of all containers and is thus capable of generating optimal schedules. The same goes for a parking agent per parking slot. It would probably not help the system *as a whole* when all parking slots compete for utilization. A more centralized single parking agent is able to assign parking slots to cargo with the goal of minimizing the travel time of the system as a whole. Depending on the application of the MAS, the cardinality of one or more agents may be changed when this seems more appropriate, but in the remainder of this thesis we work with the cardinalities as shown in Figure 3-5.

3.5.5 Agent descriptors

Before continuing to the interaction design of the agents, we first summarize all agents using agent descriptors. An agent descriptor is a compact overview containing all information of an agent, including a list of functionalities and goals from the previous design phase as well as interactions with other agents, which are derived from the scenario analysis. An example of the agent descriptor of the Demand Manager is shown below. All agent descriptors can be found in Appendix 3.

TABLE 3-3 AGENT DESCRIPTOR OF THE DEMAND MANAGER

Agent descriptor: Demand Manager	
Name:	Demand Manager
Description:	Obtains arrival and departure times of cargo as well as all relevant cargo properties
Cardinality:	One per system
Lifetime:	Ongoing
Initialization:	Obtain data from TMS
Demise:	Close open DB connections
Functionalities included:	Demand Management
Uses data:	Cargo DB, TMS
Produces data:	Cargo DB
Goals:	Obtain cargo arrival time, obtain cargo departure time, obtain cargo properties
Percepts responded to:	New arriving cargo, new departing cargo
Actions:	Log cargo arrivals and departures to cargo DB
Protocols and interactions:	Request parking slot with Parking Manager

Within all agent descriptors, many percepts, actions, protocols and interactions are defined. These more-or-less high-level descriptors are appropriate for this stage in the design. It must however be noted that during implementation, these percepts, actions and interactions need to be carefully defined and decisions have to be made how to access and write information, whether or not pre-processing is necessary and which protocols to use and which IT-infrastructure is appropriate. As these decisions cannot be made yet, mostly because every case study would be different, we leave these details to further research. We therefore skip two steps in the architectural design phase as shown in Figure 3-1; interaction diagrams and protocols. At this stage it is sufficient to have high-level descriptors of the percepts, actions, protocols and interactions. It is however important to be aware of all information needed for the system to work properly and to assess whether it is realistic to assume that all information is present or can be easily obtained when implementing the system. At this stage we assume that all data is available, accessible and pre-processed such that the system can function properly.

3.5.6 System overview

The final step of the architectural design phase is to develop the system overview diagram. This diagram represents the entire system including all agents, percepts, actions and protocols and how they are linked. This overview is the result of all design decisions thus far. Similarly to all previous diagrams, the direction of the arrow shows flow of information. The system overview diagram is shown in Figure 3-7 on the next page and Figure 3-6 shows the legend belonging to the diagram.

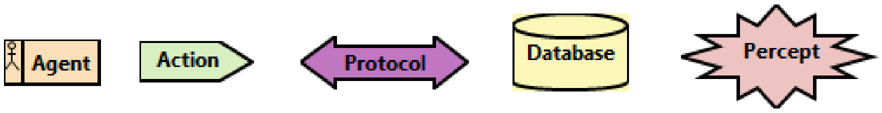


FIGURE 3-6 LEGEND OF SYSTEM OVERVIEW DIAGRAM

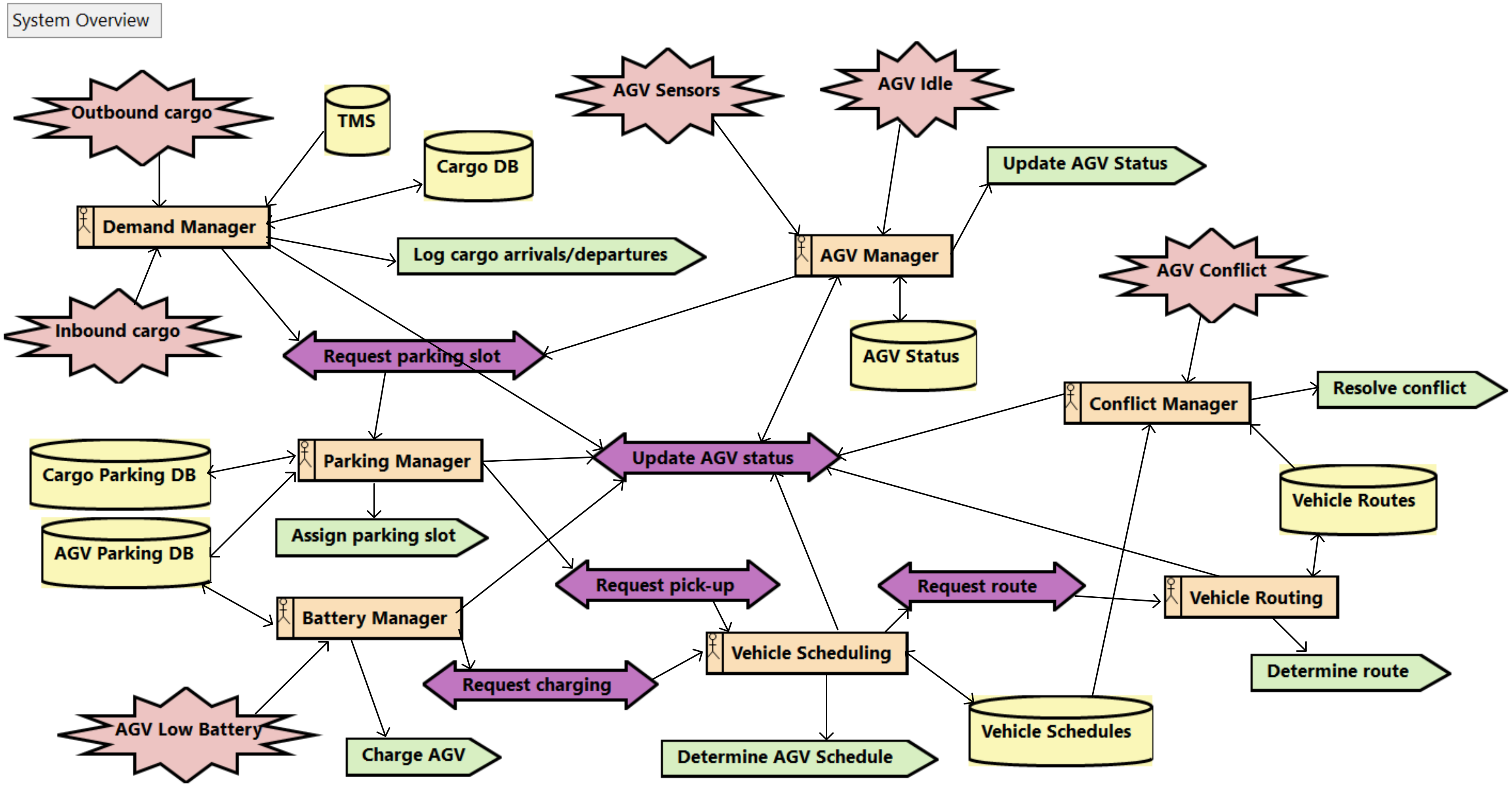


FIGURE 3-7 SYSTEM OVERVIEW DIAGRAM

3.6 Detailed design

The final design phase of the Prometheus Method is the detailed design phase and takes the agent overview, the interactions as well as the agent descriptors from the previous design phase to further specify the behavior of agents. First of all the capabilities that are needed for all agents to fulfill their functionalities are determined. After this all agents are narrowly described, including their capabilities, percepts responded to, actions to do, messages to pass and databases to use and write to. This results in an agent overview per agent.

3.6.1 Agent overview

The agent overview shows the most detailed view of an agent. It includes it capabilities, plans, all external databases connected, percepts responded to as well as the interaction with other agents using messages. All capabilities are further specified in capability overviews. In line with the previous phases we highlight and explain one agent and all other overviews can be found in Appendix 4. Figure 3-8 shows the agent overview of the Parking Manager. It can be seen on the left side that the agent has several inbound messages (originating from the Demand Manager) containing information on cargo arrivals and departures. These messages trigger the *manage cargo parking* capability of the agent and to fulfill its functionality *assign parking slot*. The agent uses the cargo parking DB to log all parking slots of all in- and outbound cargo. This data is used by other agents (e.g. as input to Vehicle Routing to determine the pick-up or drop-off point of cargo). The capability outputs the assigned parking slot via a message, which can be used to inform the TMS on where cargo is located. This information can also be shared with external transporting entities to let drivers know where to drop-off or pick-up cargo.

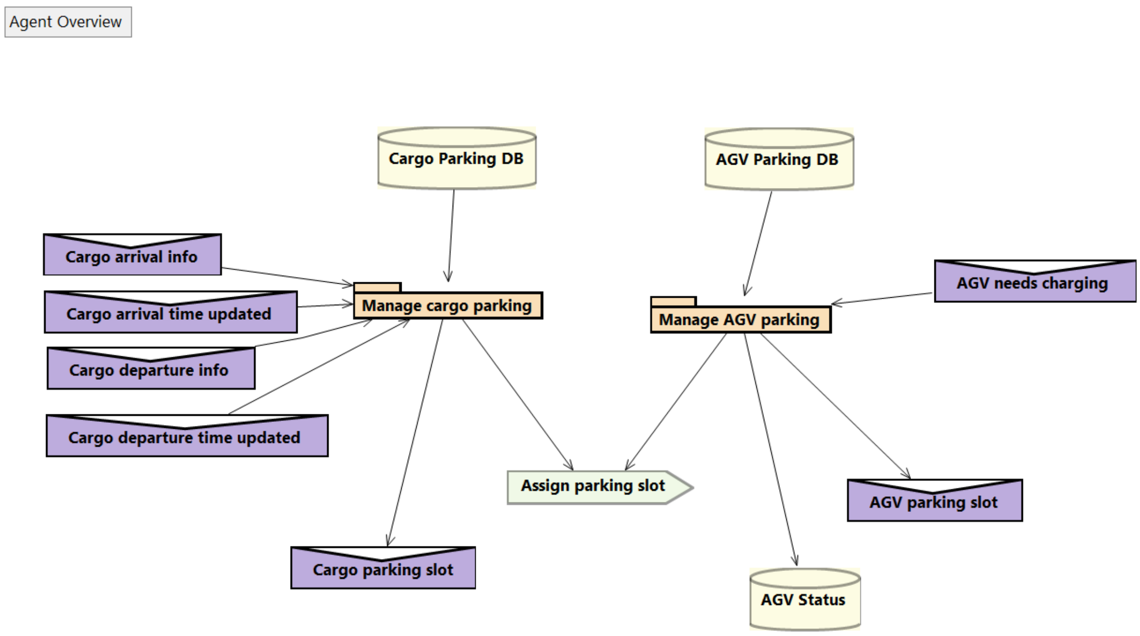


FIGURE 3-8 PARKING AGENT OVERVIEW

The right side of the overview shows the AGV parking capability, which is responsible for parking AGVs. The agent receives a message from the Battery Manager when an AGV needs to be recharged. It uses the AGV Parking DB to log all AGV parking data and it also uses the AGV status DB (to assess whether an AGV is idling, originating from the AGV Manager). The output of the agent is the AGV parking slot, which is used by the AGV Manager to maneuver the AGV to the appropriate location.

3.6.2 Capability overview

Both capabilities shown in Figure 3-8 are further specified in the capability overviews shown in Figure 3-9 and Figure 3-10. The manage cargo parking capability uses all inbound messages to determine the cargo parking slot using priority rules. It also uses all delay information to assess whether or not a new parking slot should be assigned to the cargo using some delay rules. All information is updated in their respective databases.

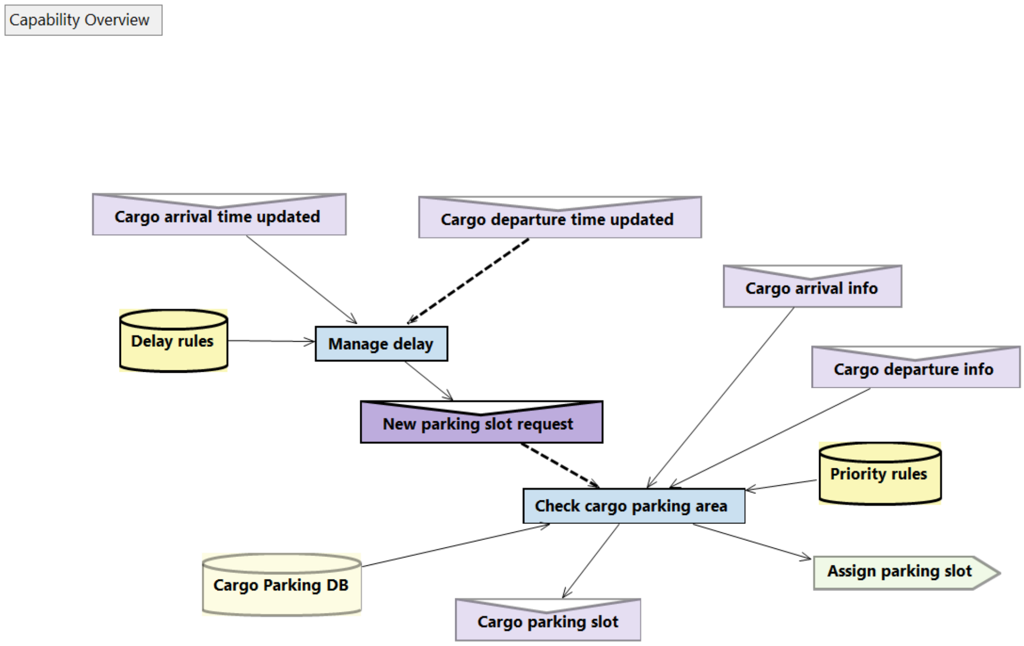


FIGURE 3-9 MANAGE CARGO PARKING CAPABILITY OVERVIEW

The AGV parking capability checks the AGV parking area when an AGV is idle or needs charging and assigns a parking slot using priority ruling. It uses the AGV Status database to know if and AGV is idling or not. If an AGV is idle it checks the AGV parking area and assigns a parking slot. When the AGV status changes to not-idle, it updates the AGV parking DB by removing the AGV from the parking slot. When an AGV is assigned to a parking slot it also passes a message with a request for a route to the assigned parking slot.

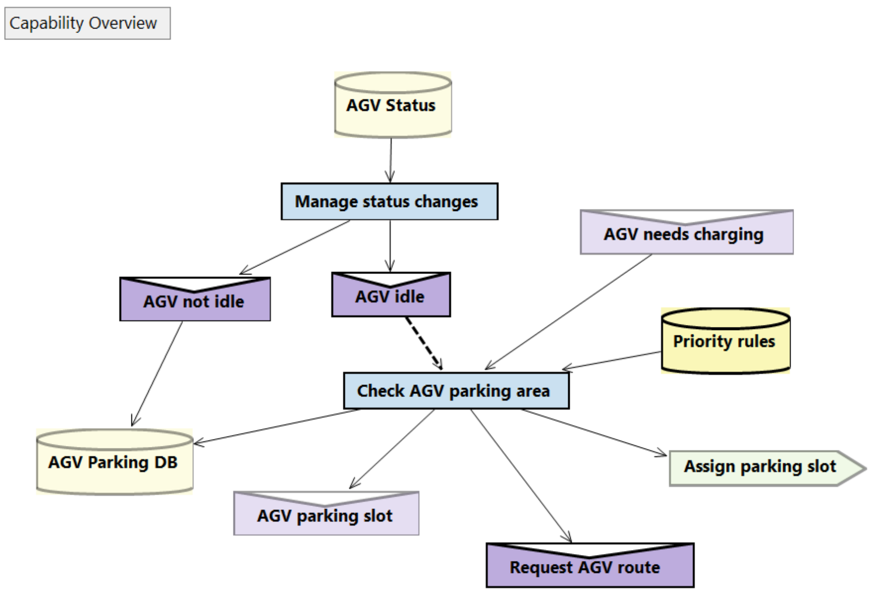


FIGURE 3-10 MANAGE AGV PARKING CAPABILITY OVERVIEW

3.7 Conclusion

This chapter has guided the process of designing the MAS using the Prometheus Method. This paragraph concludes the design and summarizes the agents. The MAS has seven different agent types all having their own capabilities and tasks. The following list gives an overview of all agents including a brief summary.

- **Demand Manager**
  The Demand Manager agent is responsible for retrieving and logging all cargo arrival and departure data, including all external systems (e.g. TMS and on-board computers) and brings this into the MAS.

- **Parking Manager**
  The Parking Manager assigns parking slots to all arriving and departing cargo and also for the AGVs when idling. Assigning parking slots to empty trailers on the terrain itself is also part of the functionality of this agent.

- **Vehicle Scheduling Agent**
  The Vehicle Scheduling agent assigns AGVs to transportation requests.

- **Vehicle Routing Agent**
  The Vehicle Routing agent determines the routing for all AGVs.

- **Battery Manager**
  The Battery Manager is responsible for effective charging schedules for all AGVs.

- **Conflict Manager**
  The Conflict Manager resolves all possible conflicts between AGVs and maintains a conflict-free environment by making stop-and-go decisions.

- **AGV Manager**
  The AGV Manager processes all data to and from the AGV controller and thus maintains the AGV status during system operation.

4 CASE DESCRIPTION: ROTRA AND LOCATION VELP

This chapter outlines our case study of consortium partner Rotra. We begin with an introduction in Section 4.1. Section 4.2 gives an overview of the fleet composition of Rotra. Section 4.3 provides insight in the current practices and Section 4.4 contains forecasts for the pilot location. Section 4.5 describes the physical system design. We conclude with a summary in Section 4.6.

4.1 Introduction

Consortium partner Rotra is a Logistics Service Provider (LSP) over 100 years old with a head office in Doesburg, The Netherlands. Rotra is a family-owned company and has over 860 employees in The Netherlands and Belgium. They specialize in worldwide forwarding by air, sea, rail and land and are currently investing in multimodal transport with a new container terminal in Doesburg, complementing their current cross-dock. Rotra wants to be ahead of the competition when it concerns innovation and thus is highly involved in the INTRALOG project. As stated in Section 1.3, Rotra plans to build a new multimodal cross-dock in Velp, which will be the pilot location for the INTRALOG project.

4.2 Fleet composition

The fast growing fleet of Rotra currently consists of approximately 70 trucks, 15 box trucks, 15 swap bodies, 10 regular trailers, 3 LHVs (Longer Heavier Vehicles) and 250 semi-trailers. The latter category is further decomposed in Figure 4-1 as this category is the focus of this research.

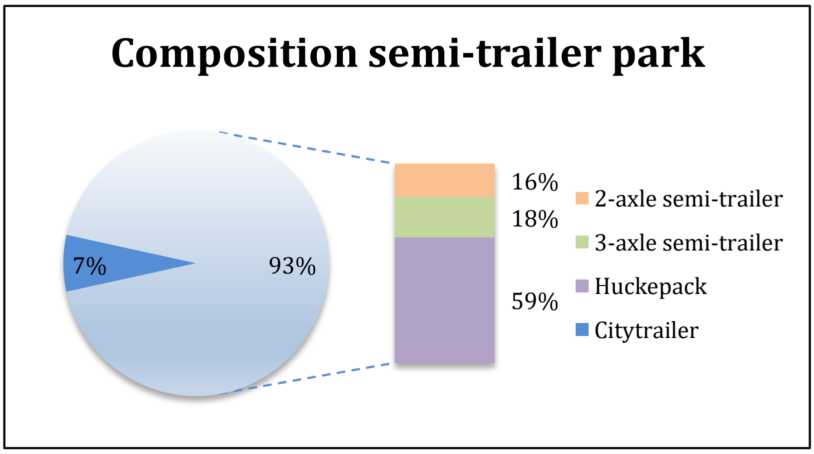


FIGURE 4-1 DECOMPOSITION SEMI-TRAILER FLEET OF ROTRA

The city trailers, which are 1-axle small-size semi-trailers (around 11 meters long), are only a small proportion of the entire fleet (7%). The remainder (93%) are standard-size 13,62 x 2,55 x 4,00 (l x w x h) semi-trailers and are mainly non-steered 3-axle semi-trailers (75% of the total fleet). The huckepack semi-trailers are also non-steered 3-axle semi-trailers, but can be lifted vertically and can be used for multimodal transport as they can be put on trains (see Figure 4-2). It can be safely assumed that the vast majority of semi-trailers of any medium- to large size LSP are non-steered 3-axle semi-trailers, especially transporting cargo throughout Europe as increasing the number of axles increases the maximum load which is more cost-effective for long-hauls. We therefore focus on non-steered 3-axle standard-size semi-trailers.

FIGURE 4-2 HUCKEPACK TRAILER LOADED ONTO A TRAIN

4.3 Current practices

Currently Rotra operates +/- 150 cargo movements a day, six days a week at their cross-dock in Doesburg. Most of the national cargo arrives at the beginning of the evening and leaves early in the morning (before 6 AM). International cargo usually leaves at the end of the evening using fixed time schedules. Docking and undocking the trailers is either done by the truck driver or using a Terberg yard tractors. The latter option is the focus of this research and these yard tractors are the basis of the AGVs to be developed. Figure 4-3 shows the steps required to handle a semi-trailer in the case of a pick-up and drop-off when employing a yard tractor.

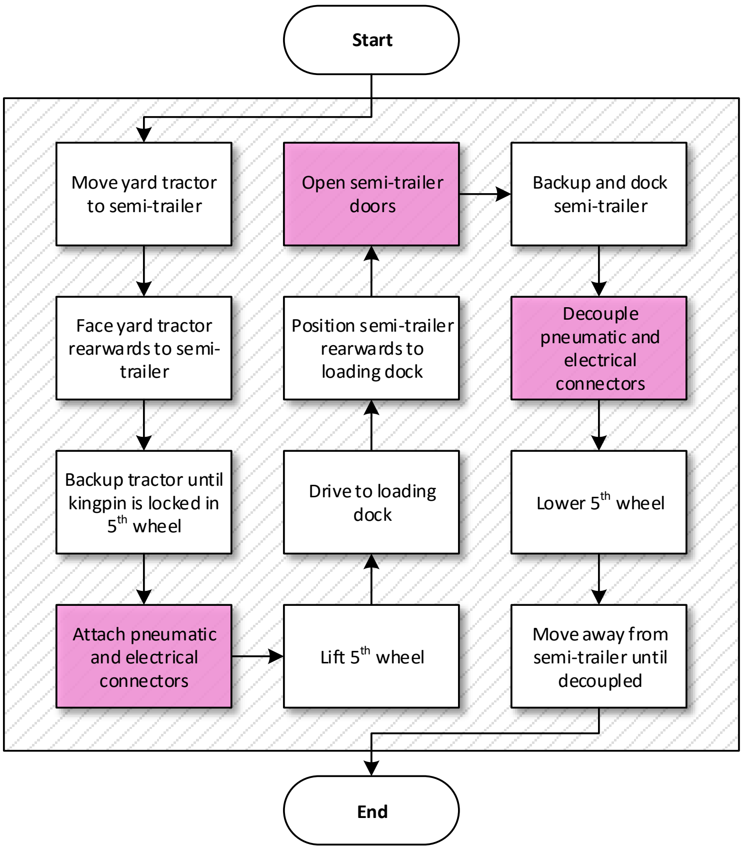


FIGURE 4-3 FLOWCHART YARD TRACTOR OPERATIONS

These steps are applicable when semi-trailers are (un)loaded from the rear-end. Based on data retrieved from Rotra this is currently around 94% of the cases, where the remainder is loaded from the side of the trailer. We therefore focus on rear (un)loading only. The steps shown in pink are manual operations and pose a challenge to the automation process. To illustrate, attaching the required lines to control the brakes of the semi-trailer is done manually and is not easy to automate as one has to work with industry standards and mechanical connections. From an automation point-of-view this is not desirable as this brings uncertainty, but this is also not desirable from a safety point-of-view. It is also hard to convince a party to buy an expensive AGV when they still require someone on the AGV to do the manual steps. This is further discussed in Section 4.5.1.

Pick-up or drop-off jobs are released to the yard tractor using a wireless connection and all information is shown on an on-board touchscreen. This includes the pick-up and drop-off location, which trailer to pick up and (if applicable) priority ruling. When a job is released by the planning department, the job is colored red. When a driver starts handling the job, he taps on the job and it turns yellow. After completing the job, the driver again taps on the screen and the job turns green. Currently Rotra operates two yard tractors simultaneously during peak hours (night and evening) and one during the day.

4.4 Pilot location forecasts

At the INTRALOG pilot location in Velp all movements should be automated. Rotra forecasts 400 vehicle movements a day in Velp. The question remains how the heterogeneous fleet of Rotra can be automated. In our definition of cargo we assumed that cargo can be decoupled from the transporting entity. This gives bounds to which vehicles can be automated. As this thesis focuses on AGVs transporting cargo, only semi-trailers are marked as cargo for the pilot study, which is 79% of the current fleet of Rotra (2- and 3-axle semi-trailers and Huckepacks). The number of axles has consequences for the vehicle controller, but as this is static information the system can easily share this information with the vehicle controller to correctly maneuver the trailer. Incorporating container movements of the container terminal are to be researched in further work. Another important design question is how manual and automated driving can co-exist at the cross-dock. For small loads or high-priority cargo it may not be beneficial to let the AGV do all the work, but let the truck driver maneuver and dock the semi-trailer himself. Whether or not this is beneficial depends on the (un)loading time, waiting time, driving regulations (maximum driving time) and the planning of the driver. Some threshold has to be designed to decide whether or not cargo is marked for AGV handling or manual handling. Also box trucks are handled manually and thus pose safety challenges. We propose a clear physical boundary between the AGV maneuvering space and all other manual operations at the cross-dock.

4.5 System design

This paragraph describes the design of the physical entities required within the system. We first denote some issues arising from practice in Section 4.5.1. The vehicles which serve as AGVs are described in Section 4.5.2. In Section 4.5.3 we briefly introduce the need for a proper IT system.

4.5.1 Functional requirements

Currently there are still many manual operations required for the handling of the semi-trailers at Rotra. To make the transition from manual to fully autonomous, there are some challenges to tackle. From the analysis of the current practices at Rotra (see Section 4.3) we conclude that the following three functional requirements are absolutely necessary:

- **Automatic coupling between AGV and semi-trailers**
  Although the fifth wheel and the kingpin are international standards on trucks and semi-trailers, there are still some manual tasks required when coupling. For example, the electrical and pneumatic lines have to be coupled to the semi-trailer when the kingpin is locked in the fifth wheel. This requires a driver who leaves the cabin of the vehicle to

attach these lines. To develop a fully autonomous system this coupling needs to be done automatically. At the time of writing there exists no such system. The complexity of an automatic coupling device is the heterogeneous fleet of semi-trailers where the connectors are all positioned at different locations. Furthermore, the semi-trailers shall not be equipped with extra hardware from a costs perspective as raised by Rotra.

- **Dock design**
  An important part of the process of docking semi-trailers is the dock itself. It provides the dimensions and tolerances for the AGV to dock the semi-trailer. Also, currently the doors of the semi-trailer are opened by the truck driver. As we do not allow personnel in our automatic system, the doors of the semi-trailers need to be opened from the inside of the distribution center. There are already market-ready solutions for this, but we have to design (in cooperation with a dock door supplier) a dock door where not only semi-trailers but also containers can be docked automatically and the doors can be opened from the inside. Another potential aspect of the dock door in the automation process is the communication between the dock and the AGV to let the AGV know when the semi-trailer is properly docked using some kind of sensor technology.

- **Wireless charging**
  We deploy electrical AGVs and thus they have to be charged at some point in time. When the AGVs are charging there are not available to the system, so the charging strategy can have serious impact on the performance of the system. Furthermore, as we do not allow personnel in the system, the charging also has to be done autonomously and most likely also wirelessly.

All these functional requirements have to be researched in order to build the business case for an autonomous system. One can imagine that LSPs do not make large investments into an automatic handling system, when there is still personnel required to (de)couple the semi-trailers or to plug in a charging cable. At the time of writing the above topics are recognized by the consortium partners of INTRALOG and are in an early stage of development by several bachelor students of the UT and HAN.

4.5.2 Vehicles

Automated Guided Vehicles (AGVs) transport the cargo (i.e., semi-trailers) on the terminal between loading docks and parking areas. These AGVs are based on the yard tractors (YTs) made by Terberg Benschop. Currently there are no automated YTs available, but within INTRALOG some parties research the transition from manual to automatic YTs. The prototype has yet to be built, but most likely an electric driven AGV will be used for pilot testing. Figure 4-4 shows a manual YT which will be the basis for the AGV.

FIGURE 4-4 SIDE-VIEW OF A YARD TRACTOR

Based on data retrieved from meetings with Terberg Benschop, the following specifications can be used as input for the simulation study (see Chapter 6).

General specifications

- Maximum forward speed (loaded/unloaded): 10 m/s
- Maximum rearward speed (unloaded): 7 m/s
- Maximum rearward speed (loaded): 2 m/s*
- Acceleration (loaded): 1,8 m/s^2
- Number of wheels: 6 (2 front, 2x2 back)
- Number of steered axles: 1
- Maximum capacity: 1 semi-trailer
- Maximum lifting capacity: 36 metric tons

Electric AGV specifications

- Average battery life: 4 hours (113 kWh) or 6 hours (169 kWh)
- Average consumption container handling: 22 kWh
- Average consumption logistic operations: 15 kWh
- Recharging time: Battery life divided by 2 (2 or 3 hours)
- Battery leveling required: after 140 operating hours
- Battery leveling time: 6 hours
- Acceleration (unloaded): 2,5 m/s^2

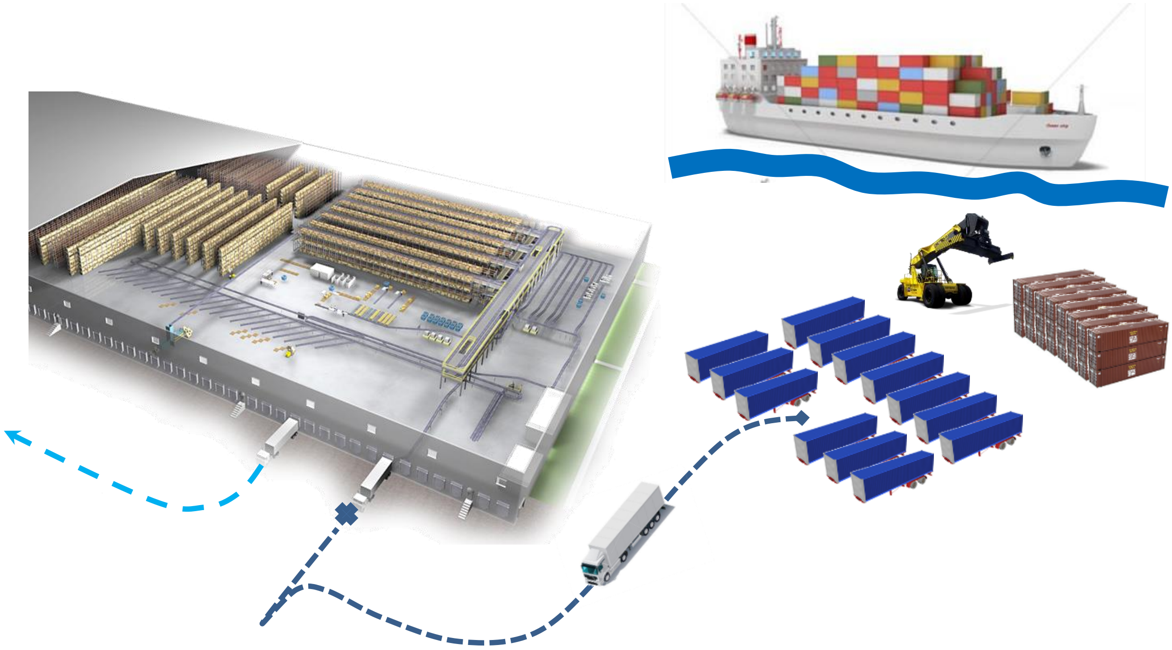
* although the maximum loaded rearward speed is theoretically 7 m/s, from a safety and stability point-of-view this speed is capped at 2 m/s.

4.5.3 Operating and information system

When the MAS has been fully designed and the work of the other work packages within INTRALOG is mature enough, we can start with a pilot test. Although this thesis does not focus on IT infrastructure, we would like to point out that the transition from conceptual/simulation to real-world pilot testing may be challenging. In order for the entire system to function properly many different hardware platforms and pieces of software have to be integrated such that they can communicate with each other. For example, the MAS requires information from higher level planning software (e.g. a TMS) and also sends information the controller of the AGV. This vehicle controller on his turn needs hardware and software for vehicle positioning and orientation. Although we recognize the importance of the IT system, we leave this to further research.

4.6 Summary

As stated in this chapter, the pilot location will feature a multimodal cross-dock with 150 loading docks, a container terminal and approximately 100 parking slots for trucks and semi-trailers. A concept drawing of a multimodal cross-dock is shown in Figure 4-5.



To summarize the most important properties of the pilot location and assumptions:

- 15.000 m^2 cross-dock (300x50x10 m);
- 150 loading docks;
- 100 parking slots;
- The cargo consists only of standard-size semi-trailers (2- and 3-axle);
- Integration of container movements of the container terminal at the pilot location is subject to further research;
- All cargo is (un)loaded from the rear;
- Forecasted vehicle movements: 400/day;
- A clear physical boundary between AGVs and manual operations is necessary;
- No humans should be involved in the handling process;
- The AGVs are based on yard tractors made by Terberg Benschop;
- We assume that the doors of the semi-trailers can be opened from the inside of the cross-dock;
- We assume that the yard tractors can autonomously (de)couple semi-trailers and are charged wirelessly and autonomously.

5 AGV FRAMEWORK

This chapter provides an AGV framework which we link to the MAS to establish the decision capabilities of all agents. Section 1 introduces the framework and Sections 2 till 9 each describe a dedicated part of the framework. Section 2.10 contains the conclusion of this chapter.

5.1 Introduction

The AGV system is an advanced Material Handling System (MHS) used to transport cargo from pick-up (P) to drop-off (D) locations. Designing this system involves multiple important and interconnected steps on strategic, tactical and operational levels. Decisions on these design issues have to be made such that the agents defined within the MAS know how the environment is shaped and what kind of intelligence an agent should inhabit to properly fulfill its function. The detailed design phase of the Prometheus Method has given a blueprint for all agents in the MAS on what to do given their goals and functionalities. In this chapter the question is answered *how exactly* the functionalities and capabilities defined previously are executed in terms of algorithms, heuristics and priority rules. To guide the design of the AGV system, we use the framework presented by Le-Anh & De Koster (2006), who conducted an extensive literature review on the design and control of these systems. Other research points out similar design issues e.g. (Vis, 2006), but lacks a comprehensive framework to guide decision making. Other researches have focused on parts of the system, such as vehicle scheduling problems (Qiu, Hsu, & Huang, 2002; Singh, Sarngadharan, & Pal, 2011), lay-out issues (Wallace, 2007), trajectory planning (Xin, Negenborn, & Lodewijks, 2014) or conflict-free routing (Breton, Maza, & Castagna, 2006; Oboth, Batta, & Karwan, 1999; Qiu & Hsu, 2001). Although these papers are of great value regarding their specific topic, they are at most partially valuable to the design of the system as a whole and most importantly: all sub-problems are interconnected as decisions on one part of the system influences or gives bounds to other parts of the system. The comprehensive framework presented in Figure 5-1 recognizes these interdependencies, includes multiple hierarchical levels and adequately addresses the entire design process.

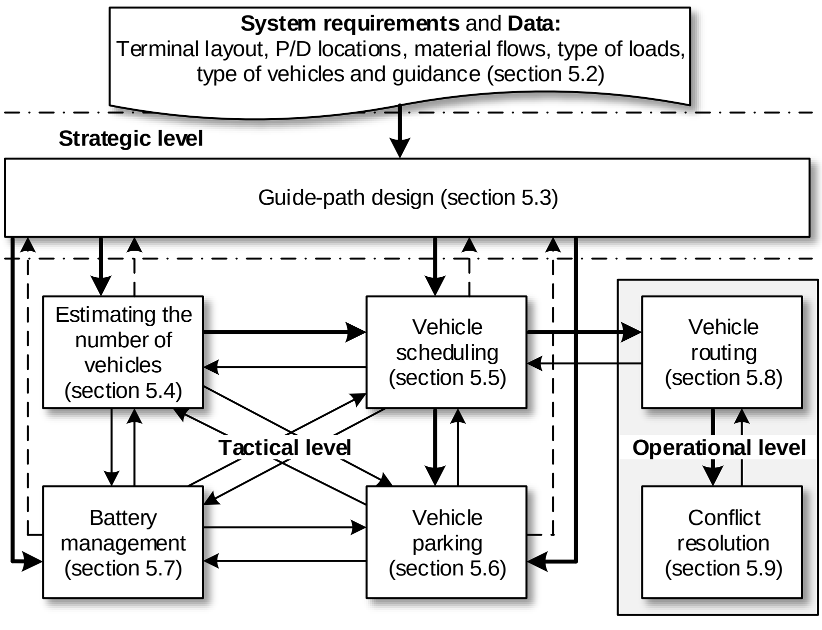


FIGURE 5-1 AGV FRAMEWORK (ADAPTED FROM LE-ANH & DE KOSTER (2006)

The design process starts with input data such as the terminal layout, P/D locations, material flow characteristics, the type of AGVs used and performance requirements. In our case, this input is obtained from industry partners (Rotra and Terberg) and educational partners (HAN and UT). The first strategic decision is designing the guide-path system, which provides bounds to lower hierarchical levels. After this, decisions on the tactical level are considered such as estimating the number of vehicles required, determining which vehicle scheduling algorithm to use, deciding upon which vehicle parking policy and battery management policy to use. These decisions influence each other and have different kinds of impact one and another indicated by thick (strong impact), thin (less strong impact) and dashed (none or less strong impact) arrows in Figure 5-1. Due to this interdependence, decisions should be made simultaneously. On the operational level, decisions on conflict-free routing are made. All steps in the design process are tackled in their respective paragraphs, shown in Figure 5-1.

5.2 Requirements and data

First, the systems requirements have to be defined and this includes all elements that characterize the cross-dock. We use the framework of (Ladier & Alpan, 2015) to assess all these elements, which are:

Shape
The shape of the cross-dock is one of the lay-out inputs to the AGV framework. We assume a rectangular shape of the cross-dock where only one side is used for automated transport (the south end) and the other for manual operations.

Number of doors
It is given that south side of the cross-dock will feature 75 loading docks.

Internal transportation
Internal transportation can be done manually or automatically. As this falls beyond the scope of our research, we do not include it in the requirements. We thus assume that there are enough resources available to not restrict the loading and unloading operations.

Service mode
The service mode of the doors can either be exclusive or mixed. The exclusive mode has separate doors for inbound and outbound cargo. Using the mixed mode, any door can be used for either loading or unloaded. We focus on the latter case.

Preemption
Preemption is allowed when the jobs assigned to the AGVs can be interrupted. Due to the nature of our problem we do not allow preemption and thus when an AGV has picked up any cargo, it should fully complete the job.

Temporary storage capacity
For the loading of departing trucks, some storage capacity is necessary at the cross-dock when the departing trailer is not yet available. This capacity can be either limited or unlimited. We assume unlimited storage capacity.

Internal resource capacity
Similarly to the storage capacity, we also assume that the internal resource capacity is sufficient and does not pose any bounds on the external operations.

Arrival times
The arrival time of cargo is first estimated when the cargo enters the proximity of the terminal. This arrival time will be updated and fixed when the cargo enters the terminal. For simulation

purposes we will use a certain arrival distribution and a certain noise between the expected arrival time and the actual arrival time.

Departure times
The departure times of outbound trucks are fixed and known upfront in compliance with current practices at Rotra.

Consolidation
Usually at a cross-dock goods from multiple arriving transports are combined to one or more departing transports. In reality it is known upfront which goods have to be loaded into a departing truck and from which arriving transport these goods originate from. Loading the departing truck can start when all (or part) of the goods have been unloaded from their transports. For simulation purposes we will use a certain distribution to decide how many inbound transports are required for a single outbound transport and the inbound transports are randomly assigned to outbound transports. We further assume that all arriving transports arrive on time and can be fully processed before the earliest due date of the departing transports requiring cargo from a particular arriving transport.

Product interchangeability
Goods are interchangeable when two products of the same type can replace each other in a truck. We strictly assume that products are not interchangeable as customers of Rotra place orders for the transport of their goods from A to B. These goods can thus not be interchanged with the products of another customer.

5.3 Guide-path design

One of the key design issues is the design of the guide-path (or flow path). This question should be answered at the strategic level as designing the lay-out of these paths has a direct impact on system performance (such as travel time, number of required vehicles and degree of congestion) as it provides bounds to where AGVs are able to drive. It is also a medium to long-term decision as changing the layout can be very costly, especially when wire guidance is used. The guide-path is a layout of trajectories that AGVs can follow by using, for example, wires in the ground connecting all P/D locations (i.e., loading docks and parking slots). Designing the lay-out can be done in various ways. First, the layout of the building, the location of the P/D points and the guide-path layout can be determined simultaneously. Second, the guide-path and the location of the P/D points can be determined, using the layout of the building as input. Third, the guide-path can be designed, using the P/D points and the lay-out of the building as input factors (Vis, 2006). We focus on the third option as the lay-out of the building and the locations of the P/D point are relatively straight-forward. The cross-dock will have a rectangular shape with 75 loading docks (P/D locations) at both sides. In this research we focus on one side of the cross-dock only, because the other side is used for manual operations. This focus is made to create a clear physical boundary between manual and automatic operations to ensure safe working conditions and to separate non-automated transport (e.g. rigid trucks) from automated transport. The key design issue is to create a well-balanced trade-off between the space required for the lay-out and the most efficient way of maneuvering, such that it can accommodate a multitude of vehicle configurations.

The input required is thus the lay-out of the building and all P/D locations. Based on concept drawings provided by Rotra for the pilot location, the lay-out shown in Figure 5-2 will provide a base for the guide-path design.

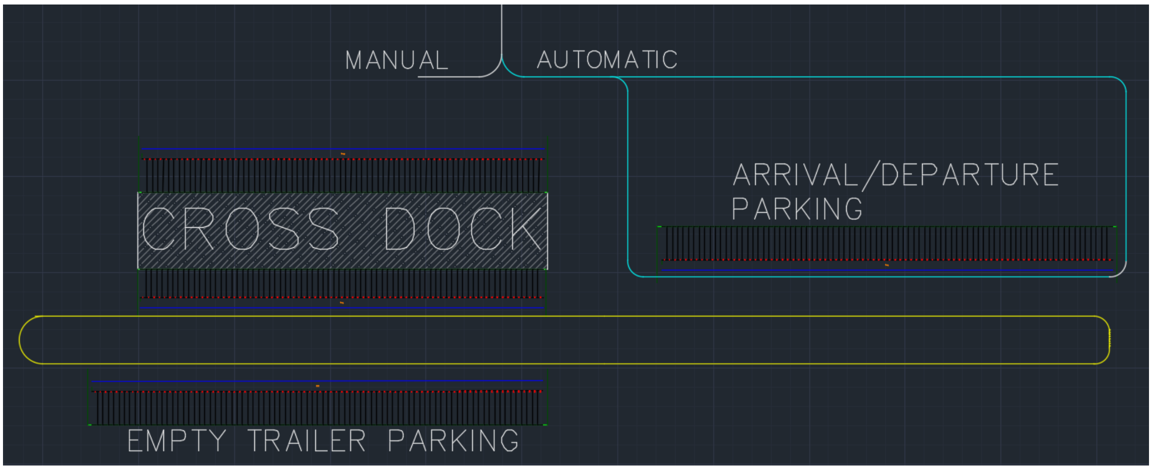


FIGURE 5-2 LAY-OUT OF PILOT LOCATION

The entrance of the terminal is at the top of Figure 5-2, where incoming cargo which will be handled manually, is separated from cargo which will be handled automatically (using AGVs). The blue line shows the trajectory truck drivers will follow on the terrain, after being assigned a parking slot at the arrival/departure (A/D) parking. The yellow line shows the main single loop for AGV guidance. It connects the main areas (A/D parking, cross-dock and empty trailing parking). The empty trailer parking is separated from the A/D parking (which only parks loaded cargo) to decrease the unloaded travel time on the terminal. When an empty trailer is needed at the cross-dock for a departing transport, it can be picked up at the parking opposite to the cross-dock, which reduces travel time. This concept is currently also being used by Rotra. The question remains how to connect the blue and yellow line (boundary between the truck driver and the AGV) and to bring further detail to the main loop to facilitate docking and parking maneuvers.

The layout is usually represented by a directed network of arcs, where P/D locations can be considered as nodes. These P/D locations consist of loading docks as well as parking slots. The direction of the arcs can be *unidirectional* or *bidirectional*. In the first case, vehicles are only allowed to travel in one direction and in the latter case vehicles are allowed to travel in both directions. Using bidirectional arcs, the travel distance can be decreased as AGVs can take short cuts, but on the other hand unidirectional flows are easier to control as no opposite traffic is allowed (Vis, 2006). It is also possible to create multiple lane guide-paths, which allows for parallel flows, but increases the space required. Another possibility is to use a *mixed uni-bidirectional* guide-path where a mixture of unidirectional and bidirectional arcs is being used. There are also different guide-path systems, *single loop, tandem* and *segmented* (Le-Anh, 2005). The single loop system consists of a, usually unidirectional, single-loop where AGVs can take no alternative routes. The tandem configuration has multiple zones and every zone has one vehicle. Loads are transferred using transfer-stations. The segmented guide-path system has one or more zones, separated into non-overlapping segments and are served by a single vehicle. Transfer buffers at the end of each segment function as an interface between the segments (Le-Anh & De Koster, 2006). For this MHS is it obvious that the latter two systems will not work because cargo then needs to be decoupled from the AGV and re-coupled to another AGV during maneuvering. From a practical point-of-view this is not desirable because (de)coupling takes a considerable amount of time compared to the travel time and requires multiple step. A single loop thus seems a straightforward and easy system to use. The single loop however only functions to connect the main areas on the terminal and does not facilitate docking maneuvers. We therefore denote this single loop as the *main road*, which has a top and a bottom section and is unidirectional, which is easy to control, but can cause congestion. We extent the single loop with *crossroads* per P/D location to (i) clear the main road to lower congestion levels and (ii) form a guide-path for the rearward docking/parking maneuver.

5.3.1 Crossroad design

The crossroads facilitate rearward docking and parking maneuvers and are thus by definition bidirectional arcs. This makes the entire system a mixed uni-bidirectional guide-path. There are multiple ways of designing the crossroads of which the three most interesting are further discussed here. Figure 5-3 shows three different crossroad designs. The cross dock is depicted at the top of the figure and the parking area at the bottom. Each loading dock and parking slot will have a crossroad, but only one crossroad is shown to illustrate the differences in design. The main roads are shown as the east-west arcs in each design and are connected with an arc at both ends of the road, consistent with Figure 5-2. The east-west direction at the top has been chosen (and thus west-east at the bottom) to decrease the travel distance from the A/D parking to the cross-dock. This means that the travel distance from the empty trailer parking to the cross-dock increases, but due to the importance of loaded cargo compared to empty cargo, the east-west direction has been chosen for the top main road.

A docking maneuver starts with a loaded AGV at the top main road on the right-hand side of the dock. The AGV turns away from the main road facing south until the back of the cargo is facing the dock. The AGV then backs up the cargo until it is docked into the (un)loading area. When the AGV is decoupled it can continue its path using the curve facing away from the cross-dock in western direction. When picking up a docked trailer the same maneuver is used, except with an unloaded AGV. The parking maneuver is identical to the docking maneuver, only the AGV starts at the bottom main road on the left-hand side of the parking slot. All north-south arcs are thus bidirectional and require a certain amount of space to straighten out the trailer behind the AGV. This also depends on the number of axles and the length of the trailer. To avoid a discussion on kinematic properties of articulated vehicles, we use a line break to resemble this space as shown in Figure 5-3. The remainder of this paragraph discusses the different designs shown below.

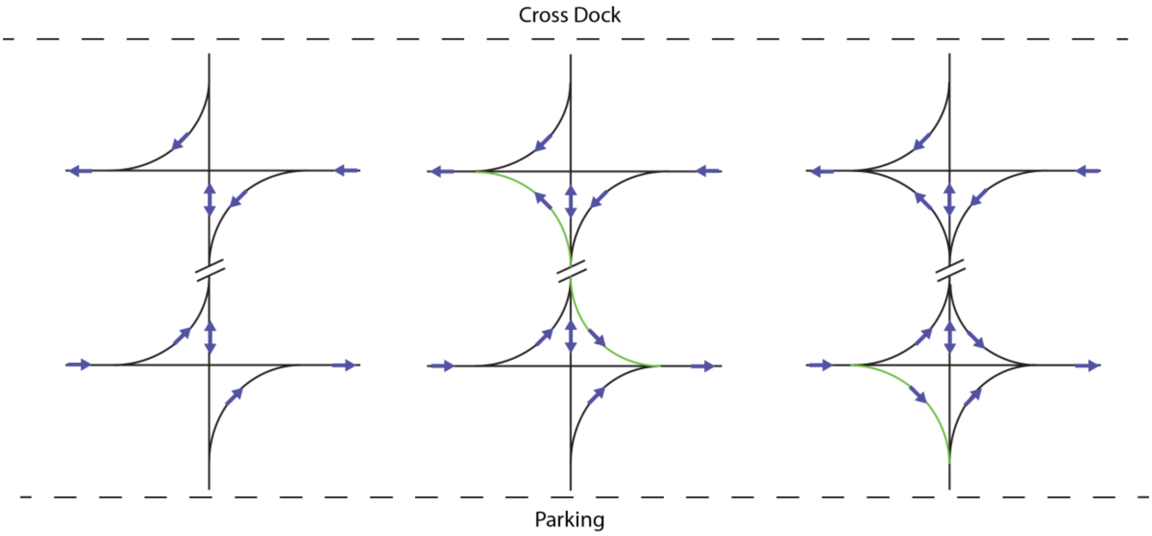


FIGURE 5-3 POSSIBLE GUIDE-PATH DESIGNS. LEFT: DESIGN 1; MIDDLE: DESIGN 2; RIGHT: DESIGN 3.

Design 1

The first design is the most straightforward one; it has a unidirectional main road to connect the main areas and bi-directional north-south arcs to facilitate docking and parking maneuvers, similarly to the other designs, see on the left of Figure 5-3. The only possible head-to-head conflict area is the north-south arc between the main roads in the case an AGV docks cargo at a certain dock and a different AGV wants to park a trailer using the same north-south arc. The extent to which this is a problem depends on the traffic intensity at the terminal, the higher the intensity the more likely this will occur. It could be argued to increase the area between the main roads to accommodate the length of two loaded AGVs, but this would simply require too much space. A better solution is to check whether or not the arc is empty before turning into a vertical arc. Depending on the complexity of the system this can be done when an AGV arrives at a vertical aisle (passes a sensor) or by using time-based information on the occupancy of arcs in the network derived from the AGV schedules. The latter option is far more complicated.

The major disadvantage of this design arises when an AGV has to travel between the two main roads. As there are no shortcuts or alternative routes the AGV always has to travel (a part of) the main road loop. To illustrate this, suppose an empty trailer has to be picked up at the cross-dock and parked at the Empty Trailer parking. Furthermore, suppose the pick-up and drop-off locations are determined randomly (uniform distribution). On average the AGV then has to travel from the center of the cross-dock to the center of the parking area. Let L be the length of the cross-dock, equal to the length of the parking area and let x be the total horizontal travel distance of the AGV when moving from point a to point b. In this scenario point a and b are both located at $\frac{1}{2}L$ as shown in Figure 5-4*.* Ignoring the horizontal distance needed to make the 90 degree turn to travel between the top and bottom road, the total horizontal travel distance x is then equal to $2 * \frac{1}{2}L = L$. This shows that on average the entire length of the cross-dock has to be traversed to pick-up or drop-off an empty trailer. For our pilot study this resembles to around 300 meters.

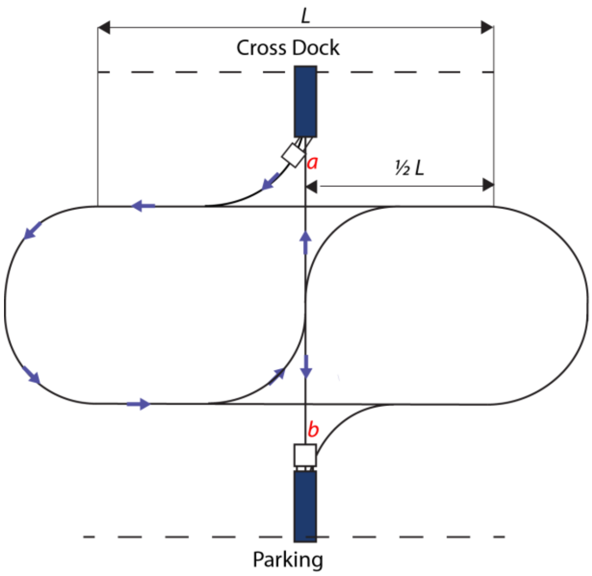


FIGURE 5-4 ILLUSTRATION OF GUIDE-PATH DESIGN 1

Design 2

The crossroad shown in the middle of Figure 5-3 tries to solve the above mentioned problem by adding two additional arcs (shown in green). These arcs facilitate moving between the two main roads at every vertical arc. The benefit is that the main roads are still unidirectional, but the travel distance can be decreased dramatically. The downside is the increase in possible head-to-head collisions from two to four, compared to Design 1. These two are added due to either an AGV moving from the top road to the bottom road or vice versa. More checks thus have to be made before entering a vertical arc. Consider the same example as in Design 1. In this new design an AGV can take a shortcut using a vertical arc. Which arc to use depends on the vehicle kinematics (e.g. its turning radius). The AGV thus first has to move west on the top main road, enter a vertical arc, leave the vertical arc heading eastwards on the bottom main road and then enter the vertical arc belonging to the parking destination. The reduction in travel distance compared to Design 1 depends on the horizontal distance required to make the turn from the bottom to the top road. Suppose the AGV has to pass d docks westwards before being able to make the turn. Note that d depends the dimensions and kinematics of the cargo. This corresponds to a fraction of the length of the cross dock of $\alpha = \frac{d}{n}$, where n is the total number of docks. These n docks are equally spaced throughout the entire length of the cross-dock. The total horizontal distance x is thus $2\alpha L$, where $\alpha \leq 1$ as illustrated in Figure 5-5.

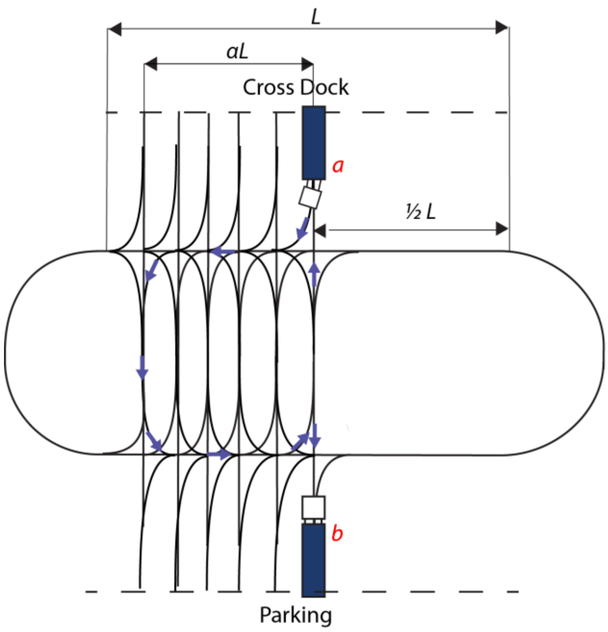


FIGURE 5-5 ILLUSTRATION OF GUIDE-PATH DESIGN 2

Figure 5-6 shows the relation between d and x, as well as the relative reduction in travel distance compared to Design 1 in the case of $n = 75$ and $L = 300$.

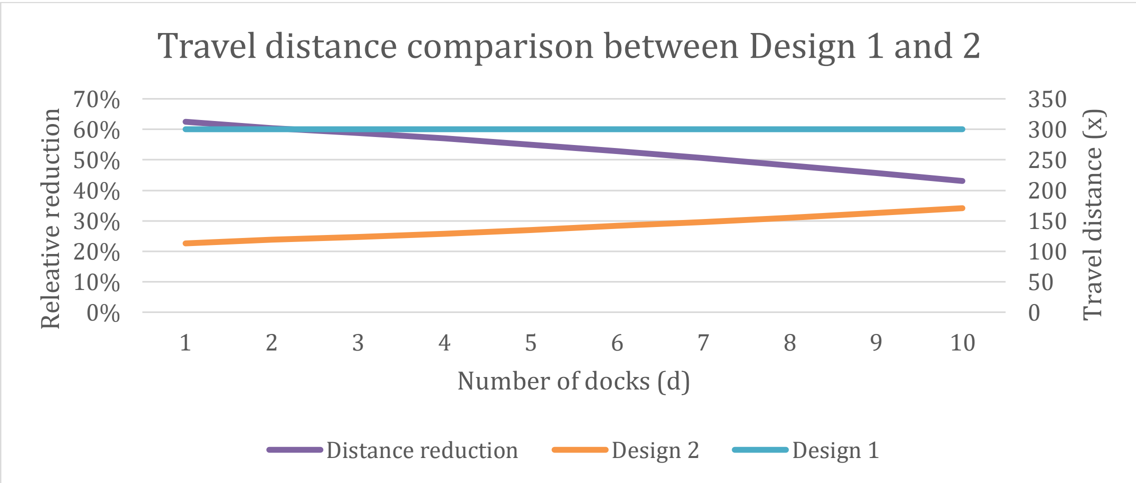


FIGURE 5-6 TRAVEL DISTANCE COMPARISON BETWEEN DESIGN 1 AND 2

Using an AGV loaded with a 3-axle non-steered semi-trailer as an example, the value of d would approximately be 4 to 5 docks based on the minimal turning radius plus a safety margin. This results in a 55-57% reduction in travel distance. Even when using LHVs, reductions around 50% are to be expected. Whether or not LHVs should be maneuvered in this way is of course subject to discussion. It must however be noted that, due to trailer swinging, not only the vertical arc the AGV want to travel on should be empty, but also some of the adjacent arcs to the left of this arc as the semi-trailer will cross these. The concept of trailer swinging is illustrated in Figure 5-7 using a LHV.

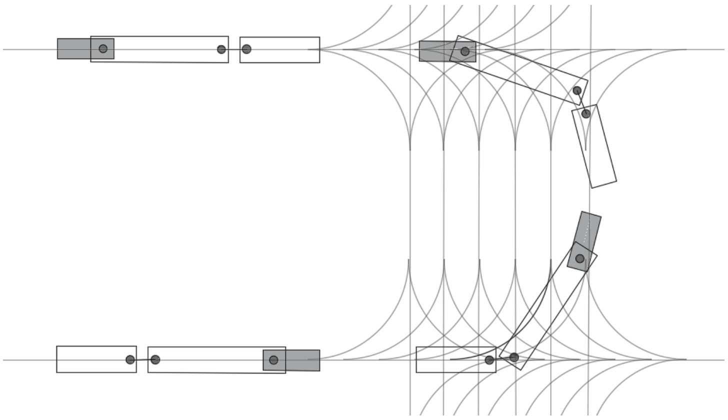
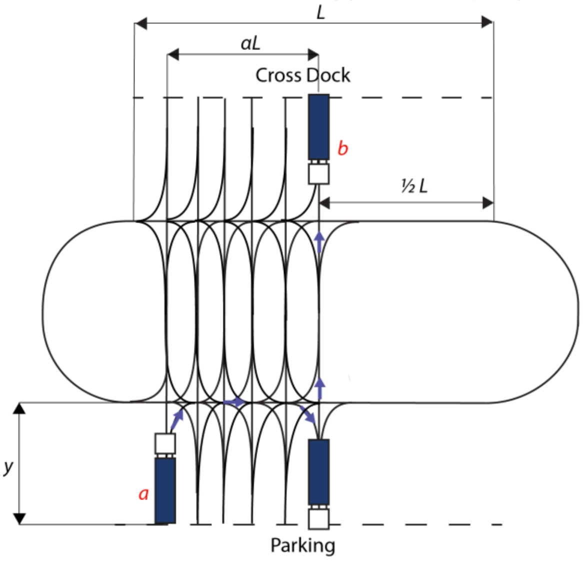


FIGURE 5-7 ILLUSTRATING OF THE SWING OF A LHV

This swing lowers the probability of using a short-cut especially when the traffic intensity is high, due to the blocking of adjacent vertical arcs. When this probability is close to zero, Design 2 is effectively equal to Design 1. When the arc required to make the short-cut is not accessible, vertical arcs more to the east can also be used or the AGV has to wait until the vertical arc is available. The first option increases the travel distance and thus decreases the benefit compared to Design 1. In any case, Design 2 is at least as good as Design 1. In practice the traffic intensity will be bounded by the number of AGVs used. In the Rotra case the number of AGVs will be small compared to the number of docks and we expect a low probability of possible conflicts.

Design 3

The final design adds one extra arc compared to Design 2 (shown in green in Figure 5-3) to enable the use of the parking area as maneuvering space. When an empty trailer should be docked eastwards of the parking slot where it is picked up, the AGV can use the parking slot opposite to the loading dock as maneuvering space (see Figure 5-8). The AGV can then enter the vertical arc from the bottom main road instead of the top road. This increases the complexity as one extra check has to be added: whether or not the parking slot is empty. Also adjacent parking slots should be empty due to the swing of the trailer. Another possible disadvantage is the increase of rearward driving. The total horizontal travel distance x in this design is αL, as the only horizontal distance is directed eastwards. Different from the other designs, this design *does* influence the travel distance in the vertical direction. This extra distance is approximately equal to the depth of the parking slot. The total travel distance (in both directions) is thus $\alpha L + y$, where y is the depth of the parking slot.



Basic algebra shows that Design 3 has a shorter total travel distance when $y < \alpha L$. Using $d = 5$, this would result in $y < 20$. As an AGV loaded with a semi-trailer already is around 18 meters long, the added benefit of Design 3 compared to Design 2 is limited. The AGV also needs more space than its entire length to straighten out the semi-trailer behind the AGV such that it can continue its path rearwards. Suppose that $y < 20$, the question still remains what the probability of an empty parking slot would be. This depends on the number of trailers present at the terminal, which may fluctuate during the day. It is difficult to provide a good estimate of this probability as this is very case-specific. Table 5-1 summarizes the above findings.

TABLE 5-1 COMPARISON GUIDE-PATH DESIGNS

Design	**# of 90° arcs**	**# of possible conflicts**	**Expected average travel distance**
1	4	2	L
2	6	4	$2\alpha L$
3	7	5	$\alpha L + y$

This section has evaluated three different cross-road designs. We have shown that all designs have advantages and disadvantages. We conclude by choosing Design 2 to be used in our guide-path, due to the large reduction in travel distance compared to Design 1 (+/- 55%) which outweighs the need for extra control. Design 3 has not been chosen as the benefit compared to Design 2 is probably non-existent due to the increase in the y-direction. Even if it is present, it would be marginal and the probability of it occurring is expected to be small. Note that Design 2 can also be used to connect the main road with the A/D parking. In this case the design is somewhat simpler as loaded AGVs moving to the cross-dock only need to travel westwards and there is no parking on the south end. The two lower east-directed 90° arcs can thus be removed.

Figure 5-9 shows our proposed layout of the terminal including the guide-path design. Not all crossroads are shown to simplify the drawing. Also an AGV parking area is included and it is strategically positioned between the three main areas (see Section 5.6). The blue arcs represent the guide-paths for the truck drivers and the yellow arcs represent the AGV guide-paths. Extra space is included to the south of the A/D parking as a safety zone between trucks and AGVs. An additional green arc has been included as a maintenance path.

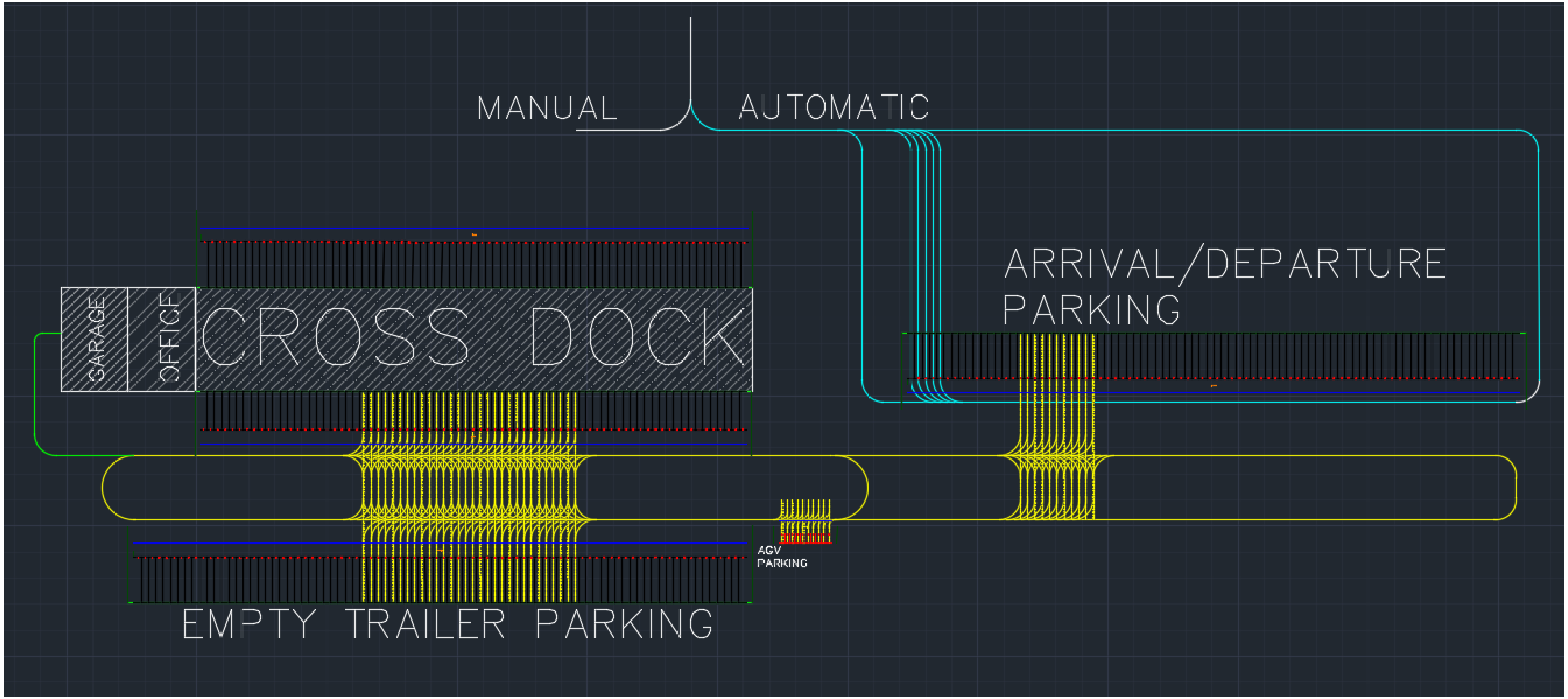


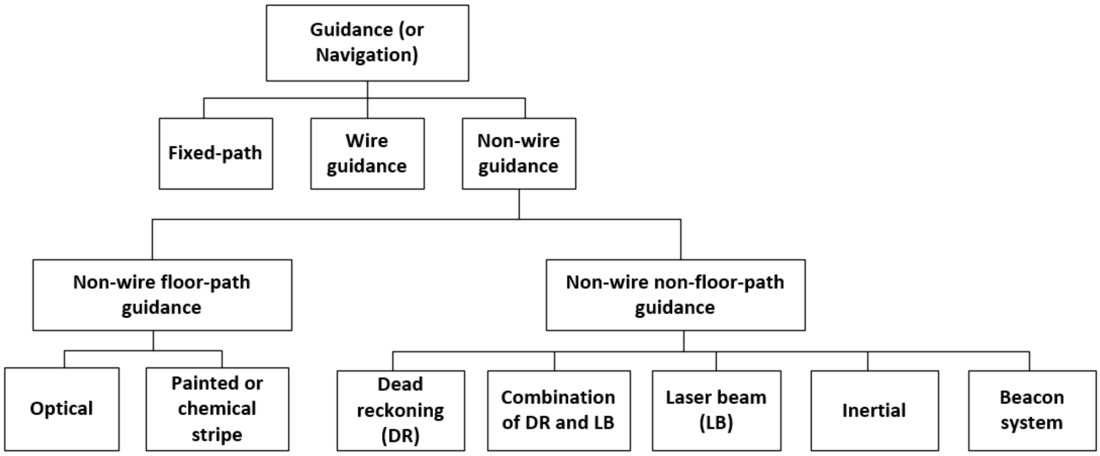
FIGURE 5-9 AUTOCAD DRAWING OF THE CROSS-DOCK AREA PARTIALLY FILLED WITH GUIDE-PATHS

5.3.2 Vehicle guidance

The guide-path design presented above is a simplified view off the trajectories followed by AGVs in real-life. The 90 degree arcs shown in Figure 5-3 are not accurately representing the curvature of an AGV (with or without semi-trailer). The actual trajectory and positioning is important information for the vehicle controller. Without going in too much detail, the trajectory followed by an AGV in real-life when turning is not a perfect circle but an eclipse. At some point on this eclipse there is a sweet spot for rearward docking as it minimizes steering effort. At this point the maneuvering is the easiest from a controller point-of-view and thus also decreases the maneuvering time and increases the probability to dock the trailer the first-time-right. This sweet spot depends on the vehicle kinematics and on the trailer properties (length, number of axles, non-steered or steered, number of articulation points, etc.). When following an eclipse shaped trajectory, the AGV thus makes a turn less sharp than 90 degrees. Within our scope it is sufficient to be aware of this phenomenon and to keep it in mind when designing the guide-path. That is, not to limit ourselves to a simplified view, such that at the implementation stage, the actual maneuvering is restricted. This means, for example, that we have to take Longer and Heavier Vehicles (LHVs) in to account, which, in The Netherlands, are road legal double-articulated vehicles with a maximum length of 25.25 meters (see Figure 5-10). These LHVs consist of two standard road transport vehicles and can be configured in multiple ways, see Kindt (2011) for more information. These combinations require more maneuvering space and more complex maneuvering than a standard semi-trailer and the lay-out should be prepared for these type of vehicles. The interested reader is referred to Michalek & Kielczewski (2013) and Kural, Prati, Besselink, Pauwelussen & Nijmeijer (2013) for more details on multiple-trailer vehicles.

FIGURE 5-10 A ROTRA LHV

For the remainder we use rough estimates on vehicle positioning and maneuvering times to assess the logistic performance of the system and let the exact vehicle localization and maneuvering to further research. However, we do address the possibilities of vehicle guidance to identify any restrictions of the guide-path design on real-world applications. This has been done using the work of Le-Anh (2005) on vehicle guidance, which features a classification of vehicle guidance as shown in Figure 5-11. This figure is slightly modified to only show the parts relevant for this research.



The three main types of vehicle guidance are: fixed-path, wire and non-wire guidance.

- The *fixed-path guidance* uses fixed tracks (e.g. rail) systems. This is a robust system, but is hard and costly to change and is thus not flexible.

- *Wire guidance* is an alternative where electrical lines are placed underground to guide the AGVs. Similar advantages and disadvantages arise with this type of system compared to the fixed-path system.

- *Non-wire guidance* is a system where virtual guide-paths are used. These are thus not restricted to any physical infrastructure and provide a very flexible system. According to Figure 5-11 this can be done using different systems.

From the discussion above on eclipse shaped paths depending on the vehicle kinematics we conclude that both fixed-path and wire guidance are not desirable. These systems restrict the maneuvering of AGVs, which limits the use of different trailer configurations. We argue that multiple wire systems could be implemented to accommodate different trailer types, but this seems a costly and tedious task. Also when using this kind of system, the trailers need to be equipped with sensors to locate the wires. This further restricts the possibilities as only the own fleet of Rotra can be equipped with these kind of sensors and charters or other external trailers cannot be used. We conclude by recommending a non-wire virtual guidance system, which is capable of guiding a heterogeneous fleet, including the fleet of external parties and is the most flexible and scalable way of vehicle localization and guidance. What the preferred type of guidance should be (e.g. laser guidance or painted stripes) we leave to further research.

5.4 Estimating the number of vehicles

The first important decision on the tactical level is estimating the number of vehicles required as this influences the performance of the AGV system to a large extent (Van der Meer, 2000). AGVs are usually also expensive, thus overestimating the number of vehicles is costly. Electric AGVs are even more expensive compare to models with a diesel engine, due to the nature of the AGV itself, but also additional costs such as charging stations and batteries. Determining the appropriate number and type of AGVs is thus an issue of major importance. This can be done either analytically or simulation-based. We employ both methods to first analytically estimate the number of vehicles as input for the simulation and then further re-optimize the number of vehicles based on certain performance criteria.
According to Egbelu (1987) the three main factors affecting the number of vehicles required are (1) guide-path layout, (2) locations of the P/D point and (3) vehicle dispatching strategies. We

focus on single-load capacity AGVs and one of the models presented by Egbelu (1987) provides a reasonably good estimation on the number of the vehicles, where his other models are normally over optimistic. The model uses a distance matrix between all P/D locations and the flow of cargo as input and estimates the number of vehicles as:

$$N = \left[\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{D_{ij}}{V}\right) + \left(\sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij}\right) * (t_u + t_l)\right] / (60T - t)$$

where n = number of P/D locations; f_{ij} = expected number of loaded trips between location i and j during period T; D_{ij} = estimated travel distance between location i and j; T = length of the period or shift during which f_{ij} occurs; V = average vehicle travel speed; t_l = mean time to load a vehicle; t_u = mean time to unload a vehicle; t = expected time loss due to battery charging.

In our case D_{ij} and f_{ij} should also reflect empty trailer movements from the empty trailer parking area to the cross-dock and vice versa. We estimate the corresponding distance using the guide-path design and the corresponding number of trips as the half of the total number of arrivals and departures (thus assuming a 50/50 distribution of arrivals and departures). For every departure, an empty trailer should be moved from the empty trailer parking area to the cross-dock. Plugging in the remainder of the parameters, provided by Rotra and Terberg, leads to the following equation:

$$N = \left[\left(\frac{360 + 180}{5}\right) + (400 + 200) * (120 + 120)\right] / (3600 * (24 - 7.5)) \approx 3\,.$$

These values are rough estimates and based on electrical AGVs and prognoses on the pilot location in Velp. It seems to provide a reasonable estimate on the number of vehicles required considering that Rotra currently deploys a maximum of two yard tractors. This estimate does not include any delays caused by congestion or other external factors. The influence of congestion has to be assessed using simulations, but as an initial estimate we use three AGVs.

5.5 Vehicle scheduling

We decide to use a decentralized system where agents only have a limited view of the system and are thus restricted to their own sets of data, beliefs and views. The key to solving complex problems within a MAS lies in the interactions between agents. One of the ways to accomplish this interaction is by using a certain auction mechanism where AGVs compete for orders (Mes et al., 2007). These mechanisms usually involve an Initiator who wants to sell a certain ‘product’ and a Participant who wants to buy the ‘product’ (Hafidz, Lin, & Murata, 2010). In this case the ‘product’ would be a transportation order and the participants are AGVs willing to fulfill the order. We use an auction mechanism within the Vehicle Scheduling agent to assign transportation jobs to the AGV Managers. This mechanism is based on the assumptions as defined in Section 5.2.

5.5.1 Auction mechanism

A well-known standard for the auction mechanism is the FIPA Contract Net Protocol (CNP). This standard has been developed by the Foundation for Intelligent Physical Agents (FIPA) to formalize the communication protocol between agents within a system (Hafidz et al., 2010). According the formal specification of the CNP, the agent uses the following sequence of steps to negotiate each contract (“FIPA Contract Net Interaction Protocol Specification,” 2002):

1. The initiator sends a Call for Proposals (CFP) to each of the participants, which specifies the task.
2. Each participant reviews the CFP and is able to make a response (or bid). Of these, some are proposals to perform the task and others may refuse to respond.

3. The initiator reviews all proposals and selects and informs a winner and informs the others of rejection.
4. The participant informs the initiator when it starts executing the task.

Once a task has been assigned to a participant, it must perform the task. This also restricts the CNP as multi-stage bidding is not allowed. Due to the stochastic nature of the system, it may very well be the case that after assigning the winner, better opportunities arise for the transportation job from other AGVs. This could be due to the large number of uncertainties within the system, such as stochastic arrival times, loading/unloading times or waiting times due to traffic congestion. Therefore we use an improved CNP with multi-stage proposals as defined by Hafidz et al., (2010) who have shown the added benefit of using multi-stage bidding. The improved CNP is shown in Figure 5-12. Within this protocol all *m* AGV agents are able to re-propose during the auction period τ. The length of this period start with the initial acceptance of the job and ends when the job executes. This is a logical auction period as changing the AGV allocation after coupling the cargo to the AGV is not desirable. At the end of period τ, the AGV is thus committed to the job. A re-proposal occurs when there is time left until the execution of the next job in the sequence of an AGV.

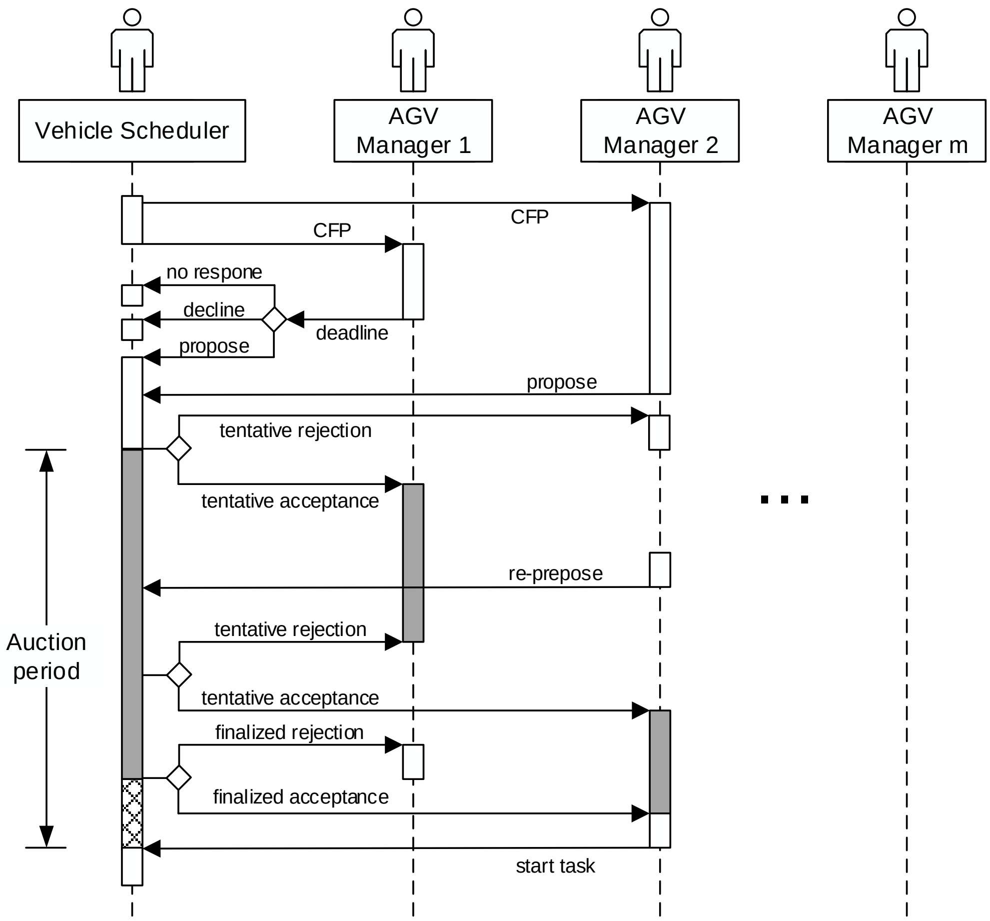


FIGURE 5-12 FIPA COMPLIANT MULTI-STAGE CONTRACT NET PROTOCOL

5.5.2 Transportation network and demand

The AGVs perform transportation jobs using the network of guide-paths as defined in Section 5.3. This network consists of a set of nodes and a set of arcs connecting these nodes. Every loading dock and every parking slot has a corresponding node. The transportation jobs arrive one-by-one and the expected arrival time is provided by the on-board computer of the cargo. Similarly to Mes et al., (2007) the orders are characterized by the following parameters: the origin node i, the destination node j, the earliest pick-up time at the origin r, the latest delivery time at the destination d (due time) and the time at which the order becomes known to the network $a \leq r$. The parameter a is first estimated when the cargo enters the proximity (e.g. within 20 km) of the terminal and updated when the cargo arrives at the terminal. We assume a random error between the expected and the actual arrival time. All parameters are provided by the Parking Manager or the Demand Manager. The orders are classified as (i) a loaded transport from the A/D parking to the cross-dock (cargo arrival); (ii) an empty transport from the parking area to the cross-dock (empty trailer movement) or (iii) a loaded transport from the cross-dock to the A/D parking area (cargo departure).

5.5.3 Schedules

All AGV Managers maintain their own schedule and consists of a sequence of actions of the following types: (i) move loaded along arc (*i,j*); (ii) move empty along arc (*i,j*) or (iii) wait at node j until time *t* (Mes et al., 2007). The sequences of arcs to travel are provided by the Vehicle Routing Agent based on the origin i and the destination j. Waiting at nodes also includes vehicle idling or charging at parking slots. Vehicle schedules are updated in the case of: (a) completion of a job (b) new job assigned to AGV (c) AGV needs charging (d) AGV needs preventive maintenance. The latter is based on a certain periodic cycle and will block the AGV schedule for a predetermined period of time. When AGVs do not operate 24/7, maintenance can be carried out when the system is down, provided that the maintenance cycle coincides with system down time.

5.5.4 Bid evaluation

The Vehicle Scheduling agent evaluates the bids. A decline may incur when an AGV is currently charging and has to reach a certain battery level threshold before continuing operations. This is determined and enforced by the Battery Manager agent. An AGV agent does not respond when it malfunctions (e.g. maintenance) to not let the Vehicle Scheduling agent wait an indefinite amount of time to receive a response. Similarly to Mes et al., (2007), we let S_m^0 denote the current schedule of AGV m and accepting a new job l will lead to n different alternative schedules, denoted by S_m^n where n is the number of jobs currently in the schedule. The number of currently scheduled jobs on AGV m is thus equal to n as the first job being excecuted is fixed. We use an insertion method, capable of inserting job l at n different positions in the current schedule. Note that idling or moving to the AGV parking is not a job. We do not allow shuffling of the current schedule as this would increase the complexity of the bid evaluation to a large extent. We do however allow multi-stage bidding. The bid value of AGV m is equal to the minimum additional costs over all alternative schedules n, including additional transportation time T, possible waiting time W and the total change of penalty costs for tardiness D (Mes et al., 2007):

$$\hat{P}_{m,l} = \min_{n}\left(c_v^t \Delta \mathrm{T}_{m,l,n} + c_v^w \Delta \mathrm{W}_{m,l,n} + \sum_{\forall o \in S_m^n} \left\{ c_o^d(D_{m,o,l,n}) - c_o^d(D_{m,o,l,0}) \right\} \right)$$

where $\Delta \mathrm{T}_{m,l.n}$ is the expected additional travel- and handling time for AGV m in schedule n to transport job l, $\Delta \mathrm{W}_{m,l.n}$ is the expected additional waiting time involved when AGV m transports job l in schedule alternative n and $D_{m,o,l,n}$ is the tardiness of job o after adding job l to schedule alternative n of AGV m. All variables are multiplied with their respective cost parameter. Tardiness occurs when a departing trailer is dropped off at the parking area after its due date (or departure time). The Vehicle Scheduling agent accepts the best bid provided by all AGV agents.

When all bids are equal the Vehicle Scheduling agent assigns the job to the AGV with the highest battery level. When this is also equal a random AGV is chosen. When during period τ an AGV v makes a better bid for job l than the currently assigned AGV m, where $v \in \{M \backslash m\}$, then the Vehicle Scheduling agent tentatively rejects $\hat{P}_{m,l}$ and tentatively accepts $\hat{P}_{v,l}$ until the end of period τ when all acceptances and rejections are made final.

5.6 Vehicle parking

Vehicle idleness is not uncommon in an automated Material Handling System (MHS) due to irregularities in demand and therefore the positioning strategy of an AGV when idling has to be determined. The two main strategies are static and dynamic. The main goal of each parking strategy is to minimize the maximum (or mean) vehicle response time. We focus on a static parking strategy such that AGVs do not block other AGVs when idling and we thus require a parking area (or dwell points) close to the loop, but not on it. The four major static vehicle positioning strategies proposed in the literature are (Le-Anh & De Koster, 2006):

- *Central-zone positioning rule*: a fixed parking area within the network designed to buffer idle vehicles. This area can be close to high probability P/D locations or where charging facilities are available.
- *Circulatory-loop positioning rule*: idle vehicles travel along a loop until a new job is assigned to the AGV.
- *Drop-off point positioning rule*: vehicles remain at the last job drop-off point.
- *Distributed-positioning rule*: multiple dwell points are defined as opposed to a single one. When a vehicle is idling, it is directed to one of the dwell points.

A clear benefit of using the first rule is that AGVs are always directed to the same dwell point where charging facilities are available. Another benefit compared to a dynamic strategy is that all dwell points are determined *a priori* and it thus does not require complex calculations during run-time. The second rule is not favorable in this case study as space is limited. The area does not allow for extra *idling* loops which would require extra space with no added benefit to operational performance, unless these extra loops tremendously decrease the vehicle response time, which we do not expect.

The *drop-off point positioning rule* can be interesting in our case when the job assigned to a vehicle after it has been idling is close to the drop-off point of the previous job. However this information is not known in advance and one would have to estimate the probability of the next job being close to the drop-off point of the vehicle. This could be a pick-up at i) the cross-dock, ii) the A/D parking or iii) the empty trailer parking. If we assume an equal probability of a pick-up occurring at one of these three areas, one out of three times the drop-off point positioning rule will have a low response time. In the remainder of the time, this rule will yield worse results compared to a dwell point which is more centrally located between these areas. A large down-side of this rule is the lack of charging possibilities. Depending on the arrival intensity of new jobs this might be a problem. When the arrival intensity is low, the probability of a vehicle being idle increases. When a vehicle is idling a relatively large amount of time at the drop-off point, it is draining the battery, whilst the battery could also have been charged if it was send to a charging point.

Due to the lack of information on new jobs and the absence of charging facilities it is not possible to fully benefit from the *drop-off point rule* and thus we focus on the *central-zone positioning rule*. Both methods can also be combined to create the *distributed-positing rule* where vehicles stay either at the drop-off point or are assigned to a parking area with chargers. We leave determining the added benefit of this more complex rule to further research.

5.6.1 Parking area location

Determining the optimal dwell points for AGVs to minimize the maximum or mean response time has been well-researched in literature, where most of them focus on uni- or bidirectional single loops (Gademann & van de Velde, 2000; Kim, 1995; Ventura & Rieksts, 2009). In this case we have a more complex guide-path design including a single unidirectional loop with bi-directional cross-roads per P/D location to facility docking maneuvers as well as decrease travel time. This makes it hard to formulate the problem of determining the optimal location of the parking area, especially because the probability of servicing a certain P/D location is not known at this stage. In this way no (weighted) maximum or mean response time can be calculated, unless a uniform distribution is assumed. In this case it can be easily shown that the location of the parking area minimizing the mean response time is the median location of all P/D points. With this given and without proving the optimality of the solution we allocate the parking area between the empty trailer parking and the A/D parking on the lower arc of the main single loop as shown in Figure 5-9. This has been chosen based on common sense as well as practical space requirements. It seems the most logical location as it is positioned between the main areas and from a practical point-of-view it does not require extra space as it is located at the main loop. When the system is in the future extended with a container terminal, this parking area will lie between the container terminal and the cross-dock, making the mean response time for both areas as low as possible.

The final decision to be made with respect to vehicle parking is the number of parking slots. We already assumed that charging locations are identical to parking locations, but charging stations are capital-intensive and thus minimizing this number can achieve serious cost reductions. On the other hand we do want the possibility of all AGVs being able to charge simultaneously to not let AGVs compete for charging times and to maximize the potential of full batteries. Every AGV should thus have its dedicated parking slot equipped with a charger. Note that all charging stations can also be replaced with a single battery pack swapping station. In this case the battery packs are swapped at the assigned parking location of the AGV.

5.7 Battery management

The final tactical design issue is managing the batteries of the AGVs. Electrical AGVs need charging when the battery is nearly depleted. Depending on the speed of charging and the capacity of the battery this can take a significant amount of time compared to the operating time of the AGV. Especially when the system is operating on a 24-hour basis, there is no natural 'break' to charge the AGVs. Unavailable AGVs may lead to the need for additional AGVs or increased waiting times. Effective battery management should minimize the effect of unavailability due to charging on the operational performance of the system. When diesel based AGVs are used, this is of much less concern as they can operate more than 30 hours on a single tank and are refueled within a couple of minutes (see Section 4.5.2). As diesel based AGVs do not pose any challenges regarding recharging/refueling, we focus on the battery management of electrical AGVs.

The review of Le-Anh & De Koster (2006) underline the lack of research on battery management for AGVs. Battery management is usually omitted from research and thus failing to address real-life challenges on implementing AGV based systems. One of the few exceptions is the work of McHaney (1995), which presents three types of charging strategies:

- *Opportunity charging*: AGVs are send to charging stations whenever they become idle.
- *Automatic charging:* AGVs run until their battery is nearly depleted and it then send to a charging station.
- *Combination charging*: a combination of the previous two.

The first strategy seems suitable when trying to maximize the battery level of all AGVs as they are charged whenever possible. From a technical point-of-view the second strategy is more favorable as this is beneficial for the life span of the battery. This strategy is of course always used when the battery of an AGV is low to not drain the battery to zero, stalling the AGV. Note that an AGV only becomes idle when all jobs of the current schedule are processed or the earliest start time of the next job is not yet reached. In the previous section we already determined that in the first case AGVs are send to the parking area, where chargers are available. However in the latter case it may or may not be beneficial to charge the AGV depending on the slack between the start time of idling and the earliest start time of the next job.

When it is decided to charge the AGV it needs to travel from its current position to the parking area. Suppose that this travel time is similar to the slack time, the AGV will arrive at the parking area and immediately needs to leave to not postpone the next job. It may even occur that the travel time of the tour via the parking area is larger than immediately moving to the next job and so the battery will be somewhat more depleted when assigning to the parking area. This is of course counter-productive and it thus makes sense to define a certain threshold value δ which should be exceeded before deciding to send the AGV to a charger. This threshold value represents the minimum charging time of the AGV, expressed in time units and can also depend on the current battery level. When the battery level is low and there is no idle time available for charging, the AGV has to be send to the charger regardless of the jobs already assigned to the AGV. Due to the multi-stage bidding protocol defined in Section 5.5.1, jobs that are delayed due to these charging efforts can be reassigned to other AGVs when this yields a better solution. The threshold value δ can thus be expressed as:

$$\{S_l - (T_{kp} + T_{pl})\} + M(1 - Y) > \delta$$

where S_l is the slack time until the next job l, T_{kp} is the total travel- and handling time of the AGV from its current location k to parking slot p, T_{pl} is the total travel- and handling time between parking slot p and the location of job l, Y is a binary variable indicating whether the battery level of the AGV is above a predetermined value K (then $Y = 1$) and M is a big number ($M > \delta$). The value K represents the absolute necessity of charging the AGV, otherwise it will come to a standstill. The value of K will be at least the percentage of battery required to process a job with the largest travel time possible, ending at the position furthest away from the parking area and in addition the battery percentage required to move from this position to the parking area. In this way any job l can be processed without stalling the AGV as long as the battery level exceeds the value of K at the start of processing the job. When δ is not exceeded the AGV will move to job l and wait there. It can for example couple a trailer at the cross-dock but not move away yet or position itself at the A/D parking and wait until the job is ready to be processed.

In addition to charging the AGVs, the batteries also need to be levelled after around 140 operating hours. This has to be done to make sure all battery cells charge equally. As Rotra operates 24 hours a day, 6 days a week (i.e., 144 hours), we assume that the batteries are levelled during system down time (on a Sunday).

5.8 Vehicle routing

On the operational level the conflict-free routing of AGVs has to be addressed given a certain sequence of jobs. The routing, scheduling and conflict resolution of AGVs are closely related and are thus addressed concurrently. The scheduling question has already been addressed on the tactical level, deciding upon *when* AGVs should process a transportation order and on the operational level the route the AGV should take to minimize the processing time and possible conflicts are addressed. Calculating and recalculating the optimal routing solution in a dynamic environment should not take too much computation time as the environment can change before a new optimal solution has been found. In contrast to conventional routing problems, AGVs need to have a certain look-ahead capability which foresees the dynamics of the vehicles around it in order to avoid collisions by stopping, slowing-down or taking an alternative route (Qiu et al., 2002). Especially in our mixed uni-bidirectional lay-out with multiple AGVs transporting large-size and heavy cargo, optimal routing and collision avoidance are essential. Let us first distinguish four types of jobs for which routes have to be found:

- *Cargo arrival:* the AGV should pick-up the cargo at the A/D parking and dock it at the cross-dock.
- *Cargo departure:* the AGV should undock the cargo and park it rearwards at the A/D parking.
- *Empty trailer pick-up:* the AGV should pick-up an empty trailer at the empty trailer parking and dock it at the cross-dock.

- *Empty trailer drop*-off: the AGV should undock an empty trailer at the cross-dock and rearward park it at the empty trailer parking.

For every job multiple routes can be taken where only one of them has the shortest distance. The route depends on whether or not the AGV is loaded. When the AGV is loaded, the length of the total combination increases and it thus needs more maneuvering space. Determining which is the best route r, comprising of a set of arcs (i, j), to take from a set of available routes R $(r \in R)$ does not solely depend on the distance, but also on expected waiting time on the route due to other AGVs occupying the same section of the route at the same time. One way of coping with this is to determine a time-based routing algorithm where certain arcs (i, j) are occupied during time-window $[t_1, t_2]$ and the algorithm should find unique time-windows for all arcs (i, j) to minimize waiting time due to congestion. A simpler approach would be to use forward sensors on the AGVs to assess near-collision issues and based on priority rules, make stop,- go,- and slow-down decisions. This approach seems to fit our case study as i) AGVs travel at the same speed and thus head-to-tail collisions are unlikely to occur ii) AGVs do not block the main loop in our design when rearward docking and iii) jobs of the same type always move in the same direction. To elaborate on this, suppose two AGVs need to perform the same type of job (e.g. a cargo arrival job). Both AGVs then need to be headed from their current position to the A/D parking and position themselves rearwards to the cargo. Note that these pick-up locations are always unique within a certain time frame. The AGVs then needs to back-up and couple the trailer, drive to the cross-dock, position itself rearwards to the dock and then dock the trailer. During this entire operation, both AGVs will never collide when there is enough space between the AGVs, depending on the cargo length, as they move with the same speed. When the AGV in front makes a turn at the cross-dock it will free up space on the main loop and thus the second AGV can pass the first one when its destination is further away than that of the first AGV.

We can therefore safely make use of the shortest path between locations i and j (depending on whether or not the AGV is loaded and the length of the cargo) and let the Conflict Manager be in charge of making sure the shortest path is also the shortest-time path (whenever possible). We thus determine *a priori* the shortest paths, using the guide-path lay-out, between all P/D locations for three cases: a) an empty AGV b) an AGV loaded with a standard-size semi-trailer (13.62m) and c) an AGV loaded with a LHV (max. 25.25m).

5.9 Conflict resolution

Together with vehicle routing, resolving conflicts during system uptime is part of the operational decisions. As already noted, the conflict resolution function should accomplish a collision-free environment and minimize the waiting time due to congestion based on priority rules. The following conflicts are possible when routing AGVs (Qiu et al., 2002):

- *Collisions:* when two or more AGVs want to make use of the same segment at the same time, there will be a collision. Note that in our case study the cross-roads are only accessed by two AGVs at the same time when i) one AGV docks a trailer at a certain dock and another AGV parks a trailer at the empty trailer parking and the docking location and the parking slot are opposite to each other or ii) during docking at the cross-dock or parking at the empty trailer parking another AGV wants to use this cross-road to move from the one side of the main loop to the other. See Figure 5-13a for examples.
- *Congestion:* Congestion occurs when during a certain period of time the number of arrivals is greater than the capacity of a road segment. See Figure 5-13b.
- *Livelocks:* A live lock may occur when traffic on the main road is given priority to the traffic on the cross-roads such that the traffic on the cross-road waits indefinitely. This occurs only when the traffic intensity is high on the main road (see Figure 5-13c).
- *Deadlocks:* These types of locks occur when multiple AGVs mutually wait for release, which will never occur (see Figure 5-13d).

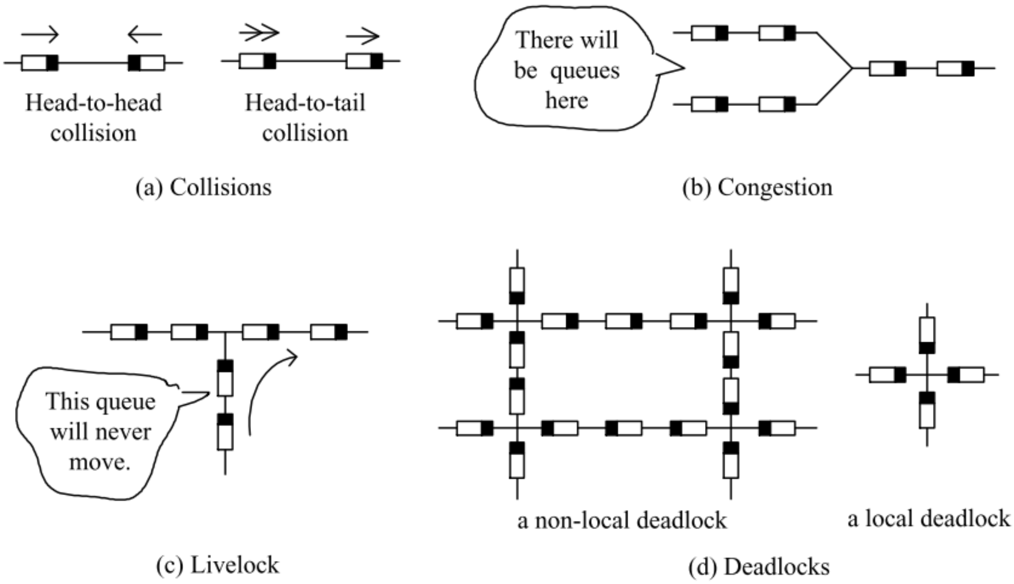
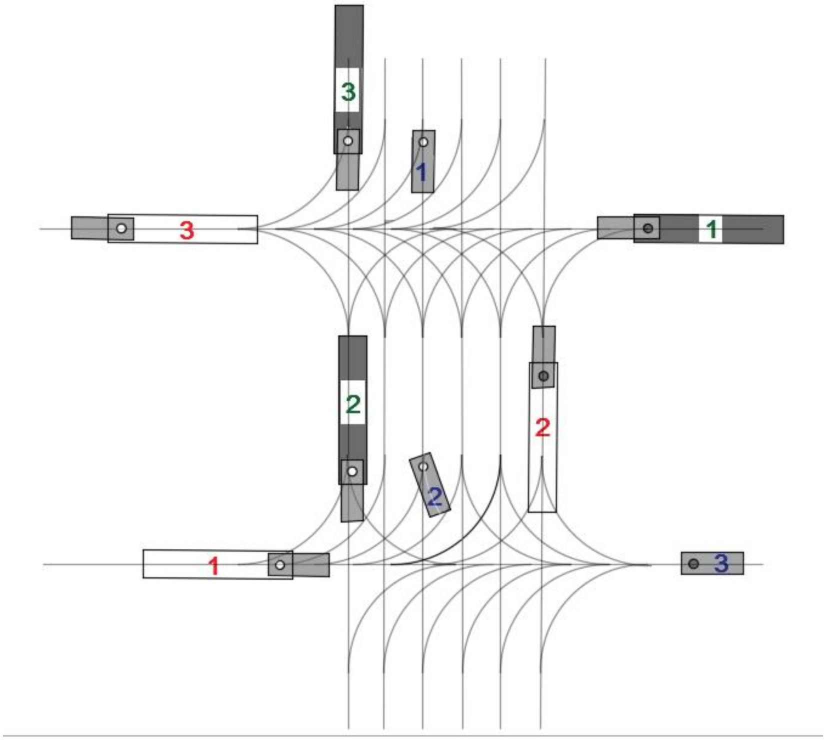


FIGURE 5-13 POSSIBLE CONFLICTS (ADAPTED FROM QIU ET AL., (2002))

Resolving these issues is part of the Conflict Resolution manager within the MAS and is important as these issues can have negative consequences on system performance. It must however be noted that these issues only pose serious problems when the traffic intensity is high and thus many AGVs are deployed. We do of course want to provide a generic solution, thus also to high traffic intensity instances, but note that when this is the case the lay-out of the guide-path will probably be of a larger scale then when fewer AGVs are deployed and thus conflict issues are less likely to arise. We use the following job priority within the system, from highest priority to lowest:

1. Cargo arrival, AGV loaded
2. Cargo departure, AGV loaded
3. Cargo arrival, AGV unloaded
4. Cargo departure, AGV unloaded
5. Empty trailer movement, AGV loaded
6. Empty trailer movement, AGV unloaded
7. AGV to AGV parking

We thus put the highest priority on cargo arrivals and departures. In general, arriving cargo needs to be unloaded as quickly as possible to not delay departing cargo. When cargo has been loaded, it is ready to leave the terminal and thus needs to be dropped-off at the A/D parking as quickly as possible. Loaded AGVs are given priority to unloaded AGVs as they can start earlier with further processing (unloading, departing cargo) and loaded AGVs are heavier and thus require more time and space to stop and pull up again. The checks on priority ruling are made at every intersection within the guide-path layout, making sure to look ahead enough intersections, such that the AGV can stop or slow-down within this space without conflict. This is illustrated in Figure 5-14. This is a snapshot of the guide-path design as previously defined. The numbers represent the order in which the AGV will move. In this situation we have i) a cargo arrival with a loaded AGV (green), ii) an AGV loaded with an empty trailer (red) and iii) an unloaded AGV (blue) all at the same time at their initial position. According to the priority rules the green AGV moves first from position 1, to 2 and then backwards from 2 to 3. The empty trailer can move from position 1 to 2 without conflict and waits there until the green AGV is at position 3. The blue AGV waits at position 1 until the green and red AGV are both in position 2. As the red AGV has to wait until the green AGV is in position 3, the blue AGV can move to position 2 and 3 when the red AGV is still in position 2. This illustrates that the priority ruling only applies when AGVs compete for the same space at the same time. Note that the probability of this happening is expected to be low as in our case study the expected number of AGVs is three and the total number of P/D locations is around 200. Nonetheless, if it occurs it should be handled properly by the Conflict Manager.



5.10 Conclusion

This chapter has discussed various ways to develop agent intelligence based on the capabilities of every agent based on an AGV framework. Besides the design of the guide-paths and determining the required number of AGVs, we decided upon the decision capabilities for all agents such that they fulfil their design objectives. We summarize our findings per agent below:

- **Demand Manager**
  This agent does not require complex algorithms, but instead provides the link between external systems (e.g. TMS) and the MAS. It is however of utmost importance that all data required is readily available and up-to-date.
- **Parking Manager**
  We use a *nearest-available* rule for all parking areas. Arriving and departing semi-trailers are assigned to the first available (i.e. not occupied) parking slot. This results in an ordered priority list where parking slot 1 is more favorable to parking slot 2 and so forth.
- **Vehicle Scheduling Agent**
  A very common way of scheduling within MASs is using an auction mechanism where AGVs compete for orders. The Vehicle Scheduling agent initiates a proposal when a new job enters the system via the Demand Manager (e.g., pick-up at cross-dock). All AGV agents evaluate this proposal and send back a bid. Based on a bid evaluation function the winner is announced. We use the auction mechanism as described in Mes, van der Heijden, & van Harten (2007).

- **Vehicle Routing Agent**
  The routing agent determines which route the AGV should take given the pick-up and drop-off location of the job. We use the shortest-path method and determine *a priori* the shortest paths between all nodes using the guide-path defined above.
- **Battery Manager**
  The Battery Manager uses an opportunity charging strategy as defined by McHaney (1995). Whenever an AGV is idling it is send to the AGV parking. The Battery Manager also monitors the battery level of all AGVs and asks the Vehicle Scheduling agent to schedule a charging job whenever the battery is below a certain threshold.
- **Conflict Manager**
  This agent's responsibility is to avoid collisions, congestion and deadlocks. We use a priority list to make stop-and-go decisions based on the current status of the AGV. When two or more AGVs want to use the same arc at the same time, the Conflict Manager evaluates which AGV has priority based on the current jobs the AGVs are processing and stops the AGV with the lowest priority.
- **AGV Manager**
  The AGV Manager has an important role within the MAS as it feeds information about the AGV to the system (e.g. current position and battery level). It furthermore uses the bid calculating function as described above to respond to the proposals of the Vehicle Scheduling agent.

6 CONCEPTUAL MODEL

This chapter provides a conceptual simulation model of the Multi-Agent System presented in the previous chapters. This chapter will guide the way to the design of a valid, credible, feasible and useful simulation model. Section 6.1 is an introduction to this chapter. Section 6.2 describes the conceptual model. Section 6.3 contains the content of the model. Section 6.4 outlines our conclusions.

6.1 Introduction

Thus far we have presented a multi-agent system for the planning and control of autonomous vehicles capable of moving and rearward docking semitrailers at a (multimodal) cross-dock. We continued by using the AGV framework of Le-Anh & De Koster (2006) applied to the Rotra case study to fully comprehend the functionalities and intelligence required for the agents in the system to fulfill their design objectives. Opposed to a real system, this is a proposed system and thus we need a simulation model to predict how the system will perform under certain operational conditions. Before we design a computer model, we first need a conceptual model which is a non-software specific description of a computer simulation model, describing its objectives, inputs, outputs, content, assumptions and simplifications (Robinson, 2008a). The remainder of this chapter provides the conceptual model step-by-step, from general modeling and project objectives to agent-specific modeling concepts and concluding with a concise overview of the conceptual model.

6.2 Conceptual model framework

To use a structured step-by-step approach of conceptual modeling we use the framework presented by Robinson (2008b) which consists of the following five steps (see Figure 6-1):

- Understanding the problem situation,
- Determining the modelling and general project objectives,
- Identifying the model outputs (responses),
- Identifying the model inputs (experimental factors),
- Determining the model content (scope and level of detail), identifying any assumptions and simplifications.

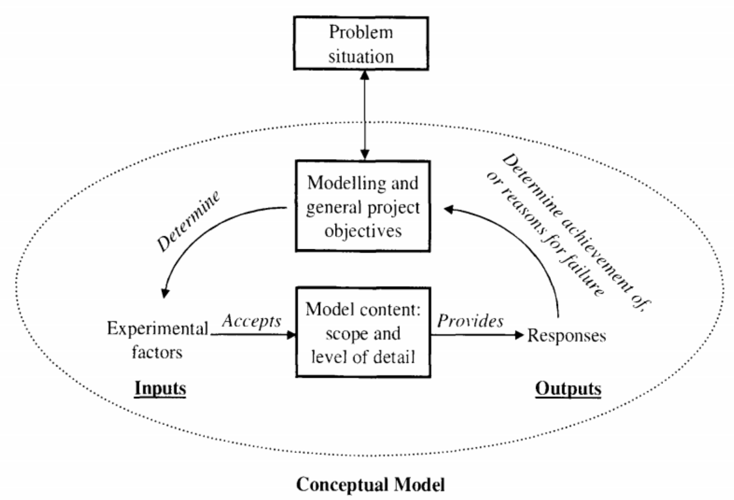


FIGURE 6-1 FRAMEWORK OF CONCEPTUAL MODELLING (ADAPTED FROM (ROBINSON, 2008B)

6.2.1 The problem situation

The key requirements of a conceptual model are validity, credibility, feasibility and usefulness (Stewart Robinson, 2008a). Understanding the problem situation, which is input for the conceptual model, is thus an important first step. We elaborated on the problem situation in previous chapters, but it remains to link the problem situation to the conceptual model requirements. In the Rotra case the model should:

- Provide sufficiently accurate insight on the number of AGVs required in the system to handle a certain number of semi-trailers within predefined time intervals (*validity*).
- Rotra and other consortium partners must have confidence in the model (*credibility*).
- Be *feasible* to build within given time and data constraints.
- Be *useful*, that is, flexible, applicable to a multitude of scenarios, visually attractive and modularly build (the ability to extend the complexity of the model by other researchers).

To summarize, the problem situation of the case study: Rotra wants to research the possibility of using autonomous vehicles to shunt semi-trailers at a new to-be-build multimodal cross-dock. Within this research an important factor is the number of AGVs required to process all semi-trailer movements during the day, given certain arrival- and departure times.

6.2.2 Modelling and general project objectives

The objective of Rotra is to minimize the Total Cost of Ownership (TCO) of the AGVs. The model should therefore provide insight in the number of AGVs required, given a certain amount of semi-trailers arriving and departing during the day. This also involves assessing the impact of the number of AGVs on waiting times and the throughput and responsiveness of the system. We will use a time-scale of 6 days a week, which is current practice at Rotra, and try to provide the greatest amount of flexibility possible in the model such that is can be used to do many different experiments. We also need many runs per experiment to be able to do good statistical analysis of the impact of the experimental factors used in the different experiments. Furthermore we use a simple 2D visualization and the model should be understood by a moderately experienced modeler such that he can further extent the system or do more experiments.

6.2.3 The model outputs

- Time-series of daily throughput and cycle times of arriving and departing semi-trailers
- Bar chart of daily AGV occupancy (working, idle, stopped, waiting, charging).
- Mean, standard deviation, minimum and maximum daily throughput of all docks
- Mean, standard deviation, minimum and maximum occupancy of the Arrival/Departure Parking and the Empty Trailer Parking
- Mean, standard deviation, minimum and maximum daily; number of jobs assigned, utilization, response time, travel time, waiting time, charging time and idle time of all AGVs.
- Mean, standard deviation, minimum and maximum daily conflicts resolved by Conflict Manager.

6.2.4 The model inputs

- Number of docks
- Number of arriving and departing transports
- Number of AGVs deployed
- Number of parking slots at Arrival/Departure parking
- Dimensions of guide-path design (distance between docks, distance between parking slots, length dock area, length crossroads, length parking slots)
- Number of loads per arriving semi-trailer (uniform distribution with a minimum 10 and maximum value)
- Handling time per load

- Look ahead length to determine how far the Conflict Manager should check for the presence of other AGVs.
- Maximum number of semi-trailers allowed in system
- (De)coupling time of semi-trailer with a truck
- (De)coupling time of semi-trailer with an AGV
- Forward and rearward driving speed of AGV
- Charging alternative of AGVs (continuous, battery swap, conventional charging).
- Parking Manager policy (available slot closest by)
- Vehicle Routing policy (shortest path)
- Vehicle Scheduling policy (auction mechanism and bid evaluation as discussed in Section 5.5)
- Conflict Manager priority ruling (as discussed in Section 5.9)
- Battery Manager policy (continuous, battery swaps or conventional charging)

6.2.5 Model scope and level of detail

The simulation model includes all operations required to pick-up and drop-off semi-trailers at various locations around the cross-dock at pilot location Velp using AGVs. We focus on the shunting of semi-trailers only and assume that the heterogeneous fleet can be viewed as homogenous when it comes to the maneuvering-, coupling- and driving times. We also assume that the inter-arrival time of semi-trailers follow a certain probability distribution. We further reviewed the worst-case-scenario (3-axle non-steered semi-trailer) for the guide-path design and use the distances associated within this guide-path design as if they were applicable for all semi-trailers. The impact of this simplification is assumed to be very small compared to the non-type specific handling and driving times (e.g. average speed of AGV does not depend on the length or load of a semi-trailer). We further ignore the inner operations of the cross-dock such as loading/unloading and assume that this requires a predetermined amount of time and it never raises any exceptions or delays. We thus also make estimates on the consolidation (which arriving trailers contain how much cargo for which departing trailers). We also assume that AGVs never break down and maintenance is carried out during system down-time. The AGVs in the system are modeled as a black-box and driving from an origin to a destination takes a certain amount of time (based on the maximum forward speed) as well as rearward docking (based on the maximum rearward speed). Next to that, the coupling and uncoupling of semi-trailer with the AGVs is assumed to never fail and requires a fixed amount of time. A detailed overview of the model scope and assumptions can be found in Appendix 5 and the corresponding level of detail in Appendix 6.

6.3 Model content

The following paragraphs discuss pieces of the conceptual model per agent in the system. The color of the flowcharts within the paragraphs are identical to the unique color of each agent, as consistently used in this thesis. We discuss the most important events for every agent and also some generic events that do not fall within the scope of a single agent.

6.3.1 Model initiation and reset

The first step we need to do, is to initialize the model, creating all required parameters, variables and objects for the model to be able to start working. Also all the model inputs need to be assigned to variables in the model in such a way that they can be easily changed to represent new experiments or when new data comes available. The most important part of the initialization is creating a track system which represents the guide-path design we have defined in Section 5.3. We therefore use an initialization method (*Init*) which does all work necessary to get the model up and running every time we run the model and use a reset method (*Reset*) to delete the track system of the previous run. We can then adjust the input parameters or experimental factors (e.g. more docks) to represent a new scenario and run *Init* to initialize this new situation. A technical description on how to initiate a track system using variables, can be found in Appendix 7.

6.3.2 The Demand Manager

The Demand Manager is one of the first agents to implement in the model. Its goal is to keep track of arriving and departing transports, logging these in databases and passing relevant information to other agents in the system. In practice the Demand Manager gets its information from external sources such as Transport Management Systems (expected arrival times, due dates, etc.) and Warehouse Management Systems (information on loads, consolidation of cargo and dock assignment). In the simulation model without external connections, we use probability distributions to generate data which is similar to the data we would get from the external sources. This simplification and abstraction is already discussed in Section 6.2.5. The appropriate place to model this data generation, is within the Demand Manager as it is its responsibility to take care of all demand related information.

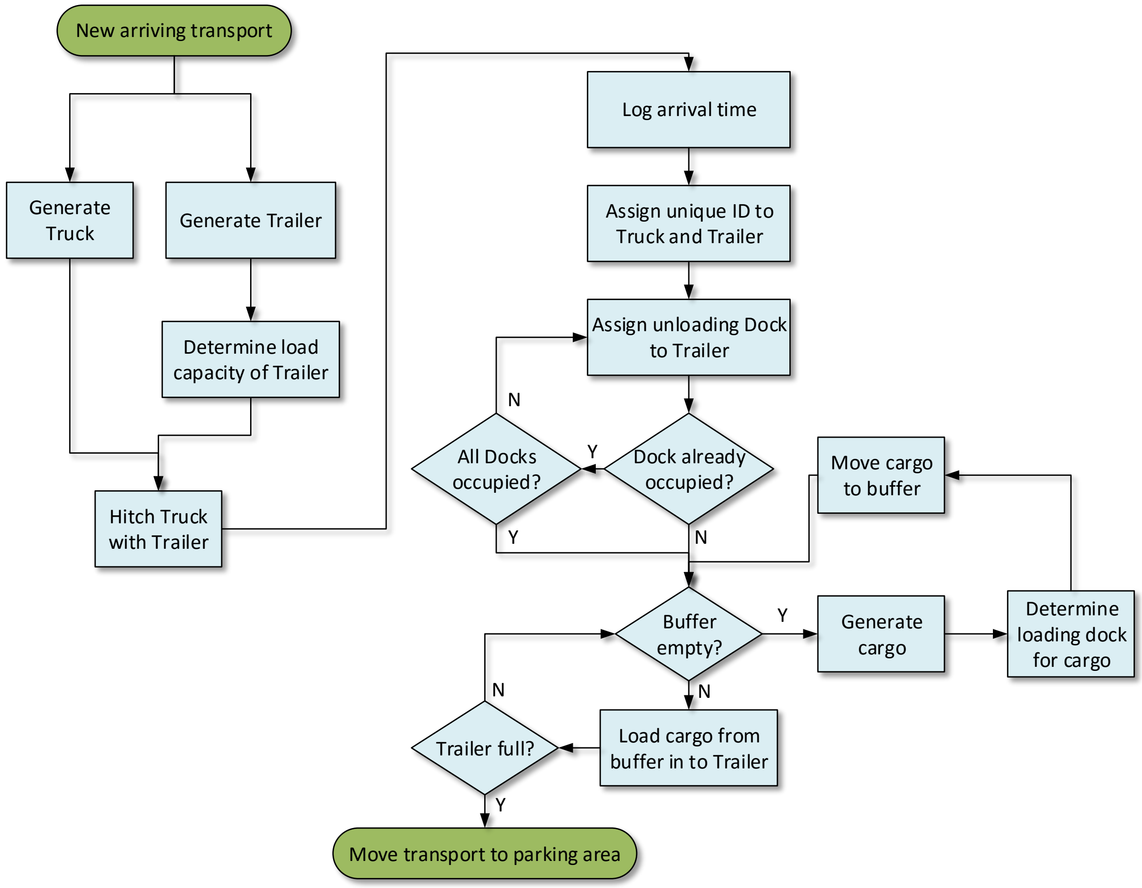


FIGURE 6-2 FLOWCHART 'NEW ARRIVING TRANSPORT'

Figure 6-2 shows how the Demand Manager responds to an event *New Arriving Transport*, generating and logging all information required for the rest of the system to function properly. This event is triggered when a new arriving transport is generated by the model, either already on the terrain of the DC or in close proximity of the DC, depending on the model preferences. First we need to determine what the unloading time of the trailer is, and this depends on the number of pallets (or other cargo) in the trailer. We use a random distribution to determine the capacity of the trailer and we always fully use this capacity. By adjusting the minimum and maximum value of the capacity and the processing time per unit of cargo, we can control unloading times and effectively simulate actual unloading times (e.g. between 15 minutes and 3 hours). Another important aspect is determining the unloading dock. We pick a random unloading dock, resulting in an even distribution among the docks, making sure to pick a different dock when the dock chosen, is already occupied. We then load the trailer with cargo until the trailer is full. Every piece of loaded cargo is also assigned a departure dock, thereby creating the consolidation function of the cross-dock, as every arriving trailer contains cargo for multiple docks, from which the departing transports will load their cargo. All trailer and cargo data is stored within a database accessible to the Demand Manager.

Another important event for the Demand Manger is the unloading of cargo at the cross-dock. When a piece of cargo is unloaded from an arriving trailer, it is put in a buffer corresponding to the departure dock the cargo was assigned to. This buffer resembles the space inside the cross-dock where cargo is temporarily stored awaiting departure. Figure 6-3 shows how the system checks whether all cargo that has been assigned to this buffer is present. Another check is made to ensure that the system is not flooded with trailers. When for example all docks are occupied and there is no free space at the empty trailer parking, trailers that have been unloaded cannot go the parking. On the other hand, buffers who want to load their cargo into a departing transport, cannot find an available dock. To overcome this problem, whenever the system limit is reached, an extra departing transport is initiated containing the cargo currently in the buffer. This will free up one dock and one empty trailer parking slot. The remainder of the cargo assigned to the particular buffer (that has not arrived in the buffer yet) will depart in a different trailer.

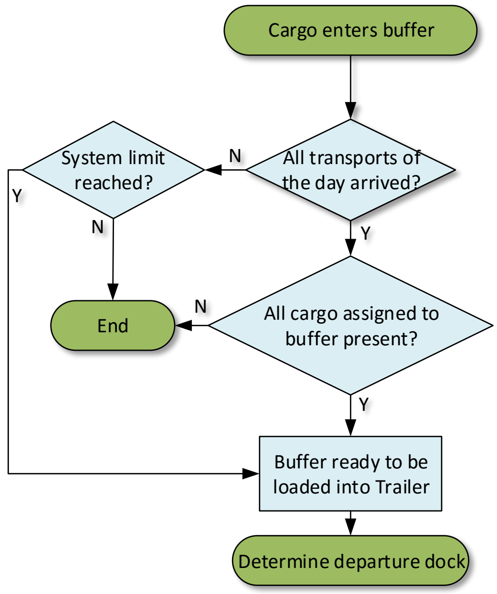


FIGURE 6-3 FLOWCHART 'CARGO ENTERS BUFFER'

Whenever the buffer is ready for departure, which thus may not hold all cargo assigned to this buffer, another method determines the closest available dock where the cargo in the buffer can be loaded into a departing trailer. The closest available dock is preferred to the dock right in front of the buffer as this dock may be occupied (e.g. due to unloading of a different trailer). It would be sub-optimal to wait for this dock to become available, if a dock close to it, is already available for departure. The trade-off is between the waiting time saved due to choosing a different dock and the extra time it takes to move the cargo inside the cross-dock to a dock which is further away. As we do not focus on the internal operations of the cross-dock, our preference goes out to chosen the first available dock. More information on how the closest available dock is chosen and subsequently, which empty trailer is selected for the departure, can be found in Appendix 8.1.

6.3.3 The Parking Manager

The Parking Manager frequently interacts with the Demand Manager, assigning parking slots to arriving and departing transports, but also assigning parking slots for empty trailers whenever the Demand Manger requests one. The flowchart of Figure 6-4 shows how the Parking Manager handles an arriving transport. From this figure we can see that the Parking Manager is also actively involved in preventing the flooding of the system, as it only interacts with the Vehicle Schedule Agent when he seems fit (i.e., dock available and system limit not reached). The trailer waits at the Arrival/Departure parking until both conditions are met.

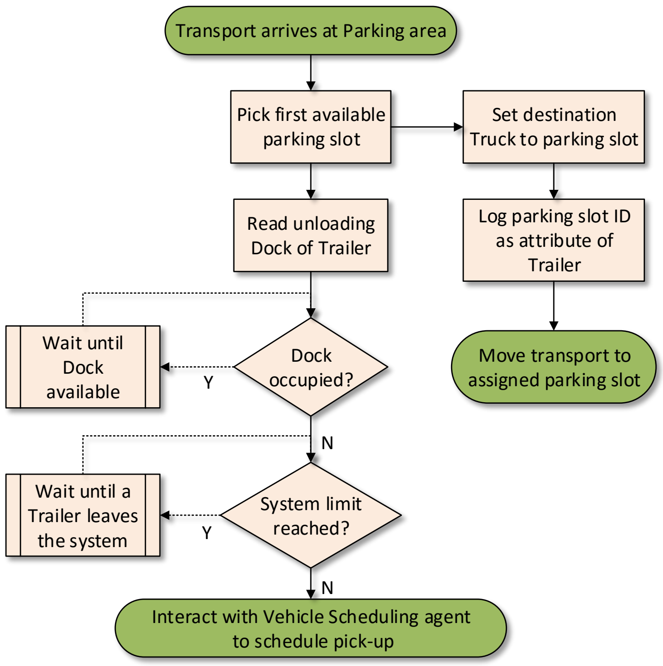


FIGURE 6-4 FLOWCHART 'TRANSPORT ARRIVES AT PARKING'

An event related to the Parking Manger is the actual arrival of a trailer at a parking slot (A/D Parking or Empty Trailer Parking). First of all the trailer needs to be unhitched from the carrying unit (truck or AGV) which takes a certain amount of time. Depending on whether the event was triggered by a truck or an AGV, it will either remove the truck from the system or continue with the schedule of the AGV, as shown in Figure 6-5. Other events related to the Parking Manager are requests for a parking slot for departing transports and request for a parking slot at the Empty Trailer parking, both of which are presented in Appendix 8.2.

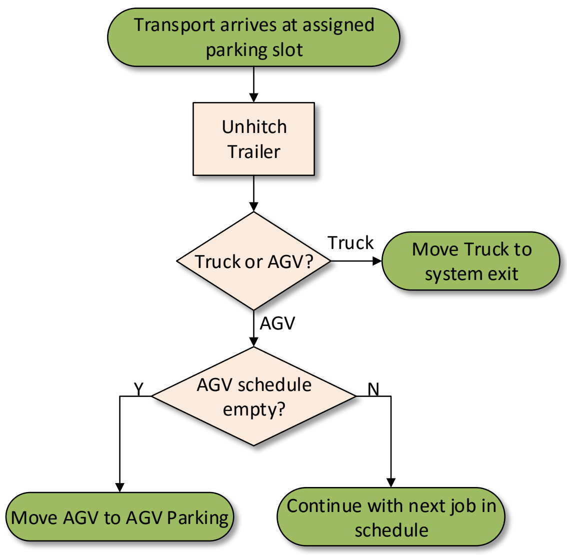


FIGURE 6-5 FLOWCHART 'TRANSPORT ARRIVES AT ASSSIGNED PARKING SLOT'

6.3.4 Vehicle Scheduling Agent

Whenever a trailer needs to be pick-up or dropped-off, the Vehicle Scheduling Agent is triggered to assign an AGV to the job. There are a variety of jobs (discussed in Section 5.9), but they all undergo the same process within the scheduling agent. The agent first receives information on the type of job, determines its priority and collects information on the origin and destination of the job. The auction protocol is then initiated and the auction winner is assigned the job. When the current position is at the AGV Parking Area, it will first determine how much time is left until the earliest possible start of the job. When the AGV is able to drive to the origin of the job within this amount of time, it has some slack. The AGV will wait this slack time at the AGV Parking Area as it rather waits there (and continue charging) than wait at the origin of the job where no charging facilities are present. This is of less importance when swappable battery packs are used for the AGVs. However, another benefit is the absence of an idling AGV, and thus not blocking a part of the track system. The guide-path is designed in such a way that the AGV never blocks other AGVs when stationed at the AGV Parking Area. The scheduling process is shown in Figure Figure 6-6.

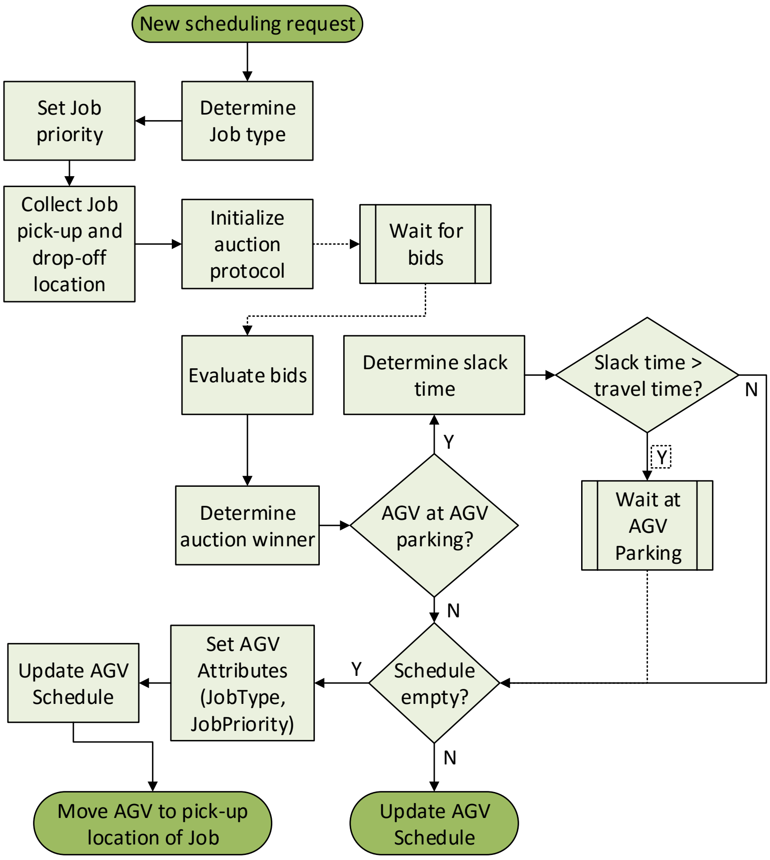


FIGURE 6-6 FLOWCHART 'NEW SCHEDULING REQUEST'

Closely related to this event are bid calculations by the AGV Managers and route calculations between the pick-up and drop-off locations by the Vehicle Routing Agent. Both of these are discussed in their respective paragraphs below.

6.3.5 AGV Manager

Whenever a job needs to be scheduled, the AGV Manager receives a Call for Proposal (CFP) from the Vehicle Scheduling agent. The AGV Manager responds to this CFP with a bid, using the total travel-, handling- and waiting time required for the AGV to process the job given its current schedule. The total travel time depends on the length of the route(s) connecting the pick-up and drop-off locations of the job. This is calculated by interacting with the Vehicle Routing agent, which returns the total travel time to the AGV Manager as shown in Figure 6-7. The AGV Manager not only responds to bids, but also continuously processes real-world input from the environment (e.g. current location and battery status) and feeds this to the MAS.

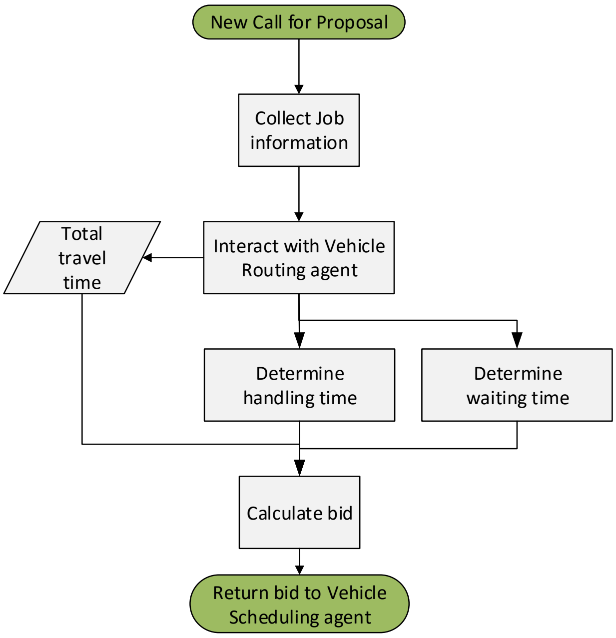


FIGURE 6-7 FLOWCHART 'NEW CALL FOR PROPOSAL'

6.3.6 Vehicle Routing Agent

The main tasks of the Vehicle Routing Agent are (i) determining the shortest route between P/D locations and (ii) determining the shortest path between the location of the AGV and the P/D locations. The AGV Manager passes information on the P/D locations of a job to the Vehicle Routing Agent. Based on this information and the current schedule of the AGV, the agent determines the shortest route and calculates the total forward and rearward driving time based on the AGV characteristics. This information is passed back to the AGV Manager as shown in Figure 6-8.

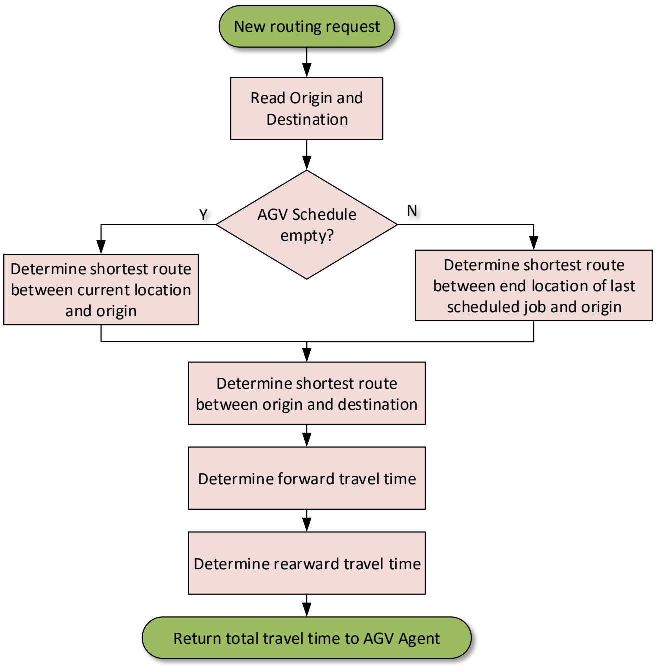


FIGURE 6-8 FLOWCHART 'NEW ROUTING REQUEST'

6.3.7 Conflict Manager

The Conflict Manager makes sure all AGVs do not collide or create deadlocks. In this section we focus on the traffic control of the unidirectional single loop connecting the main areas at the terminal. The single loop consists of a top part (along the cross-dock and the A/D parking) and a bottom part (along the Empty Trailer Parking and AGV Parking). The single loop consists of connected track segments all with an equal length (equal to the distance between the docks). On every one of these segments a sensor is placed. Whenever an AGV hits a sensor, the checks shown in Figure 6-9 are done. For both the top and the bottom part of the single loop, the next track segment, the next curve segment (connecting the single loop with the crossroad) and the next crossroad is checked whether it is occupied. If the next track or curved segment is occupied, the AGV that triggered the sensor is stopped. If the crossroad is occupied it is checked whether the AGV on it is currently driving backwards. This means the AGV on the crossroad is busy with a docking or parking maneuver. If this is the case, the AGV on the crossroad is given priority. If the AGV on the crossroad is not driving backwards, but this crossroad *is* part of the route of the AGV who triggered the sensor, the AGV on the crossroad is also given priority as this crossroad should be unoccupied before the other AGV can enter it. If the AGV is driving on the top part of the single loop, the dock area is also checked for the presence of another AGV. It is checked whether this other AGV is still hitched to the semi-trailer (it is thus busy with unhitching). If this is not the case, the AGV who triggered the sensor is stopped. This means that the AGV leaving the dock area is given priority to the AGV on the single loop. For the segments on the bottom part, this check is comparable, but it checks the next empty trailer parking slot instead of the next dock. If all checks are negative, the following set of tracks, curves, docks, parkings and crossroads are checked. This continues until the 'look ahead length' is reached. This is an integer value representing how many successive track segments are checked for occupancy and thus indirectly determines the minimum distance between AGVs on the single loop.

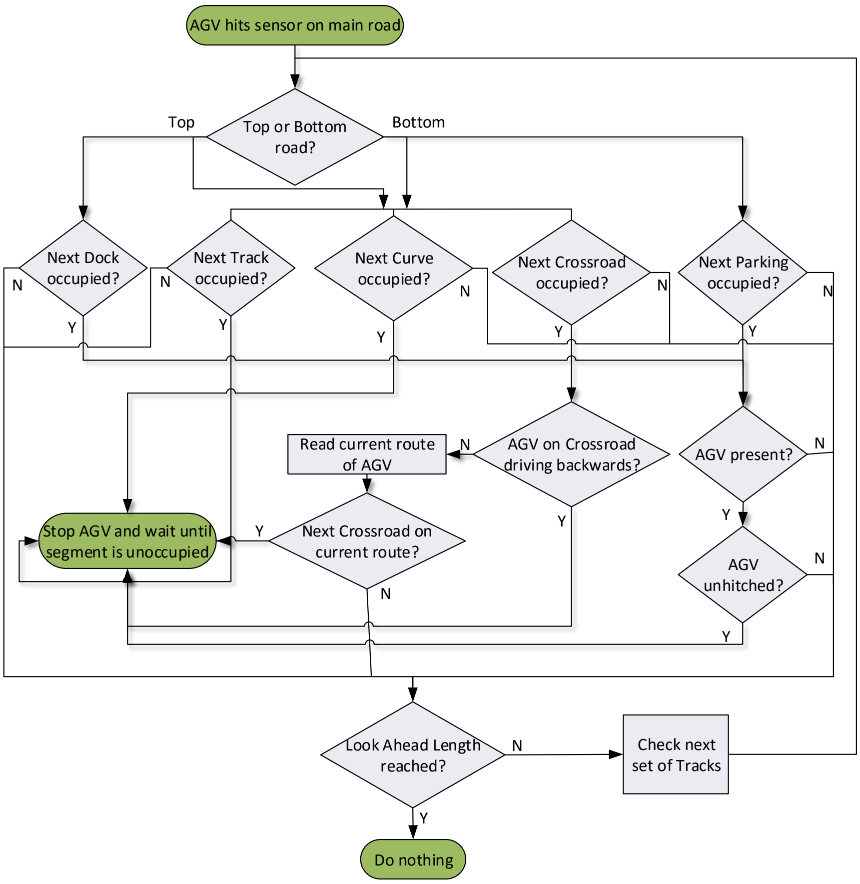


FIGURE 6-9 FLOWCHART 'AGV HITS SENSOR ON MAIN ROAD'

6.3.8 Battery Manager

The Battery Manager monitors the battery status of all AGVs and determines whether it is time for an AGV to go to a charger (i.e., the AGV parking area). Checking the battery status is triggered every time a new job is added to the schedule of an AGV. For every job in the schedule of an AGV the estimated ready time of the job is incorporated. Based on this ready time, the Battery Manager estimates when the battery is below a certain threshold value (such that it can reach the AGV parking area before the battery is fully drained). This process is shown in Figure 6-10. To exemplify, a new job added to the schedule may be the 5th job in the current schedule. The ready time of this new job is 20 minutes from now. Based on the current battery level and the consumption rate, the Battery Manager can check whether the battery is below the predefined threshold value in 20 minutes from now. When this is the case, a charging job is added to the schedule. This job sends the AGV to its dedicated parking slot and also adds the ready time of charging the AGV to the schedule, such that the AGV Agent incorporates the (mandatory) charging time in its bid calculation when a new CFP reaches the AGV Agent during charging or when a charging job is already scheduled.

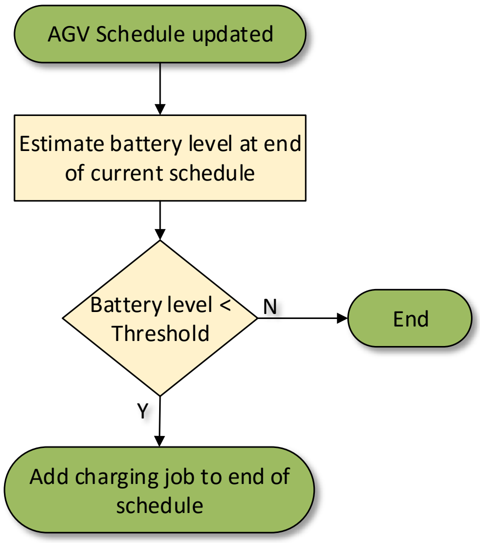


FIGURE 6-10 FLOWCHART 'AGV SCHEDULE UPDATED'

6.3.9 Connecting the agents

The events presented in the previous paragraphs all make up a small part of the system. Many of these events contain triggers or end with triggers for other events. To exemplify this interconnectivity, Table 6-1 shows how the flowcharts presented above are linked to each other. Note that not all events discussed are shown, as some end events do not trigger other events or trigger trivial events which we left out of the discussion.

TABLE 6-1 INTERCONNECTIVITY OF EVENTS

Event	Wait until	Intermediate / end event	Trigger of
New arriving transport		Move transport to parking area	Transport arrives at parking area
Transport arrives at parking area		Move transport to assigned parking slot	Transport arrives at assigned parking slot
	1. Arrival dock available 2. Number of trailers below system limit	Interact with Vehicle Scheduling Agent	New scheduling request
Transport arrives at assigned parking slot	Truck unhitched semi-trailer	Move truck to exit AND AGV to AGV Parking OR continue with next job in schedule	
New Scheduling request	Return of all bids	Update AGV schedule	New Call for Proposal
		Move AGV to pick-up location of job	AGV Schedule updated
New Call for Proposal		Interact with Vehicle Routing Agent	New routing request
		Return bid to Vehicle Scheduling Agent	
New routing request		Return total travel time to AGV agent	
AGV Schedule updated		Add charging job to AGV schedule	
AGV hits sensor on main road	Segment is unoccupied	AGV hits sensor on main road	Stop AGV
Cargo enters buffer		Determine departure dock	New departure dock request
New departure dock request	Semi-trailer is pick-up by AGV	Interact with Parking Manager	New empty trailer request
New empty trailer request		Interact with Vehicle Scheduling Agent	New scheduling request
Empty trailer arrives at dock	Trailer is loaded	Interact with Parking Manager	Departure parking request
Full trailer arrives at dock	Trailer is unloaded	Interact with Parking Manager	New parking slot request ET parking
Departure parking request		Interact with Vehicle Scheduling Agent	New scheduling request
New parking slot request ET parking		Interact with Vehicle Scheduling Agent	New scheduling request

6.4 Conclusion

In this chapter we have built a conceptual simulation model in which we outlined the problem situation, the modelling objectives, the model inputs and outputs, the scope and level of detail of the model content (i.e., the MAS) and identified all assumptions and simplifications. We conclude with our findings in this paragraph. First of all, the problem situation of the case study is that Rotra wants to research the possibility of using autonomous vehicles to shunt semi-trailers at a new to-be-build multimodal cross-dock. Within this research an important factor is the number of AGVs required to process all semi-trailer movements during the day, given certain arrival- and departure times. The model should therefore provide insight in the number of AGVs required, given a certain amount of semi-trailers arriving and departing during the day. This also involves assessing the impact of the number of AGVs on waiting times and the throughput and responsiveness of the system. We focus on all operations required to pick-up and drop-off semi-trailers at various locations around the cross-dock at pilot location Velp and omit the inner working of the cross-dock. The AGVs in the system are modeled as a black-box and driving from an origin to a destination takes a certain amount of time (based on the maximum forward speed) as well as rearward docking (based on the maximum rearward speed). Next to that, the coupling and uncoupling of semi-trailer with the AGVs is assumed to never fail and requires a fixed amount of time.

7 MODEL VERIFICATION AND VALIDATION

This chapter shows how the conceptual model presented in the previous chapter is verified and validated. Section 7.1 is an introduction to this chapter. Section 7.2 describes the verification process and Section 7.3 the validation process. Section 7.4 contains the conclusions.

7.1 Introduction

An important step of any simulation study is the verification and validation (V&V) process. Without this we cannot be confident about the credibility of our simulation results. Verification is the process of making sure that the conceptual model has been implemented the correct way in the computer model and validation is the process of ensuring that the model represents the real world (Robinson, 1997). We implemented our model in the discrete-event simulation software package Tecnomatix Plant Simulation. The basis is the layout of the pilot location in our case study to ensure recognition of all stakeholders and most importantly, the stakeholders at Rotra. The semi-trailers are shown as small images of the semi-trailers currently used by Rotra. We distinguish unloaded and loaded transport with a small pallet icon in the top right corner of every semi-trailer. The AGVs move over the guide-path and can hitch and unhitch semi-trailers at the appropriate locations. The docks are modeled as a transfer station which functions either as an unloading station or as a loading station, based on the job at hand. The inner area of the cross-dock is modeled as buffers per dock. These buffers contain the loads that have been unloaded. Figure 7-1 shows a screenshot of the simulation model.

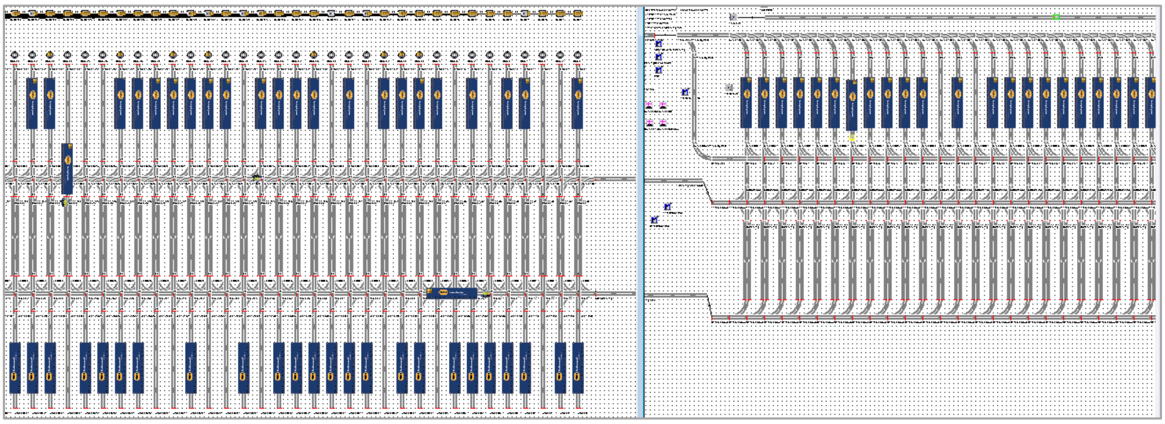


FIGURE 7-1 SCREENSHOT OF IMPLEMENTED SIMULATION MODEL

We omitted the AGV parking from the screenshot to get a clearer view of the model. In this figure we see three deployed AGVs. The AGV on the far left has just picked up a departing transport, the AGV on the bottom main road is also transporting a departing transport and the AGV on the top main road is currently on its way to pick up a semi-trailer. We see at the arrival/departure parking on the right that two parking slots have been reserved for the two departing transports currently being processed. We also see a truck unhitching a semi-trailer at the seventh parking slot at the A/D parking. The remainder of this chapter describes the verification and validation of the simulation model and provides a conclusion.

7.2 Verification

We use multiple techniques to verify the simulation model to make sure the conceptual model is translated properly to the simulation model. Law (2007) lists eight techniques which can be used to verify a simulation model. We employ some of these techniques, which in our opinion are the most suitable to our simulation model.

- **Debugging modules or subprograms**
  Large-scale simulation models such as ours are difficult to debug as a whole. We debugged our code while programming the simulation model, making sure that all components added to the model were first tested separately. After the addition of a new component (e.g. a new agent) to the simulation model, we made sure it was incorporated correctly by debugging the new component running within the entire model.
- **Running the model under a variety of settings**
  We ran the model under various settings a lot when building the model. Especially with modeling guide-path design, which consists of hundreds of track segments based on input parameters (e.g. number of docks, distance between docks and parking lengths). We changed these input parameters frequently and visually assessed whether the guide-paths were correct. We also checked whether the track segments where connected to their respective predecessors and successors such that all track segments together make up a connected network of paths. We also used this technique on all other input parameters (e.g. number of AGVs deployed, un(loading)times, number of arriving semi-trailers, number of parking slots). We concluded that the model responded correctly under a variety of settings.
- **Traces**
  Traces were used frequently to verify whether where triggered at the right moments and resulting in the correct end events. We used this for example with the assignment of parking slots and docks to semi-trailers. We checked whether the '*nearest-available rule'* was carried out properly by using traces. Also the Conflict Manager was mostly verified using this technique by checking the code when two or more AGVs were in conflict and assessing whether the Conflict Manager made the right decision on which AGV to stop.
- **Run under simplifying assumptions**
  We used some simplifying assumptions to cope with the complexity of the model to first verify and validate the model and then later extend the model with more complexity. For example, we first deployed only one AGV to verify whether the bid evaluation function of the Vehicle Scheduling agent worked correctly and verify whether the AGV took the correct routes. When we were confident enough the Vehicle Scheduling agent functioned properly, we extended the number of AGVs to a more realistic number (e.g. three). Also to test the random assignment of arriving docks to semi-trailers we first tested the model with only ten docks and twenty arriving semi-trailers. We also used some extreme cases to verify the Conflict Manager by deploying thirty AGVs to increase the odds of collisions and to check whether the agent responded appropriately in all cases.
- **Animation**
  As our simulation model heavily leans on moving units, we used the animation capability a lot to verify all whether AGVs and semi-trailers move to the right location, using the correct route and were picked-up by the right AGV at the right time. Animation was also used to visually check whether the Conflict Manager resolved any conflicts properly.

7.3 Validation

We validate our model using the techniques discussed in Robinson (1997). We try to gain as much prove as possible to show that the model is not incorrect. One of the issues of our validation process is the lack of a real world to compare our simulation model to. We based our simulation on the pilot location as discussed in Chapter 4, but this location is still non-existent. We thus do not have any real-life performance measures to validate our model with. Despite the lack of real world data, we do employ validation techniques, such as conceptual model validation, white-box validation and black-box validation.

7.3.1 Conceptual model validation

The conceptual model presented in the previous chapter is a simplified form of the real-world, including assumptions (Appendix 5), simplifications and a certain level of detail (Appendix 6). We have to make sure that all these assumptions and simplifications on which we base our simulation program are appropriate with respect to the objectives of the simulation study. According to Robinson (1997) there is no formal way of validating the conceptual model. We discussed all assumptions and simplifications in our model with our consortium partner Rotra (the problem owner). Based on their approval we validated the conceptual model.

7.3.2 White-box validation

We used white-box validation to assess the behavior of small parts of the simulation model. The functionality of every agent was first validated separately and then the interaction between the agents was validated. We used many checks and data analysis to validate all modules of the simulation. We used many tables to store data of the simulation model and checked this data per module to see whether the data corresponds to what is expected. The remainder of this paragraph shows the validation of the Demand Manager, the Parking Manager and the Vehicle Routing Agent based on the simulation outputs. The validation of the other agents is less data-driven and is validated by checking the code, using visual checks and as a whole by using black-box validation.

Demand Manager Validation

Validated using:

- Arrival times (mean inter-arrival time should be 2 minutes);
- Number of arrivals (should be equal to input data on number of arriving transports);
- Trailer ID (should be unique and in ascending order based on arrival time);
- Number of exits of a semi-trailer should be equal to the number of cargo assigned to a semi-trailer (validation of unloading process);
- The minimum content of any trailer should be 0 (empty after unloading process);
- The maximum content of any trailer should be 33 (validation of input parameter);
- The content at system exit should be equal to the content loaded into the semi-trailer (validation of loading process);
- The number of entries minus the number of exits of a semi-trailer should be equal to the content at system exit (validation of unloading and loading process together);
- The assignment of arrival- and departure docks should be random and follow a $U\{1,75\}$ distribution with an average of 38.0 and a standard deviation of 21.64.

Table 7-1 shows an excerpt of the data retrieved from the simulation model that was used to validate the above list.

TABLE 7-1 EXCERPT OF OUTPUT DATA (DEMAND MANAGER)

Run	Trailer ID	Arrival Time	Content At Exit	Minimum Content	Maximum Content	Entries	Exits
1	1	00:04.7	25	0	25	46	21
1	2	02:04.7	33	0	33	50	17
1	3	04:04.7	33	0	33	59	26
1	4	06:04.7	33	0	33	53	20
1	5	08:04.7	33	0	33	48	15
1	6	10:04.7	9	0	31	41	32

Figure 7-2 and Figure 7-3 show the assignment of docks to arriving and departing semi-trailers. It can be seen that it is a random assignment and follows a $U\{1,75\}$ distribution based on the averages and standard deviations shown in the figures.

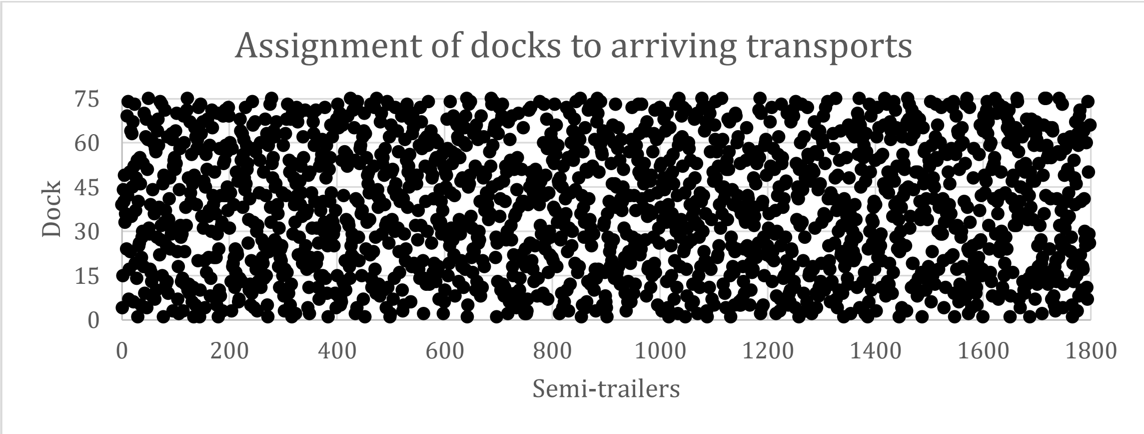


FIGURE 7-2 ASSIGNMENT OF DOCKS TO ARRIVING TRANSPORTS (μ = 38.01, σ = 21.58)

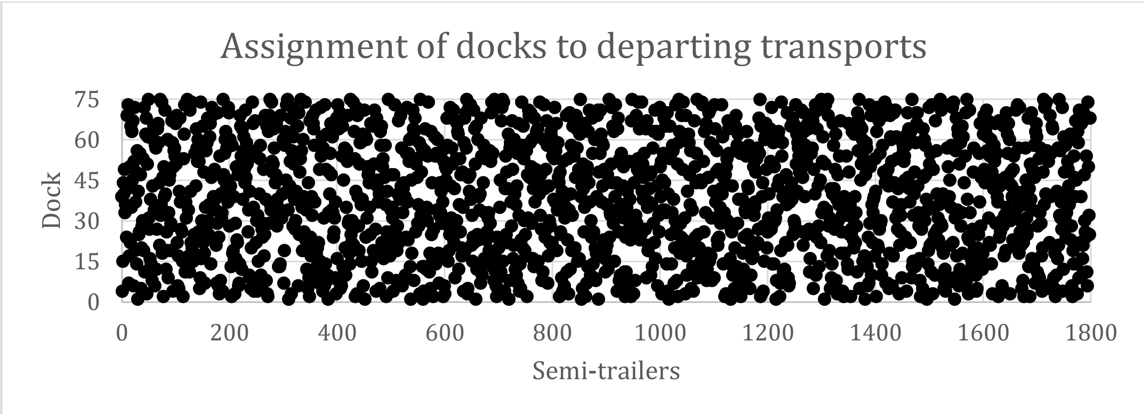


FIGURE 7-3 ASSIGNMENT OF DOCKS TO DEPARTING TRANSPORTS (μ = 37.92, σ = 21.57)

Parking Manager Validation
Validated using the:

- Assignment of the arrival parking (validation of arrival parking policy);
- Assignment of the empty trailer parking (validation of empty trailer parking policy);
- Assignment of the departure dock to empty trailer (validation of empty trailer parking policy);
- Assignment of the departure parking (validation of departure parking policy).

Figure 7-5 shows the assignment of arrival parking slots among semi-trailers (sorted on arrival time). This graph shows that at the beginning of the run most semi-trailers are parked at the beginning of the parking area (lower parking slot IDs) and this corresponds to the arrival parking policy (first-available parking slot). Halfway through the run, semi-trailers are starting to depart and thus take up space at the parking area. This results in a more occupied parking area and thus arriving transports are assigned to parking slots further away (higher parking slot ID).
The empty trailer parking policy is picking the neareast available parking slot based on the dock of a semi-trailer in the case of an arriving transport and vice versa for a departing transport. Figure 7-4 shows a scatter plot for the first case and Figure 7-7 shows a scatter plot for the latter. Examining both graphs we can safely conclude that the empty trailer parking policy is validated.
Figure 7-6 shows the assignment of parking slots to departing transports, clearly this is much more scattered than in Figure 7-5 as departing transports do not depart based on their arrival time (and thus ID) and thus a more distorted picture is obtained. However it can be seen that roughly the last fifty departing transports have a very low parking slot ID. This is because all

arriving transports have been picked-up with AGVs by then and the departing transports have left the terminal, thereby freeing up space at the parking area.

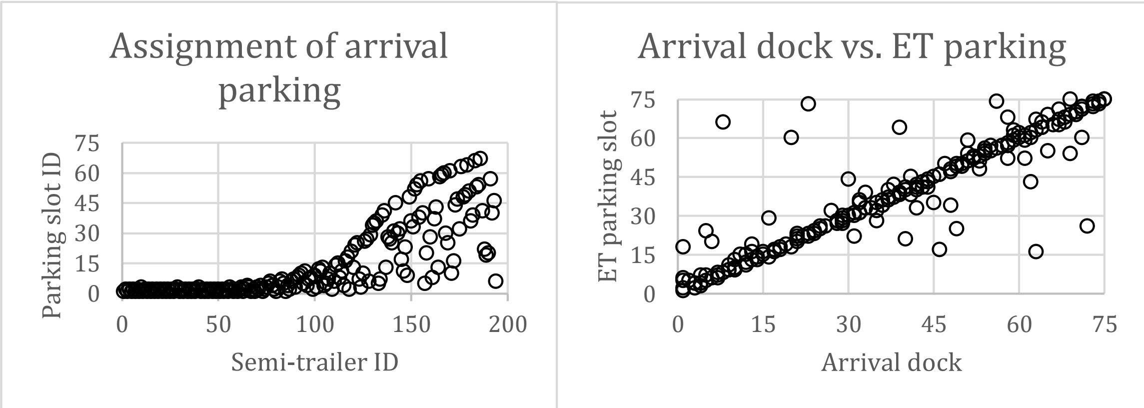


FIGURE 7-5 SCATTER PLOT OF THE ARRIVAL PARKING

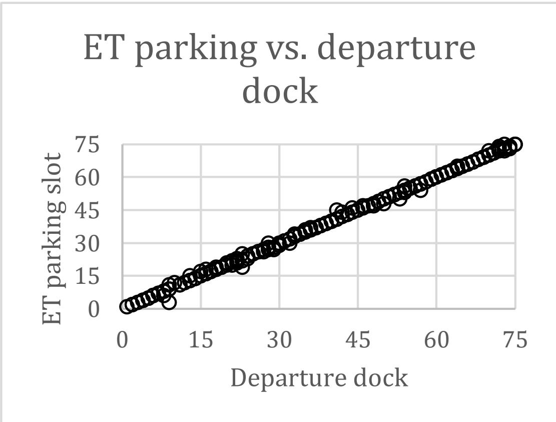


FIGURE 7-7 SCATTER PLOT OF THE EMPTY TRAILER PARKING (DEPARTURE)

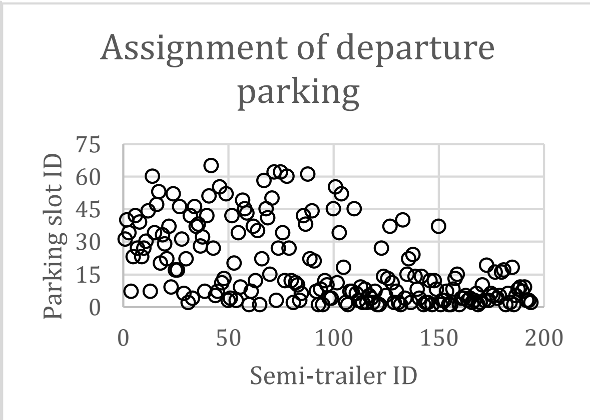


FIGURE 7-6 SCATTER PLOT OF THE DEPARTURE PARKING

Vehicle Routing Agent Validation

Validated using the:

- Travel distance between locations (validates the shortest-route policy);
- Travel time between locations (validates the shortest-route policy using the average speed of the AGV).

We obtained the travel- distance and time of all semi-trailers in a simulation run and checked whether this distance is correct, given the arrival/departure parking, arrival/departure dock and ET parking assigned to the semi-trailer, using the guide-path design as a reference point. When the ratio between the expected travel distance and the actual travel distance is 100%, we know that the routing agent has correctly calculated the path a semi-trailer should take from arrival up until departure. This is illustrated in Figure 7-8 and the corresponding comparison between the expected and actual travel time is shown in Figure 7-9.

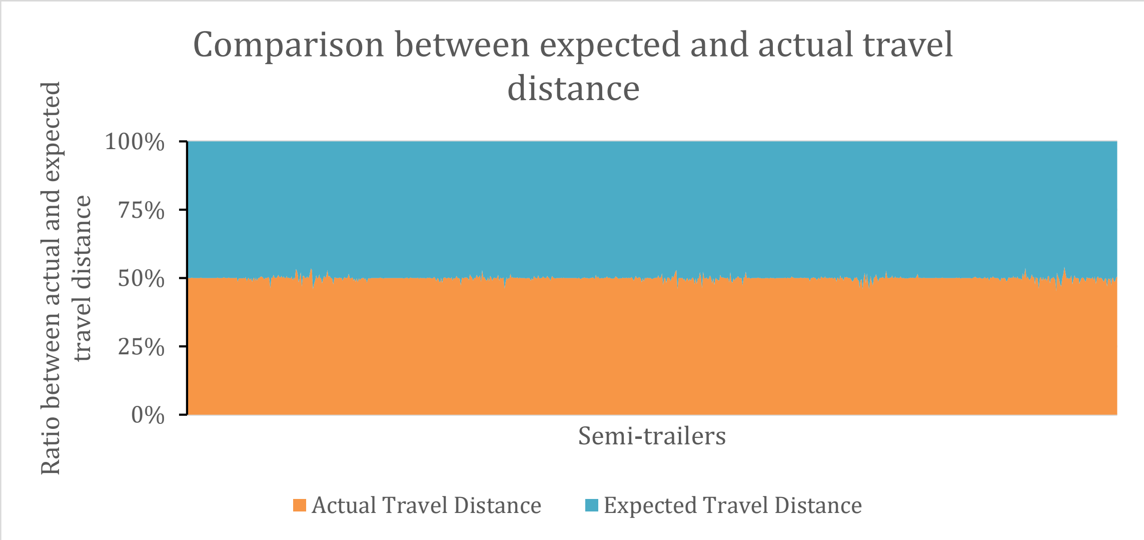


FIGURE 7-8 COMPARISON BETWEEN EXPECTED AND ACTUAL TRAVEL DISTANCE (200 SEMI-TRAILERS)

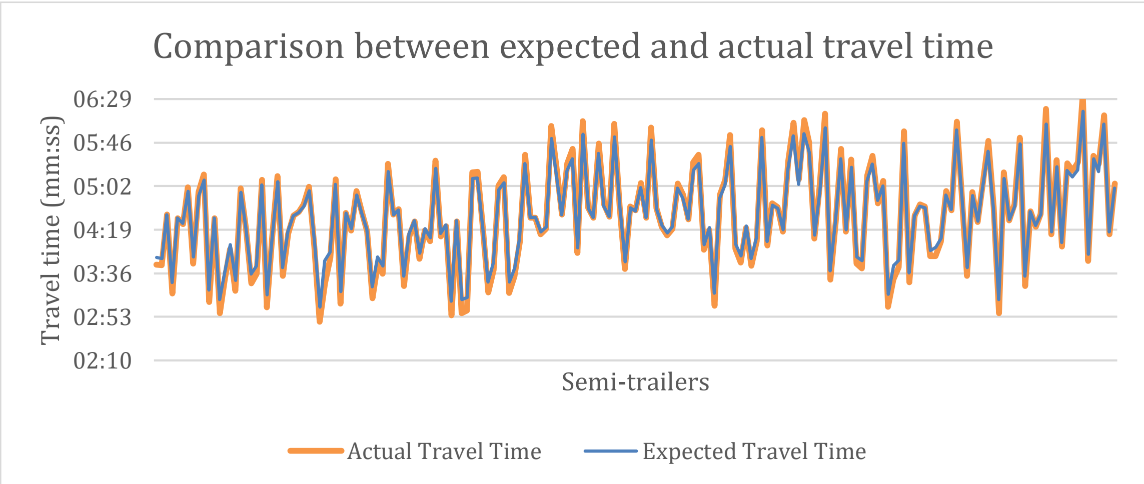


FIGURE 7-9 COMPARISON BETWEEN EXPECTED AND ACTUAL TRAVEL TIME (200 SEMI-TRAILERS)

From these figures we conclude that the routing part of the simulation performs as expected and is thereby verified.

7.3.3 Black-box validation

The final part of our validation process is the black-box validation. We use this method to validate the overall behavior of our model. We check whether given the input parameters, realistic outputs are obtained. We performed three experiments varying the number of AGVs deployed (1, 2 and 3) and for each experiment we did ten replications all with 200 arriving transports. Table 7-2 gives an overview of the simulation settings and the corresponding results (more detailed simulation outputs can be found in Appendix 9). When viewing Table 7-2 it is important to note that the number of departing transports does not have to be the same as the number of arriving transports. It could very well be that an arriving transport has 20 pallets and the same trailer leaves with 33 pallets and at some point in time there are no more pallets left for some departing trailers. This occurs more often when more AGVs are deployed and thus arriving transports are picked up earlier and the system limit is reached earlier. When this system limit is reached, departing transports will leave earlier while arriving transports have not been unloaded yet and thereby increasing the odds of a discrepancy between the number of loaded and unloaded pallets. To check whether the (small) mismatch between the number of arriving and departing transports is not due to a scheduling error, we also checked if the non-departed semi-trailers are positioned at the ET parking (i.e., the arriving semi-trailer has been unloaded, but no request has been made for a departure).

We validate the overall system by looking at the statistics of the AGVs. For example, to validate the Vehicle Scheduling and AGV Manager we look at the distribution of jobs among the AGVs. In the long run one can assume that the average pick-up and drop-off location of a semi-trailer is somewhere in the middle of the cross-dock (due to uniform distribution of assigning the arrival and departure docks). The same analogy goes for the average position of the AGV when a new job needs to be assigned to an AGV. As the Vehicle Scheduling Agent assigns the job to the AGV that can process the job the quickest, one can also assume that the distribution of jobs among AGVs should be equal. We checked this by looking at the travel distance of the deployed AGVs. Figure 7-10 shows such a comparison when three AGVs are deployed using ten runs. We conclude that there is an even distribution of jobs among the AGVs (33%) and thus job assignment is validated, as well as the interaction between the Vehicle Scheduling agent and the AGV Managers.

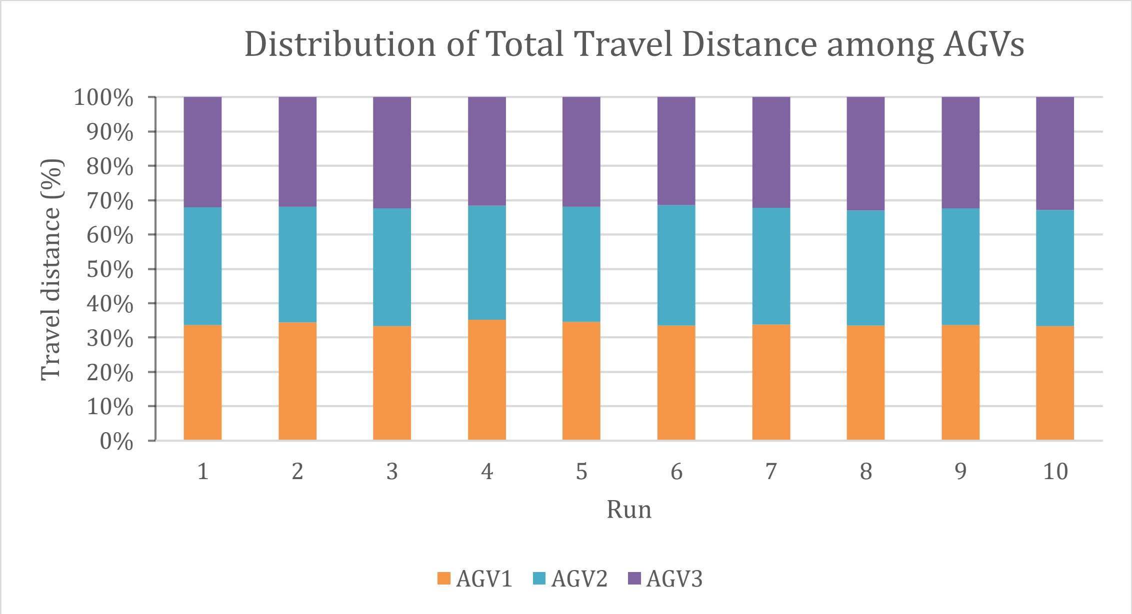


FIGURE 7-10 COMPARISON OF TRAVEL DISTANCE BETWEEN AGVS

TABLE 7-2 SUMMARY OF SIMULATION OUTPUT FOR VALIDATION

					TRAVELED DISTANCE [KM]			WORKING [%]*			WAITING [%]*			
RUN	# of AGVS	# Arriving Transports	# Departing Transports	Trailers on ET	AGV 1	AGV 2	AGV 3	AGV 1	AGV 2	AGV 3	AGV 1	AGV 2	AGV 3	Mean Life Time [hh:mm]
1	3	200	194	6	140.00	142.50	133.70	68.29	67.60	63.81	31.71	32.40	36.19	7:28
2	3	200	191	9	146.10	143.00	135.50	70.37	68.75	63.67	29.63	31.25	36.33	7:40
3	3	200	196	4	142.80	146.90	139.10	69.13	68.00	65.45	30.87	32.00	34.55	7:35
4	3	200	193	7	153.50	144.10	137.80	69.22	65.66	60.33	30.78	34.34	39.67	7:49
5	3	200	190	10	145.70	141.00	134.70	70.70	65.22	61.89	29.30	34.78	38.11	7:36
6	3	200	189	11	139.80	145.80	131.10	66.48	68.49	61.76	33.52	31.51	38.24	7:38
7	3	200	190	10	140.00	140.40	133.50	68.77	66.66	64.02	31.23	33.34	35.98	7:31
8	3	200	192	8	140.50	139.90	138.50	69.48	67.20	66.95	30.52	32.80	33.05	7:25
9	3	200	192	8	143.30	145.00	138.50	68.87	67.72	65.65	31.13	32.28	34.35	7:32
10	3	200	194	6	142.50	143.80	140.50	67.37	67.72	66.20	32.63	32.28	33.80	7:38
1	2	200	200	0	241.60	237.00		70.76	70.64		29.24	29.36		13:15
2	2	200	200	0	238.00	227.20		72.05	70.03		27.95	29.97		12:47
3	2	200	200	0	239.20	235.30		70.51	68.92		29.49	31.08		12:57
4	2	200	199	1	238.00	228.10		73.01	69.65		26.99	30.35		12:44
5	2	200	199	1	234.10	234.80		69.68	69.13		30.32	30.87		12:54
6	2	200	200	0	229.50	234.40		72.19	73.75		27.81	26.25		12:34
7	2	200	200	0	238.30	230.30		72.25	71.29		27.75	28.81		12:49
8	2	200	200	0	239.80	234.30		73.44	70.31		26.56	29.69		12:55
9	2	200	198	2	233.40	232.10		70.97	71.29		29.03	28.71		12:41
10	2	200	200	0	235.30	236.00		73.46	73.72		26.54	26.28		12:52

TABLE 7-3 SUMMARY OF SIMULATION OUTPUT FOR VALIDATION (CONTINUED)

					TRAVELED DISTANCE [KM]			WORKING [%]*			WAITING [%]*			
RUN	# of AGVS	# Arriving Transports	# Departing Transports	Trailers on ET	AGV 1	AGV 2	AGV 3	AGV 1	AGV 2	AGV 3	AGV 1	AGV 2	AGV 3	Mean Life Time [hh:mm]
1	1	200	200	0	528.20			76.72			23.28			31:05
2	1	200	200	0	522.70			76.27			23.73			30:46
3	1	200	200	0	529.00			76.50			23.50			30:55
4	1	200	200	0	528.10			76.92			23.08			30:48
5	1	200	200	0	526.40			76.12			23.88			31:03
6	1	200	200	0	530.70			75.82			24.18			31:20
7	1	200	200	0	523.30			75.36			24.64			30:43
8	1	200	200	0	535.20			77.36			22.64			31:19
9	1	200	200	0	523.20			75.50			24.50			30:52
10	1	200	200	0	525.60			76.56			23.44			31:03

* During all runs the share of working- and waiting time adds up to 100% for all AGVs. This implies that the time the AGV is stopped during the run due to congestion is negligible. Together with visual checks this is proof for a valid Conflict Manager as AGVs are not stopped or blocked longer than necessary to avoid collisions.

We also validate our model by varying the number of AGVs deployed and checking how this impacts the system performance. Figure 7-11 shows the relation between the number of AGVs deployed and the total travel distance of the AGVs. It is interesting to see that the total travel distance of all AGVs, increases when there are less AGVs in the system. This is as expected because when more AGVs are deployed, the odds increase of an AGV being near the pick-up location of a new job. This AGV is chosen by the scheduling agent as this minimizes the total travel distance. Therefore when multiple AGVs are deployed they tend to be *zoning*. One AGV could for example be doing a lot of jobs from the cross-dock to the ET parking and back while another AGV may be alternating between picking-up arriving semi-trailers and dropping-off departing semi-trailers.

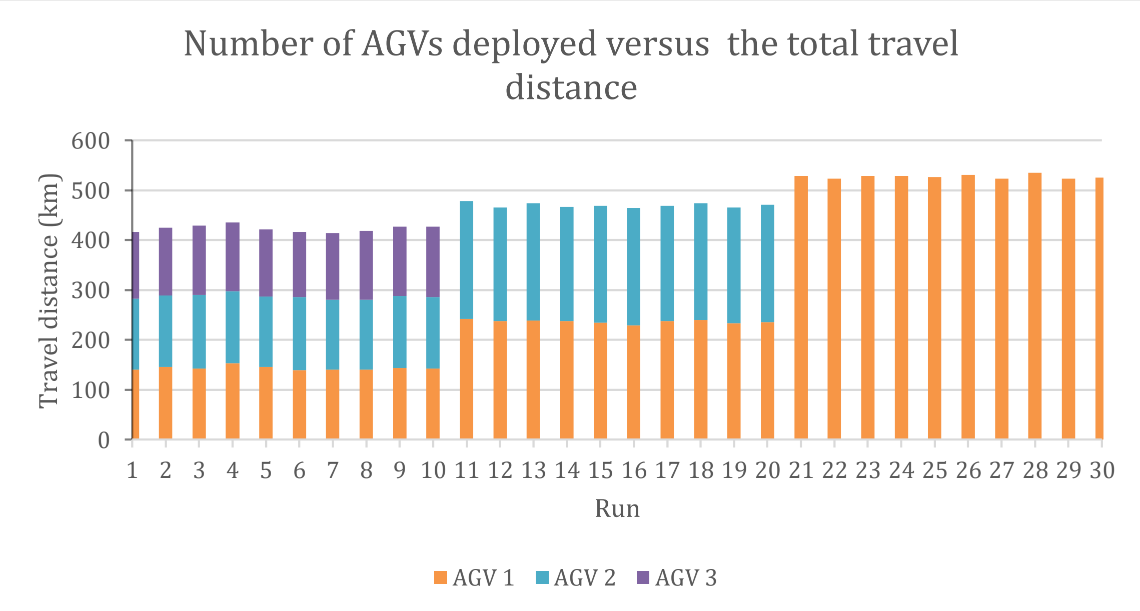
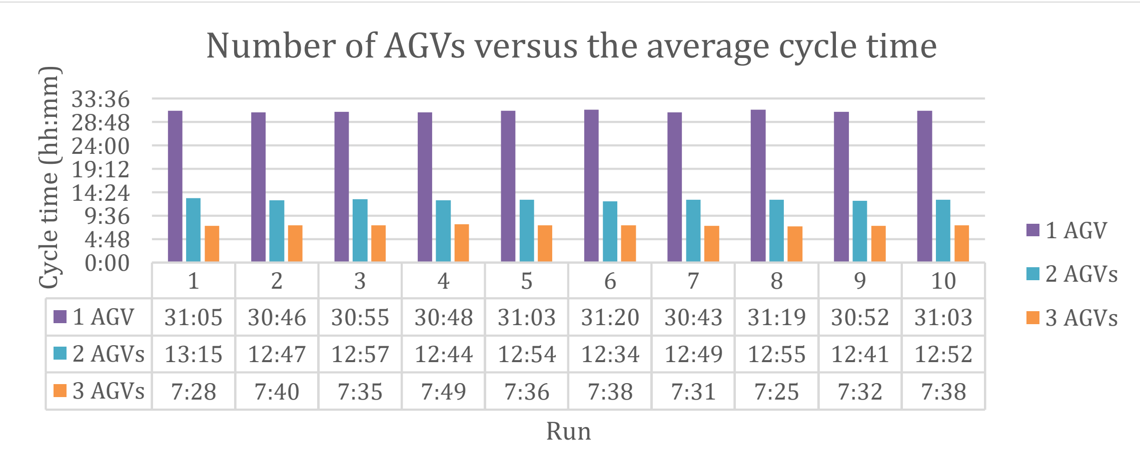


FIGURE 7-11 NUMBER OF AGVS DEPLOYED VERSUS THE TOTAL TRAVEL DISTANCE

The final validation check is the relation between the number of AGVs and the average cycle time of a semi-trailer as shown in Figure 7-12. The cycle time consists of (un)loading time, (de)coupling time, travel time and waiting time. The latter category is a large part of the total cycle time as empty trailers have to wait until they are assigned to a departing transport. A clear decrease in cycle time can be seen when the number of AGVs increases.



	1	2	3	4	5	6	7	8	9	10
1 AGV	31:05	30:46	30:55	30:48	31:03	31:20	30:43	31:19	30:52	31:03
2 AGVs	13:15	12:47	12:57	12:44	12:54	12:34	12:49	12:55	12:41	12:52
3 AGVs	7:28	7:40	7:35	7:49	7:36	7:38	7:31	7:25	7:32	7:38

FIGURE 7-12 NUMBER OF AGVS DEPLOYED VERSUS THE AVERAGE CYCLE TIME

7.4 Conclusion

By using multiple verification and validation techniques, we showed that our simulation model is a valid, credible and useful simulation model. The combination of white-box and black-box validation provides enough evidence to safely assume that our model valid. Although the model may not be 100% valid and thus may contain some small discrepancies, the extent to which we tested our model provides enough credibility. Therefore we conclude that the model resembles reality close enough to obtain realistic and useful results.

8 CONCLUSIONS AND RECOMMENDATIONS

This chapter contains the conclusions of this research (Section 8.1) and recommendations for future research (Section 8.2).

8.1 Conclusions

This thesis focused on the design of a multi-agent system for advanced material handling systems. Our goal was to develop a generic automated planning and control system based on agent technology for the pick-up and docking of semi-trailers by means of AGVs in a collision- and conflict free environment in such a way that it yields a cost-effective, near-optimal solution. We put our design of the multi-agent system to work using a case study provided by Rotra for which we build a verified and validated simulation model. Below we discuss our findings per research question.

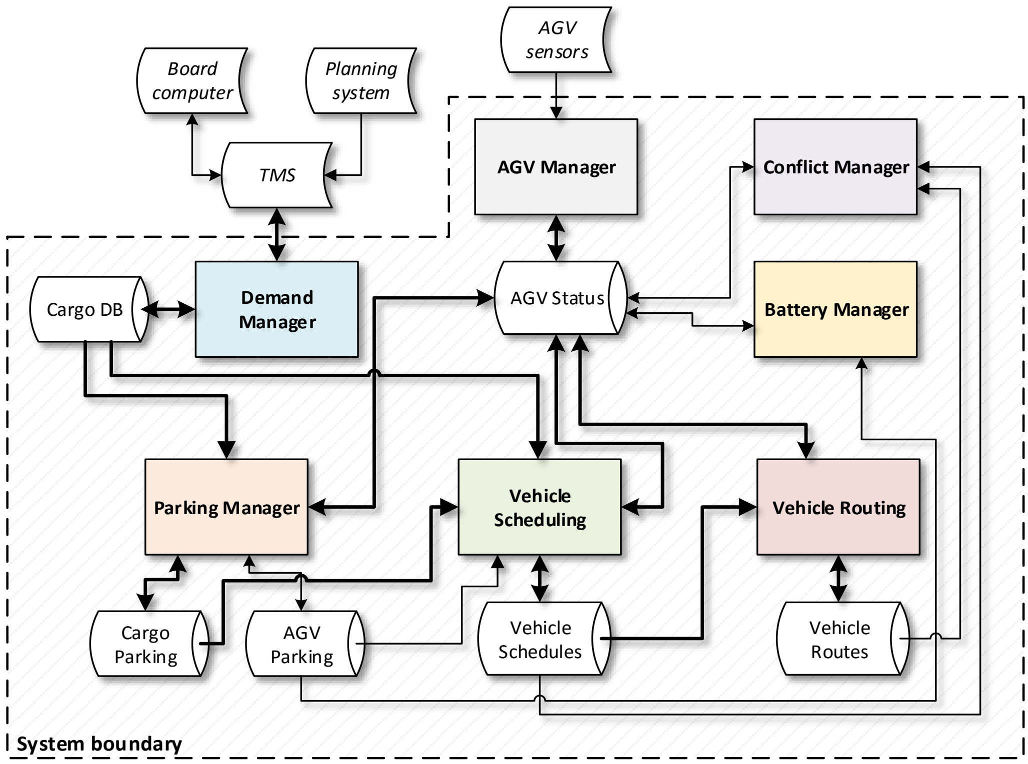
1. **How should the system be decomposed into functional specifications?**
   We analyzed the pilot location of our case study to develop a functional specification for the multi-agent system. Using the expertise of many of the consortium partners within INTRALOG, especially of Rotra, we obtained a thorough understanding of the system. We argued that the Prometheus methodology is a comprehensive, useful and easily implemented method for the design of agent-based systems. Using the system specification phase of the Prometheus methodology and the input of our consortium partners we conclude and recommend that the multi-agent system should consist of the following (generic) clusters of functionalities:

   1. **Demand Management (DM)**
      This functionality monitors inbound/outbound cargo, obtaining information about expected arrival/dispatch time and cargo description.
   2. **Park Management (PM)**
      This functionality assigns pick-up and drop-off locations to all cargo and AGVs.
   3. **Vehicle Scheduling (VS)**
      This functionality decides when, where and which AGV should pick-up and drop-off cargo or empty trailers.
   4. **Vehicle Routing (VR)**
      This functionality determines the route such that the AGV can pick-up and drop-off cargo or empty trailers
   5. **Conflict Resolution (CR)**
      This functionality monitors all AGV movements and makes sure there are no collisions and resolves conflicts.
   6. **Battery Management (BM)**
      This functionality determines when and where AGVs should be refilled or recharged.

2. **How should the agents be designed in terms of functionalities?**
   From the functionality specification we continued with an architectural design phase where the agents and their interconnectivity are defined. We assessed the validity of the agents by using data coupling diagrams and agent acquaintance diagrams. We conclude and recommend that multi-agent system should consist of the following agents:

- **Demand Manager**
  The Demand Manager agent is responsible for retrieving and logging all cargo arrival and departure data, including all external systems (e.g. TMS and on-board computers) and brings this into the MAS.
- **Parking Manager**
  The Parking Manager assigns parking slots to all arriving and departing cargo and also for the AGVs when idling. Assigning parking slots to empty trailers on the terrain itself is also part of the functionality of this agent.
- **Vehicle Scheduling Agent**
  The Vehicle Scheduling agent assigns AGVs to transportation requests.
- **Vehicle Routing Agent**
  The Vehicle Routing agent determines the routing for all AGVs.
- **Battery Manager**
  The Battery Manager is responsible for effective charging schedules for all AGVs.
- **Conflict Manager**
  The Conflict Manager resolves all possible conflicts between AGVs and maintains a conflict-free environment by making stop-and-go decisions.
- **AGV Manager**
  The AGV Manager processes all data to and from the AGV controller and thus maintains the AGV status during system operation.

3. **How should the agents interact with each other and the environment?**
   Using scenario analysis we obtained insight into which states the system and the environment can be and how agents should respond in these various scenarios. From this analysis we concluded that the interaction between the agents and the environment can be summarized with the diagram below.

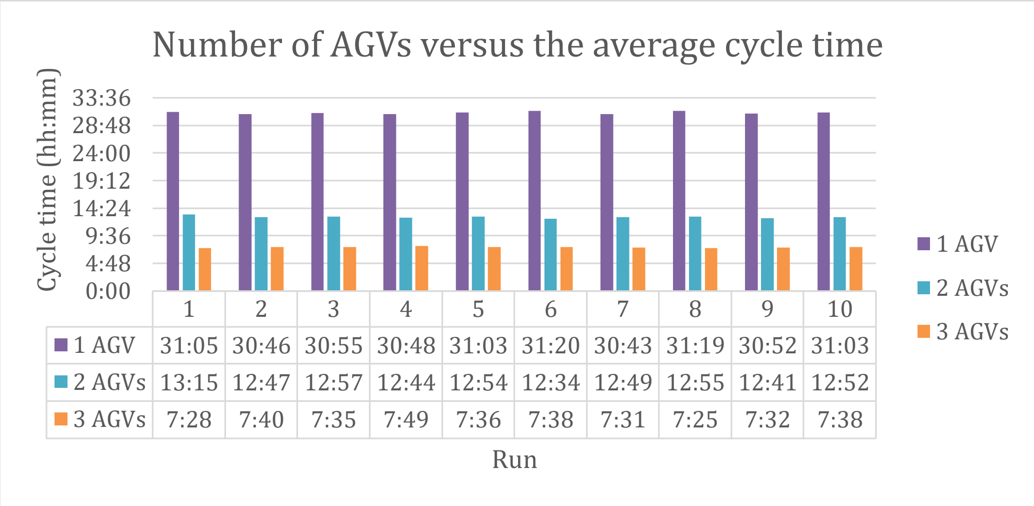


4. **Which decision capabilities should the different types of agents have?**
We finished our multi-agent system design with a detailed design phase as part of the Prometheus methodology. This phase resulted in capability overviews of all agents. We used these capabilities to develop intelligent agents using an AGV framework. From this translation we conclude with the following capabilities per agent:

- **Demand Manager**
  This agent does not require complex algorithms, but instead provides the link between external systems (e.g. TMS) and the MAS. It is however of utmost importance that all data required is readily available and up-to-date.
- **Parking Manager**
  We use a *nearest-available* rule for all parking areas. Arriving and departing semi-trailers are assigned to the first available (i.e. not occupied) parking slot. This results in an ordered priority list where parking slot 1 is more favorable to parking slot 2 and so forth.
- **Vehicle Scheduling Agent**
  A very common way of scheduling within MASs is using an auction mechanism where AGVs compete for orders. The Vehicle Scheduling agent initiates a proposal when a new job enters the system via the Demand Manager (e.g., pick-up at cross-dock). All AGV agents evaluate this proposal and send back a bid. Based on a bid evaluation function the winner is announced. We use the auction mechanism as described in Mes, van der Heijden, & van Harten (2007).
- **Vehicle Routing Agent**
  The routing agent determines which route the AGV should take given the pick-up and drop-off location of the job. We use the shortest-path method and determine *a priori* the shortest paths between all nodes using the guide-path defined above.
- **Battery Manager**
  The Battery Manager uses an opportunity charging strategy as defined by McHaney (1995). Whenever an AGV is idling it is send to the AGV parking. The Battery Manager also monitors the battery level of all AGVs and asks the Vehicle Scheduling agent to schedule a charging job whenever the battery is below a certain threshold.
- **Conflict Manager**
  This agent is responsible to avoid collisions, congestion and deadlocks. We use a priority list to make stop-and-go decisions based on the current status of the AGV. When two or more AGVs want to use the same arc at the same time, the Conflict Manager evaluates which AGV has priority based on the current jobs the AGVs are processing and stops the AGV with the lowest priority.
- **AGV Manager**
  The AGV Manager has an important role within the MAS as it feeds information about the AGV to the system (e.g. current position and battery level). It furthermore uses the bid calculating function as described above to respond to the proposals of the Vehicle Scheduling agent.

5. **How to build a valid simulation model for the multi-agent system?**
We have shown how to build a conceptual model based on the multi-agent system with a sufficient level of detail. In this conceptual model we have shown how all agents respond to different events and how these events are interconnected by using flowcharts. We implemented our model in the discrete-event simulation software package Tecnomatix Plant Simulation. Using multiple verification and validation techniques we conclude that the implemented simulation model is a flexible, useful model and an accurate representation of the real world. One of the used validation techniques was assessing the impact of the number of AGVs on the cycle time of the semi-trailers. From the figure below we conclude

that the deploying more AGVs results in a significant decrease in the cycle time and therefore recommend that a minimum of three AGVs are required for an acceptable cycle time at our case study.



8.2 Recommendations for future research

Although this research presented a generic design of a multi-agent system for the planning and control of autonomous vehicles, we have made multiple assumptions along the way and have had a limited scope. We therefore recommend and challenge future researchers to extend this research by focusing on the following theoretical and practical topics:

Theoretical

1. Having built a valid simulation model, the experimental design of the simulation model still needs to be done. The analysis of the output data should provide answers to the number of AGVs needed and assess the impact on Key Performance Indicators (KPIs) for the system, such as costs, utilization, throughput time, travel time and waiting time using various scenarios.
2. We have focused on the cross dock of the pilot location in our case study. This pilot location will also feature a container terminal and the integration of the container movements with the semi-trailer movements is an interesting and promising extension of this research.
3. The MAS design should be applied to more case studies to validate the genericity of the MAS.
4. More research is required on which vehicle guidance and orientation system is suitable for our automated Material Handling System.
5. Extension of the simulation model by assessing the impact of different vehicle parking strategies on system performance.
6. Extension of the simulation model by assessing impact of different vehicle scheduling strategies on system performance.
7. Extension of the simulation model by assessing impact of different battery charging alternatives on system performance.
8. Extension of the simulation model by assessing impact of different vehicle routing strategies on system performance.
9. Extension of the simulation model by incorporating uncertainty of arrival times, driving times and handling times.
10. Extension of the simulation model by incorporating multi-stage bidding of the vehicle scheduling agent.

Practical

1. The design of an autonomous coupling device between the AGV and semi-trailer.
2. The design of an autonomous battery charging system.
3. The design of a dock door and corresponding apron space to facilitate automatic docking.
4. The design of an IT infrastructure for the MAS and the interfaces between external systems.
5. Further research is required on how to mature the MAS from conceptual/simulation to pilot testing/implementation, including the identification of all practical requirements.

BIBLIOGRAPHY

Autonieuws. (2015). Nederland wil voortrekkersrol zelfrijdende auto. Retrieved July 11, 2015, from http://www.autoweek.nl/nieuws/31844/nederland-wil-voortrekkersrol-zelfrijdende-auto

Becker, M., Syyed, G., Wenning, B.-L., & Görg, C. (2007). On Mobile Agents for Autonomous Logistics : An Analysis of Mobile Agents considering the Fan Out and sundry Strategies Markus Becker , Bernd-Ludwig Wenning and Carmelita G ¨ org. *Int. J. Accounting, Auditing and Performance Evaluation*, *I*(1). http://doi.org/10.1504/IJSOI.2007.014515

Breton, L., Maza, S., & Castagna, P. (2006). A multi-agent based conflict-free routing approach of bi-directional automated guided vehicles. *2006 American Control Conference*, *00*(33), 2825–2830. http://doi.org/10.1109/ACC.2006.1656652

Egbelu, P. J. (1987). The Use Of Non-Simulation Approaches In Estimating Vehicle Requirements in an automated Guided Vehicle Based Transport System.

Erol, R., Sahin, C., Baykasoglu, A., & Kaplanoglu, V. (2012). A multi-agent based approach to dynamic scheduling of machines and automated guided vehicles in manufacturing systems. *Applied Soft Computing*. http://doi.org/10.1016/j.asoc.2012.02.001

Farahvash, P., & Boucher, T. O. (2004). A multi-agent architecture for control of AGV systems. *Robotics and Computer-Integrated Manufacturing*, *20*(6 SPEC. ISS.), 473–483. http://doi.org/10.1016/j.rcim.2004.07.005

FIPA Contract Net Interaction Protocol Specification. (2002). *Foundation for Intelligent Physical Agents*, (SC00029H), 9. Retrieved from http://www.fipa.org/specs/fipa00029/SC00029H.pdf

Gademann, a. J. R. M., & van de Velde, S. L. (2000). Positioning automated guided vehicles in a loop layout. *European Journal of Operational Research*, *127*(3), 565–573. http://doi.org/10.1016/S0377-2217(99)00341-0

Gath, M., Herzog, O., & Edelkamp, S. (2014). Autonomous and flexible multiagent systems enhance transport logistics. *2014 11th International Conference and Expo on Emerging Technologies for a Smarter World, CEWIT 2014*, 0–5. http://doi.org/10.1109/CEWIT.2014.7021143

Hafidz, M., Lin, H., & Murata, T. (2010). Dynamic Task Assignment of Autonomous AGV System Based on Multi Agent Architecture. *Progress in Informatics and Computing (PIC)*, 1151–1156. http://doi.org/10.1109/PIC.2010.5687984

HTSM Automotive Roadmap 2016-2020. (2015). Retrieved July 11, 2015, from http://www.automotivenl.com/nl/databases/nieuws/automotivenl/item/6635-automotive-roadmap

Kim, K. H. (1995). Positioning of automated guided vehicles in a loop layout to minimize the mean vehicle response time. *INT. J. Production Economics*, *39*, 201–214. Retrieved from http://www.sciencedirect.com/science/article/pii/092552739500022G

Kindt, M. (2011). Longer and Heavier Vehicles in practice. *Directorate General for Public Works and Water Management (Rijkswaterstaat) – Traffic and Shipping Department*, 112.

Kural, K., Prati, A., Besselink, I., Pauwelussen, J., & Nijmeijer, H. (2013). Validation of Longer and Heavier Vehicle Combination Simulation Models. *SAE International Journal of Commercial Vehicles*, *6*(2), 340–352. http://doi.org/10.4271/2013-01-2369

Ladier, A.-L., & Alpan, G. (2015). Cross-docking operations: Current research versus industry practice. *Omega*. http://doi.org/10.1016/j.omega.2015.09.006

Law, A. M. (2007). *Simulation modeling and analysis*. *McGraw-Hill series in industrial engineering and management science*.

Le-Anh, T. (2005). *Intelligent Control of Vehicle-Based Internal Transport Systems*. Retrieved from http://repub.eur.nl/res/pub/6554/

Le-Anh, T., & De Koster, M. B. M. (2006). A review of design and control of automated guided vehicle systems. *European Journal of Operational Research*, *171*(1), 1–23. http://doi.org/10.1016/j.ejor.2005.01.036

McHaney, R. (1995). Modelling battery constraints in discrete event automated guided vehicle simulations. *International Journal of Production Research*, *33*(11), 3023–3040. http://doi.org/10.1080/00207549508904859

Mes, M., van der Heijden, M., & van Harten, A. (2007). Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems. *European Journal of Operational Research*, *181*(1), 59–75. http://doi.org/10.1016/j.ejor.2006.02.051

Michalek, M., & Kielczewski, M. (2013). Helping a driver in backward docking with N-trailer vehicles by the passive control-assistance system. In *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC* (pp. 1993–1999). IEEE. http://doi.org/10.1109/ITSC.2013.6728522

Oboth, C., Batta, R., & Karwan, M. (1999). Dynamic conflict-free routing of automated guided vehicles. *International Journal of Production Research*.

Oliveira, E., Fischer, K., & Stepankova, O. (1999). Multi-agent systems: Which research for which applications. *Robotics and Autonomous Systems*, *27*(1), 91–106. http://doi.org/10.1016/S0921-8890(98)00085-2

Padgham, L., & Winikoff, M. (2005). *Developing intelligent agent systems: A practical guide* (Vol. 13).

Parunak, H. V. D. (1998). Practical and Industrial Applications of Agent-Based Systems. *Technology*, 1–41.

Qiu, L., & Hsu, W.-J. (2001). A bi-directional path layout for conflict-free routing of AGVs. *International Journal of Production Research*, *39*(10), 2177–2195. http://doi.org/10.1080/00207540110038531

Qiu, L., Hsu, W.-J., & Huang, S.-Y. (2002). Scheduling and routing algorithms for AGVs: A survey. *International Journal of Production Research*, *40*(3), 745–760. http://doi.org/10.1080/00207540110091712

Robinson, S. (1997). Simulation Model Verification And Validation: Increasing The Users' Confidence. *Winter Simulation Conference Proceedings,* 7–13. http://doi.org/10.1109/WSC.1997.640377

Robinson, S. (2008a). Conceptual modelling for simulation Part I: definition and requirements. *Journal of the Operational Research Society*, *59*(3), 278–290. http://doi.org/10.1057/palgrave.jors.2602368

Robinson, S. (2008b). Conceptual modelling for simulation Part II: a framework for conceptual modelling. *Journal of the Operational Research Society*, *59*(3), 291–304. http://doi.org/10.1057/palgrave.jors.2602368

Singh, N., Sarngadharan, P. V., & Pal, P. K. (2011). AGV scheduling for automated material distribution: A case study. *Journal of Intelligent Manufacturing*, *22*(2), 219–228. http://doi.org/10.1007/s10845-009-0283-9

Van der Meer, J. R. (2000). *Operational Control of Internal Transport*. *Ph.D. Series Research in Management* (Vol. 1).

Ventura, J. a., & Rieksts, B. Q. (2009). Optimal location of dwell points in a single loop AGV system with time restrictions on vehicle availability. *European Journal of Operational Research*, *192*(1), 93–104. http://doi.org/10.1016/j.ejor.2007.09.014

Vis, I. F. A. (2006). Survey of research in the design and control of automated guided vehicle systems. *European Journal of Operational Research*, *170*(3), 677–709. http://doi.org/10.1016/j.ejor.2004.09.020

Wallace, A. (2007). Multi-agent negotiation strategies utilizing heuristics for the flow of AGVs. *International Journal of Production Research*, *45*(2), 309–322. http://doi.org/10.1080/00207540600726752

Winikoff, M., & Padgham, L. (2004). The prometheus methodology. In *Methodologies and Software Engineering For Agent Systems* (11th ed., pp. 217–234). Springer US. Retrieved from http://link.springer.com/content/pdf/10.1007/1-4020-8058-1_14.pdf

Wooldridge, M. (2009). *An Introduction to MultiAgent Systems* (Second Edi). John Wiley & Sons Ltd.

Xin, J., Negenborn, R. R., & Lodewijks, G. (2014). Trajectory planning for AGVs in automated container terminals using avoidance constraints: A case study. In *IFAC Proceedings Volumes (IFAC-PapersOnline)* (Vol. 19, pp. 9828–9833). IFAC Secretariat.

Xing, W., Peihuang, L., Qixiang, C., Chidong, Z., Ke, S., & Chen, J. (2012). Design and Control of Material Transport System for Automated Guided Vehicle. *UKACC Internation Conference on Control*, (September), 3–5.

APPENDICES

1 SYSTEM FUNCTIONALITIES

Demand Management Functionality	
Description	This functionality monitors inbound/outbound cargo, obtaining information about expected arrival/dispatch time and cargo description
Goals	Obtain expected arrival time, Obtain, departure time, Obtain cargo information
Actions	Log arrival/departure and cargo information
Triggers	Cargo arrival, Cargo departure
Information used	Cargo arrival time, Cargo dispatch time, cargo description
Information provided	Arrived cargo, dispatched cargo, cargo description, delayed cargo

Park Management Functionality	
Description	This functionality assigns pick-up and drop-off locations to all cargo and AGVs
Goals	Parking slot allocation
Actions	Assign parking slot
Triggers	Cargo arrival, Cargo dispatch, AGV idle
Information used	Arrival database, Dispatch database, AGV status, AGV parking database, Cargo parking database
Information provided	AGV parking, Cargo parking

Vehicle Scheduling Functionality	
Description	This functionality decides when, where and which AGV should pick-up or drop-off cargo
Goals	Pick-up schedule, Drop-off schedule
Actions	Determine optimal schedule(s)
Triggers	Cargo arrival, Cargo dispatch
Information used	Arrival database, Dispatch database, AGV status
Information provided	AGV schedule

Vehicle Routing Functionality	
Description	This functionality determines the route an AGV should take
Goals	AGV routing
Actions	Determine optimal routes
Triggers	AGV scheduled for pick-up or drop-off
Information used	AGV Status, Guide-path design. AGV parking, Cargo parking
Information provided	AGV route

Conflict Resolution Functionality	
Description	This functionality monitors all AGV movements and makes sure there are no collisions and resolves conflicts
Goals	Collision- and conflict free routing
Actions	Resolve conflict
Triggers	(Conflicting) AGV routing, Collision detection
Information used	AGV routing, Collision sensors, Priority rules. AGV status
Information provided	Stop and go decisions

Battery Management Functionality	
Description	This functionality determines when and where AGVs should be refilled or recharged
Goals	Maximizing AGV availability
Actions	Determine refueling/recharging schedule
Triggers	Low power status
Information used	AGV Status
Information provided	Recharge schedule, AGV routing (from VR)

2 SCENARIO DEVELOPMENT

2.1 Cargo departure

Scenario: *Cargo departure*

Key for functionality and data abbreviations

DM	Demand Management
PM	Parking Management
VS	Vehicle Scheduling
VR	Vehicle Routing
CR	Conflict Resolution
BM	Battery Management
AG	AGV Controller
Car. Arr. T.	Cargo Arrival Time
Car. Prop.	Cargo Properties
Car. Park	Cargo Parking Slot
Park. Avail.	Parking Availability
AGV Stat.	AGV Status
AGV Sched.	AGV Schedule
G.P. Des.	Guide-Path Design

	Step type	**Step**	**Functionality**	**Data used and produced**
1	Percept:	*New outbound cargo*		
2	Goal:	Obtain arrival time	DM	Car. Arr. T.
3	Goal:	Obtain properties	DM	Car. Prop.
4	Action:	Request parking slot	PM	Car. Arr. T. Car. Prop.
5	Goal:	Assign parking slot	PM	Car. Arr. T. Car. Prop. Park. Avail. Car. Park. Park. Avail.
6	Goal:	Determine parking arrival time	PM	Car. Arr. T. G.P. Des. Car. Arr. T.
7	Action:	Request pick-up	VS	Car. Arr. T. Car. Prop. Car. Park.
8	Goal:	Determine AGV schedule	VS	Car. Park. AGV Stat. AGV Sched.
9	Action:	Request route	VR	AGV Sched.
10	Goal:	Determine AGV Route	VR	G.P. Des. AGV Sched. AGV Stat. AGV Route
11	Goal:	Update AGV Status	CR	AGV Stat. AGV Stat.
12	Action:	Send info to AGV Controller	VR	AGV Sched. AGV Route
13	Other:	Wait for AGV to be done		

14	Scenario:	*AGV conflict*		AGV Routing AGV Stat.
15	Percept:	*AGV is done*		
16	Goal:	Update AGV Status	CR	AGV Stat. <u>AGV Stat.</u>
17	Goal:	Update Cargo Properties	DM	Car. Prop. <u>Car. Prop.</u>

2.2 AGV idle

Scenario: *AGV idle*

Key for functionality and data abbreviations

DM	Demand Management
PM	Parking Management
VS	Vehicle Scheduling
VR	Vehicle Routing
CR	Conflict Resolution
BM	Battery Management
AG	AGV Controller
AGV Park.Avail.	AGV Parking Availability
AGV. Park	AGV Parking Slot
AGV Stat.	AGV Status
AGV Sched.	AGV Schedule
G.P. Des.	Guide-Path Design

	Step type	**Step**	**Functionality**	**Data used and <u>produced</u>**
1	Percept:	*AGV status is idle*		
2	Action:	Request parking slot	PM	AGV Stat.
3	Goal:	Assign parking slot	PM	AGV Stat. AGV Sched. AGV Park.Avail <u>AGV. Park.</u> <u>AGV Park.Avail</u>
4	Goal:	Determine parking arrival time	PM	AGV Stat. G.P. Des.
5	Action:	Request route	VR	AGV Sched.
6	Goal:	Update AGV Schedule	VS	AGV Sched <u>AGV Sched.</u>
7	Goal:	Determine AGV Route	VR	G.P. Des. AGV Sched. AGV Stat. <u>AGV Route</u>
8	Action:	Send info to AGV Controller	VR	AGV Sched. AGV Route
9	Goal:	Update AGV Status	CR	AGV Stat. <u>AGV Stat.</u>
10	Other:	Wait for AGV to be done		
11	Scenario:	*AGV conflict*		AGV Routing AGV Stat.
12	Percept:	*AGV is done*		
13	Goal:	Update AGV Status	CR	AGV Stat. <u>AGV Stat.</u>

2.3 AGV low battery

Scenario: *AGV low battery*

Key for functionality and data abbreviations

DM	Demand Management
PM	Parking Management
VS	Vehicle Scheduling
VR	Vehicle Routing
CR	Conflict Resolution
BM	Battery Management
AG	AGV Controller
Car. Arr. T.	Cargo Arrival Time
Car. Prop.	Cargo Properties
Car. Park	Cargo Parking Slot
AGV Stat.	AGV Status
AGV Sched.	AGV Schedule
G.P. Des.	Guide-Path Design
Char. Avail.	Charger Availability
Char. Slot	Charging slot

	Step type	**Step**	**Functionality**	**Data used and <u>produced</u>**
1	Percept:	*AGV low battery*		
2	Action:	Request charging	BM	AGV Stat. AGV Sched.
3	Goal:	Assign charging station	BM	AGV Stat. AGV Sched. Char. Avail. <u>Char. Slot.</u> <u>Char. Avail.</u>
4	Goal:	Determine charger arrival time	BM	AGV Stat. G.P. Des.
5	Goal:	Update AGV Schedule	VS	AGV Sched <u>AGV Sched.</u>
6	Action:	Request route	VR	AGV Sched.
7	Goal:	Determine AGV Route	VR	G.P. Des. AGV Sched. AGV Stat. <u>AGV Route</u>
8	Action:	Send info to AGV Controller	VR	AGV Sched. AGV Route
9	Goal:	Update AGV Status	CR	AGV Stat. <u>AGV Stat.</u>
10	Other:	Wait for AGV to be done		
11	Scenario:	*AGV conflict*		AGV Routing AGV Stat.
12	Percept:	*AGV is done*		
13	Goal:	Update AGV Status	CR	AGV Stat. <u>AGV Stat.</u>
14	Goal:	Update Cargo Properties	DM	Car. Prop. <u>Car. Prop.</u>

2.4 AGV conflict

Scenario: *AGV conflict*

Key for functionality and data abbreviations

DM	Demand Management
PM	Parking Management
VS	Vehicle Scheduling
VR	Vehicle Routing
CR	Conflict Resolution
BM	Battery Management
AG	AGV Controller
Car. Arr. T.	Cargo Arrival Time
Car. Prop.	Cargo Properties
Car. Park	Cargo Parking Slot
AGV Stat.	AGV Status
AGV Sched.	AGV Schedule
G.P. Des.	Guide-Path Design
Prio. Rul.	Priority Rules

	Step type	**Step**	**Functionality**	**Data used and <u>produced</u>**
1	Percept:	*AGV conflict*		
2	Goal:	Solve conflict	CR	AGV Stat. AGV Route AGV Sched. G.P. Des. Prio. Rul. <u>AGV Stat.</u>
3	Action:	Stop/go decision	CR	AGV Stat. <u>AGV Stat.</u>
4	Action:	Send info to AGV Controller	CR	AGV Stat.

3 AGENT DESCRIPTORS

3.1 Parking manager

Agent descriptor: Parking Manager	
Name:	Parking Manager
Description:	Assigns parking slots to cargo and AGVs
Cardinality:	One per system
Lifetime:	Ongoing
Initialization:	Load parking layouts and initial occupation
Demise:	Close open DB connections, Park AGVs
Functionalities included:	Park cargo, park AGVs
Uses data:	Cargo DB, Cargo parking DB, AGV parking DB, Travel Times DB, AGV Status
Produces data:	Cargo Parking DB, AGV Parking DB, AGV Status
Goals:	Assign parking slot, determine parking arrival time
Percepts responded to:	New entry Cargo DB, Idle AGV
Actions:	Assign parking slot
Protocols and interactions:	Request pick-up with Vehicle Scheduling, Update AGV status with AGV Manager, Request parking slot with AGV Manager

3.2 Vehicle scheduling

Agent descriptor: Vehicle Scheduling	
Name:	Vehicle Scheduling
Description:	Determines schedules for AGVs
Cardinality:	One per system
Lifetime:	Ongoing
Initialization:	Initialize empty AGV schedules
Demise:	Clear all AGV schedules
Functionalities included:	AGV Scheduling
Uses data:	Cargo DB, Vehicle Schedules DB, AGV Parking DB, Cargo Parking DB, AGV Status
Produces data:	Vehicle Schedules DB, AGV Status
Goals:	Task selection, Vehicle dispatching
Percepts responded to:	New entry Cargo Parking DB, New entry AGV Parking DB
Actions:	Determine AGV schedule
Protocols and interactions:	Request route with Vehicle Routing, Update AGV status with AGV Manager, Update schedule with Battery Manager & Parking Manager

3.3 Vehicle routing

Agent descriptor: Vehicle Routing	
Name:	Vehicle Routing
Description:	Assigns routes to AGVs
Cardinality:	One per system
Lifetime:	Ongoing
Initialization:	Load guide-paths
Demise:	Clear all routes
Functionalities included:	AGV Routing
Uses data:	Vehicle Routes DB, Vehicle Schedules DB, AGV Status
Produces data:	Vehicle Routes DB, AGV Status
Goals:	Determine AGV route
Percepts responded to:	Requests for route
Actions:	Determine route
Protocols and interactions:	Request route with Parking Manager, Vehicle Scheduling & Battery Manager, Update AGV Status with AGV Manager

3.4 Battery manager

Agent descriptor: Battery Manager	
Name:	Battery Manager
Description:	Manages AGV recharging policy
Cardinality:	One per system
Lifetime:	Ongoing
Initialization:	Load AGV battery status, Load battery station occupation
Demise:	
Functionalities included:	Maximizing AGV availability
Uses data:	AGV Parking DB, AGV Status
Produces data:	AGV Status
Goals:	Battery Management
Percepts responded to:	AGV status low battery
Actions:	Request charging
Protocols and interactions:	Update AGV Schedule with Vehicle Scheduling, Update AGV Parking with Parking Manager, Update AGV Status with AGV Manager

3.5 Conflict manager

Agent descriptor: Conflict Manager	
Name:	Conflict Manager
Description:	Resolves conflicts during AGV operation
Cardinality:	One per system
Lifetime:	Ongoing
Initialization:	
Demise:	
Functionalities included:	Collision- and conflict free environment
Uses data:	Vehicle Schedules DB, Vehicle Routes DB
Produces data:	AGV Status
Goals:	Collision free routing
Percepts responded to:	AGV status conflict
Actions:	Resolve conflict, stop-go decisions
Protocols and interactions:	Update AGV status with AGV Manager

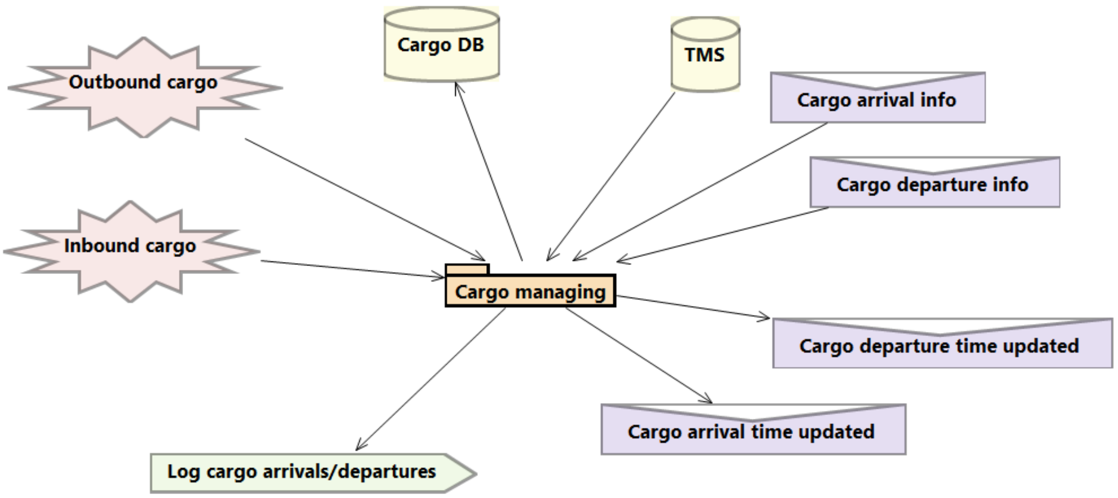
3.6 AGV manager

Agent descriptor: AGV Manager	
Name:	AGV Manager
Description:	Passes AGV status information to MAS
Cardinality:	One per AGV
Lifetime:	Ongoing
Initialization:	Initialize AGV status
Demise:	Clear AGV status
Functionalities included:	AGV Controller (external to system)
Uses data:	AGV Sensors (external to system)
Produces data:	AGV Status
Goals:	Vehicle movement
Percepts responded to:	AGV sensors, AGV status update
Actions:	Control AGV (external to system), Update AGV Status
Protocols and interactions:	Update AGV status with Parking Manager, Vehicle Scheduling, Vehicle Routing, Battery Manager and Conflict Manager

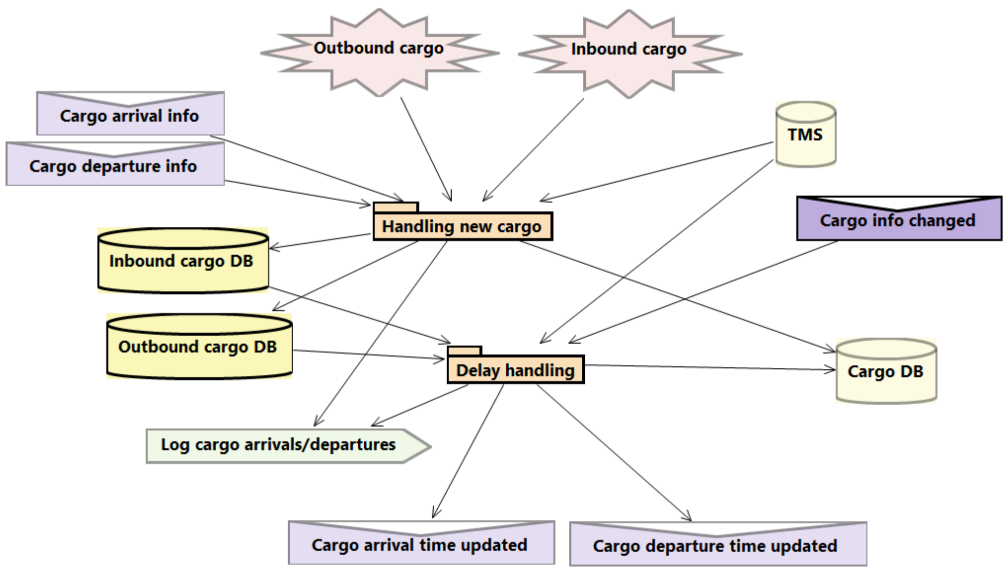
4 AGENT OVERVIEWS

4.1 Demand Manager

Agent Overview

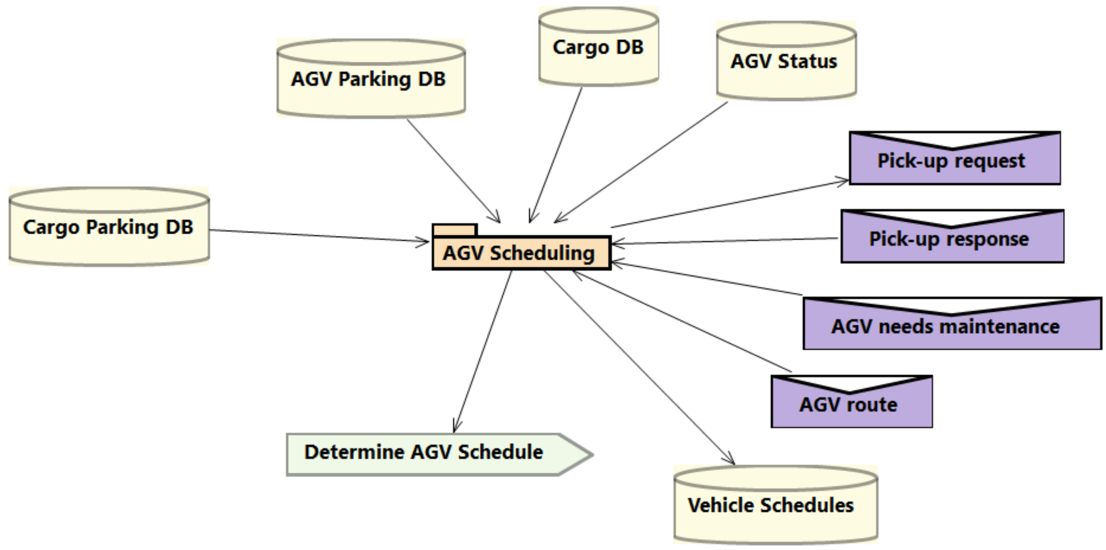


Capability Overview

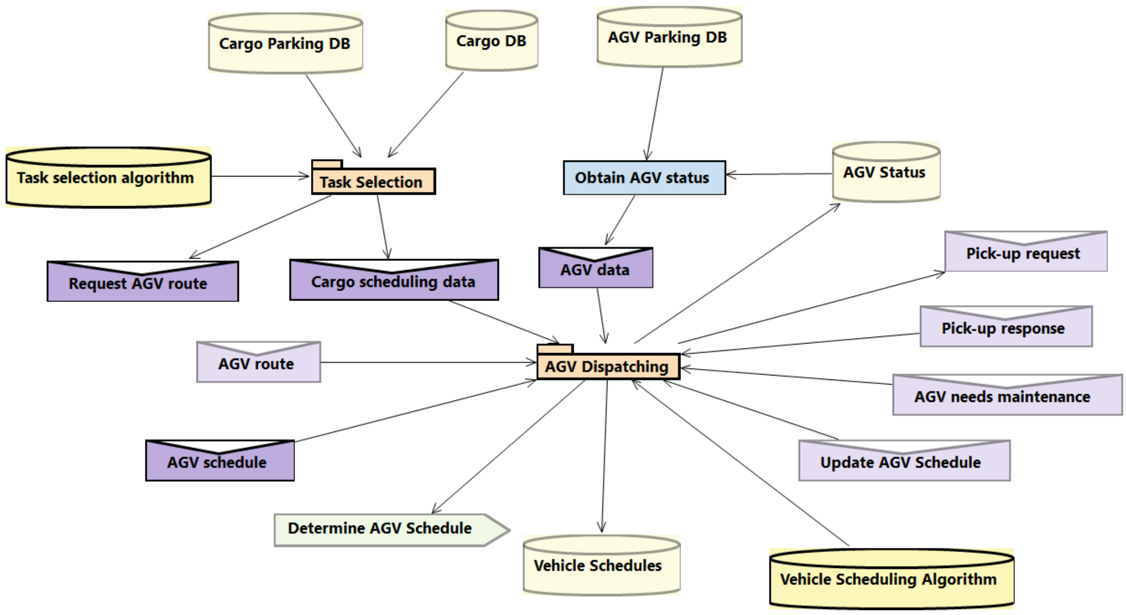


4.2 Vehicle Scheduling agent

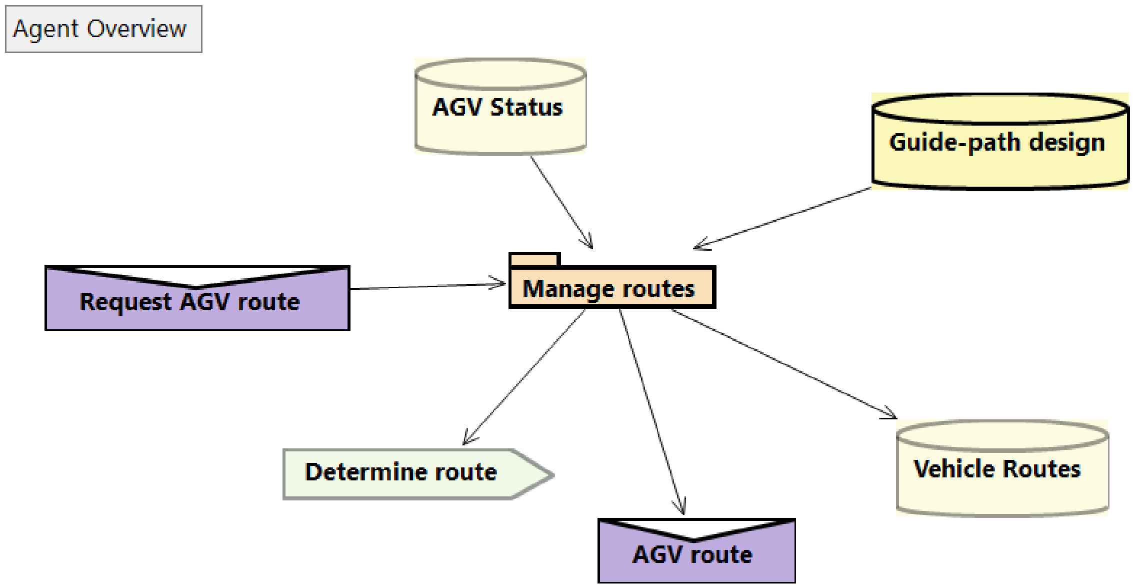
Agent Overview

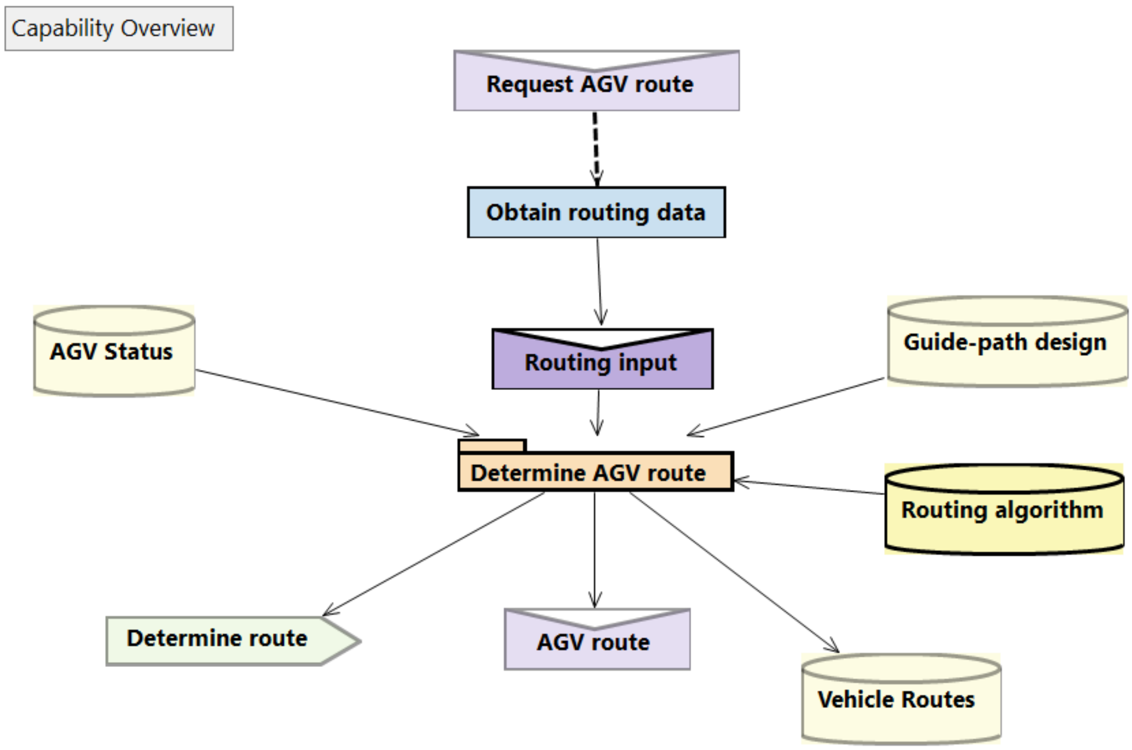


Capability Overview



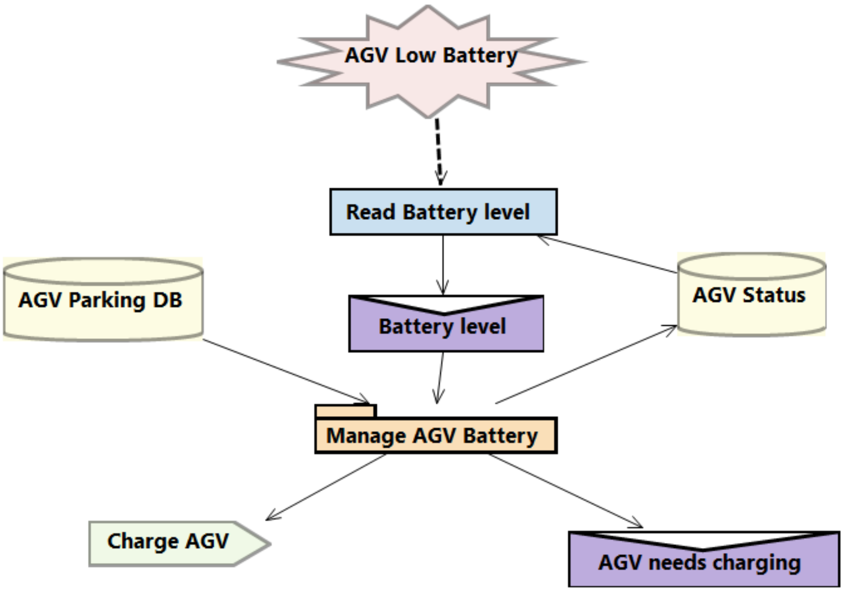
4.3 Vehicle Routing agent



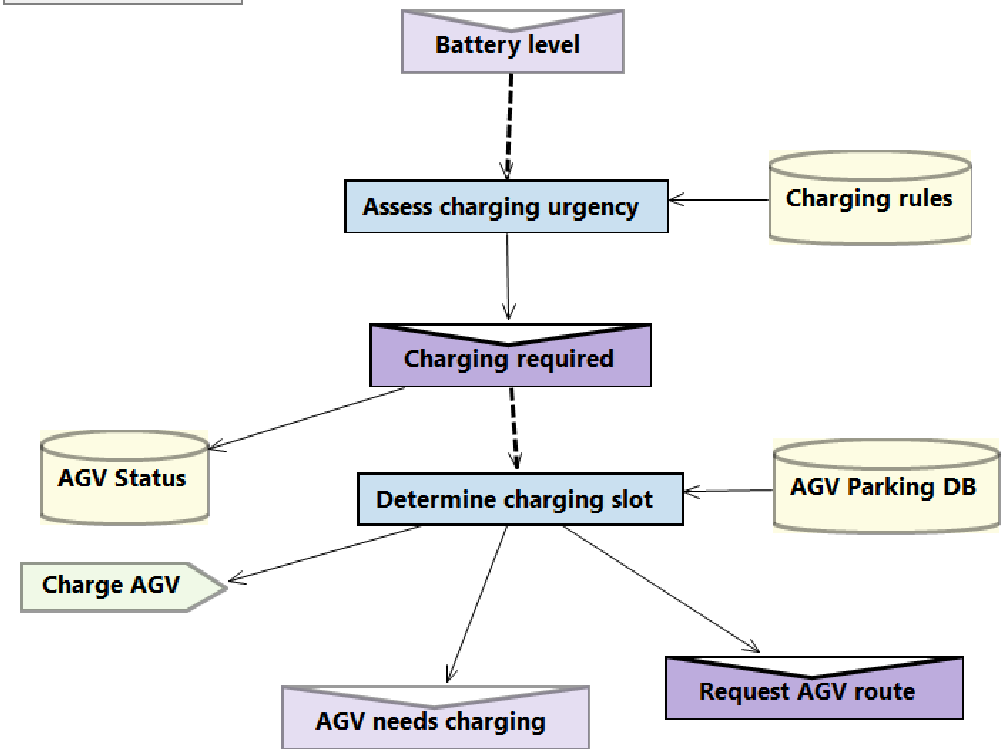


4.4 Battery Manager

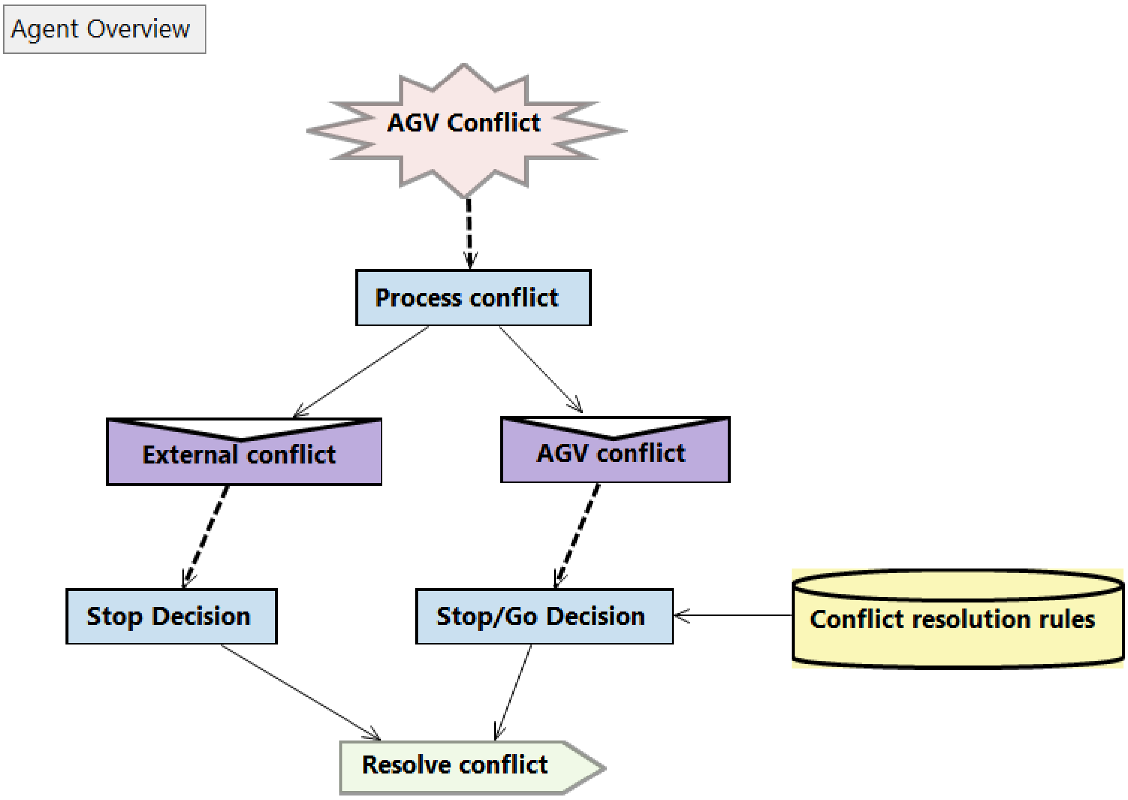
Agent Overview



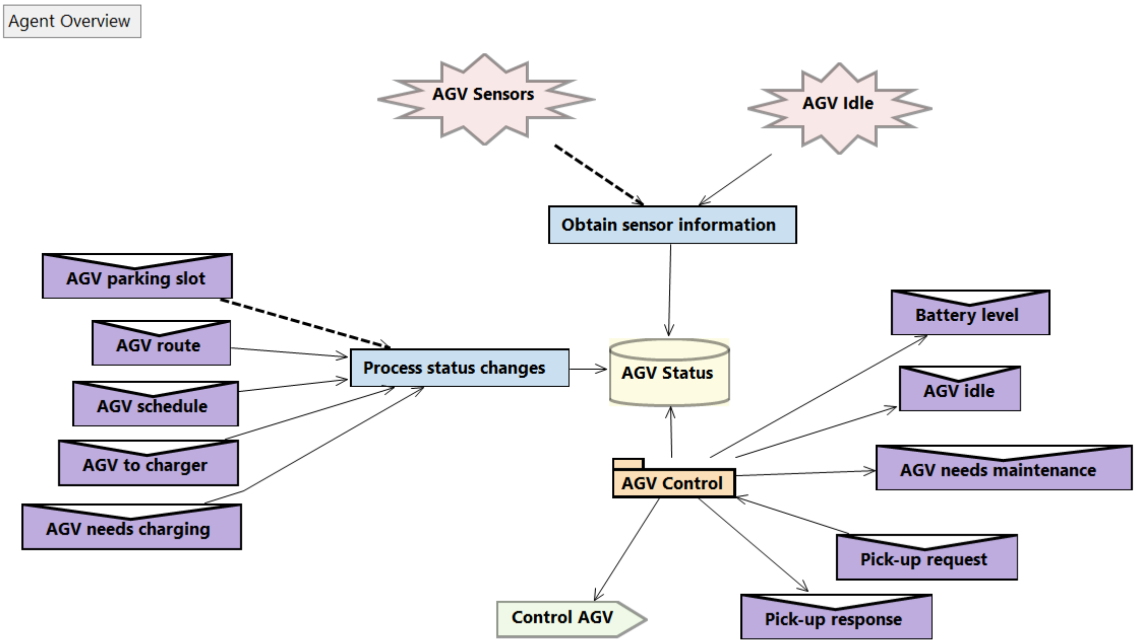
Capability Overview

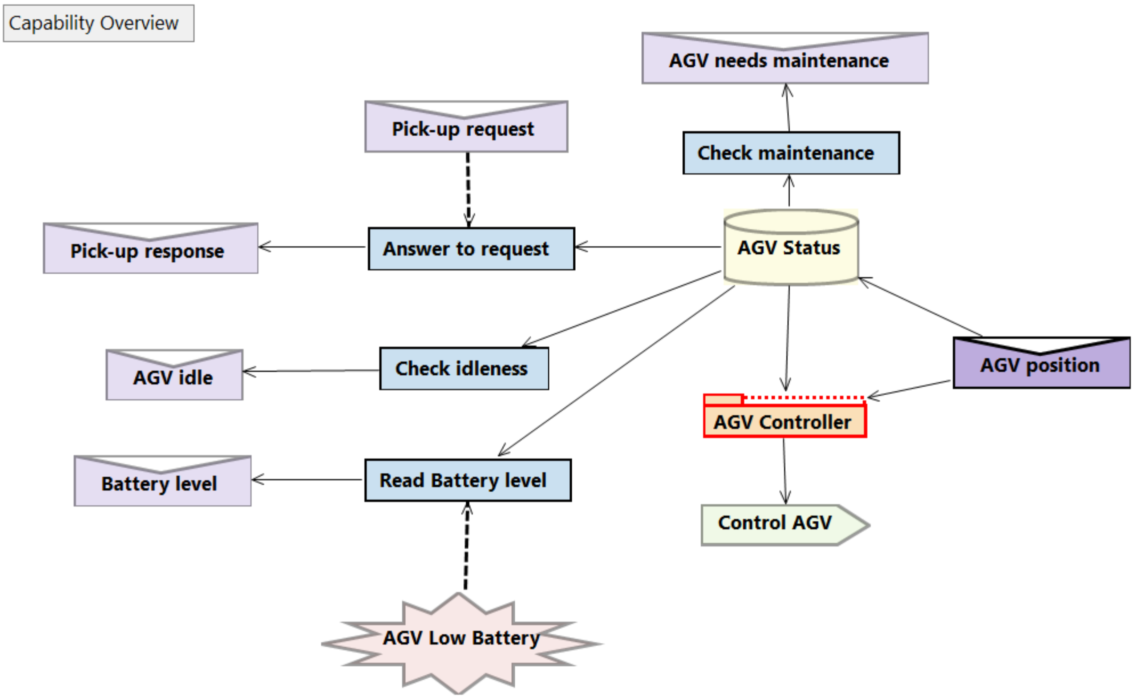


4.5 Conflict Manager



4.6 AGV Manager





5 CONCEPTUAL MODEL: SCOPE AND ASSUMPTIONS

Component	Include/ exclude	Justification
Entities		
Agents	Include	Implements MAS in simulation
Docks	Include	Response: dock utilization
Parking areas	Include	Responses: A/D parking, Empty Trailer parking and AGV parking utilization
Tracks	Include	Represents guide-path design
Semi-trailers	Include	Response: throughput of semi-trailers
Cargo in semi-trailers	Include	Experimental factor, determines unloading time and consolidation function
Garage	EXCLUDE	Maintenance assumed to be handled during system down-time
Activities		
Drop-off semi-trailer by Truck	Include	Key influence on parking area utilization
Pick-up at Arrival/Departure Parking	Include	Key influence on throughput
Drop-off at Arrival/Departure Parking	Include	Key influence on throughput
Rearward docking/parking	Include	Key influence on throughput
Empty trailer movements between cross-dock and Empty Trailer Parking	Include	Key influence on throughput
(De)coupling between AGV and semi-trailer	Include	Key influence on throughput, experimental factor
Loading/unloading semi-trailer	Include	Key influence on throughput, experimental factor
AGV Charging	Include	Key influence on throughput, experimental factor
Conflict/collision avoidance	Include	Facilitates real-life representation of system
Pick-up semi-trailer by truck	Include	Key influence on parking area utilization
Internal cross-dock operations	EXCLUDE	Limited impact on throughput of AGVs, assumed to have no impact on (un)loading operations
Pick-up semi-trailer by Truck	EXCLUDE	Instead remove semi-trailer from system. No influence on system performance.
Queues		
Cross-dock queues	Include	Facilitates consolidation of transport, assumed to have infinite capacity

Resources		
Trucks	Include	Required for transporting semi-trailers
AGVs	Include	Required for transporting semi-trailers
AGV charging equipment	Include	Experimental factor
Personnel in cross-dock	EXCLUDE	Assumed to be always available
Handling equipment in cross-dock	EXCLUDE	Assumed to be always available

6 CONCEPTUAL MODEL: LEVEL OF DETAIL

Component	Detail	Include/ exclude	Justification
Entities			
Agents	Quantity: depends on agent type	Include	One per system or one per entity/resource
	Other: bundling of methods	Include	Visually grouping of methods represents (functionality of) agent
Docks	Quantity: model input	Include	Model input.
	Attribute: (un)loading time	Include	Model input. Determines (un)loading time of one pallet
	Attribute: station type	Include	Switches dock between loading and unloading
	Attribute: failure	EXCLUDE	Assumed that loading/unloading never fails or gives delays
	Attribute: occupied	Include	Shows whether dock is scheduled for a job or is currently processing a job
Parking areas	Quantity: 3 (AGV, A/D & ET).	Include	Responses: A/D parking, Empty Trailer parking and AGV parking utilization
	Quantity: depends on input	Include	Model input. Number of parking slots per parking area
Tracks	Quantity: determined by guide-path design	Include	Represents guide-path design
	Attribute: length	Include	Model input
	Attribute: curve	Include	Connects horizontal tracks with vertical tracks
	Attribute: direction	Include	Determines difference between unidirectional and bidirectional tracks
	Attribute: capacity	EXCLUDE	The Conflict Manager should handle collisions/congestion
Semi-trailers	Quantity: model input	Include	Model input
	Arrival pattern: real life arrivals	EXCLUDE	Data not available. Instead include probability density function (e.g. Poisson)
	Departure pattern: real life departures	EXCLUDE	Data not available. Instead use check whether all cargo assigned to a departing transport has arrived, then depart.
	Attribute: consolidation	Include	Determines how many cargo (e.g. pallets) a semi-trailer contains and where it should go within the cross-dock
	Attribute: trailer ID	Include	Assigns unique ID to semi-trailer
	Attribute :type of movement	Include	Shows whether semi-trailer is arriving, departing or is moving empty

	Attribute: arrival dock	Include	Shows which unloading dock is assigned to a semi-trailer (uniform distribution)
	Attribute: departure dock	Include	Shows which loading dock is assigned to a semi-trailer
	Attribute: Arrival parking	Include	Shows which parking slot is assigned to a semi-trailer when arriving
	Attribute: ET parking	Include	Shows which parking slot is assigned to a semi-trailer at the Empty Trailer parking
	Attribute: Departure parking	Include	Shows which parking slot is assigned to a semi-trailer when departing
	Attribute: routing	Include	Determines route semi-trailer should take based on parking locations and arrival/departure docks
	Attribute: vehicle kinematics	EXCLUDE	All semi-trailers are assumed to be the same
Cargo in semi-trailers	Quantity: model input	Include	Model input. Use a probability density function to determine number of cargo in each trailer and thereby the unloading time
	Attribute: unloading dock	Include	Shows where cargo is unloaded
	Attribute loading dock	Include	Shows which cross-dock buffer the cargo should be unloaded to
	Attribute: arriving trailer	Include	Shows in which semi-trailer cargo has arrived
	Attribute: departing trailer	Include	Shows in which semi-trailer cargo has departed
Garage	Quantity: 0	EXCLUDE	Maintenance assumed to be handled during system down-time
Activities			
Drop-off semi-trailer by Truck	Quantity: number of arriving transports	Include	Experimental factor
Pick-up at Arrival/Departure Parking	Quantity: number of arriving transports	Include	Experimental factor
Drop-off at Arrival/Departure Parking	Quantity: number of departing transports	Include	Experimental factor
Rearward docking/parking	Cycle time: fixed	Include	Depends on rearward driving speed, model input.
	Routing: end of cross-road	Include	AGV or AGV+semi-trailer stops at end of the cross-road and then backs up.
	Breakdown	EXCLUDE	Assumed that docking or parking semi-trailer never fails

	Other: AGV load-specific	EXCLUDE	No difference between rearward driving time whether AGV is loaded with semi-trailer or not
Empty trailer movements between cross-dock and Empty Trailer Parking	Quantity: 2 per trailer	Include	From cross-dock to ET when finished unloading and from ET to cross-dock when ready for departure
(De)coupling between AGV and semi-trailer	Cycle time: fixed	Include	Model input
Loading/unloading semi-trailer	Quantity: depends on cargo in semi-trailer	Include	Key influence on throughput, experimental factor
AGV Charging	Cycle time: variable	Include	Depends on model input (continuous charging, battery swaps or conventional charging)
	Breakdown	EXCLUDE	AGV charging never fails
	Set-up: depends on model input	Include	AGV charging is replaced with a set-up/changeover when battery swaps are used
	Resources: charging facility	Include	Located at AGV parking area.
Conflict/collision avoidance		Include	Facilitates real-life representation of system
Pick-up semi-trailer by truck		Include	Key influence on parking area utilization
Internal cross-dock operations		EXCLUDE	Limited impact on throughput of AGVs, assumed to have no impact on (un)loading operations
Queues			
Cross-dock queues	Quantity: same as number of docks	Include	Temporarily stores cargo
	Capacity	EXCLUDE	Assumed to be infinite
	Queue discipline	EXCLUDE	Does not affect loading/unloading times
	Routing: cargo attribute dependent	Include	Determines where cargo should be stores based on departure dock attribute of cargo.
Resources			
Trucks	Quantity: total of arriving and departing transports	Include	Required for transporting semi-trailers
	Where required: drop-off and pick-up semi-trailer at A/D parking	Include	Required for positioning semi-trailer at terminal and to free up space at parking when departing transports arrives at A/D parking
	Shifts: 24 hours, 6 days a week	Include	Model input

AGVs	Quantity: variable	Include	Experimental factor.
	Where required: any activity involving semi-trailer movement within system boundary	Include	Key influence on semi-trailer throughput
	Shifts: 24 hours, 6 days a week	Include	Model input
AGV charging equipment	Quantity: same as number of AGVS	Include	Experimental factor
	Where required: any activity related to charging AGVs	Include	Enables charging of AGVs
	Other: charging type	Include	Experimental factor (continues charging, battery swaps or conventional charging)
Personnel in cross-dock	Quantity	EXCLUDE	Assumed to be always available
Handling equipment in cross-dock	Quantity	EXCLUDE	Assumed to be always available

7 MODELING THE GUIDE-PATH DESIGN

Table VII-1 contains the basic variables needed for the initialization of the track system. The values of these variables is input to the simulation. A short description is provided as well as realistic values.

TABLE VII-1 BASIC VARIABLES FOR MODEL INITIATION

Variable name	**Description**	**Realistic value(s)**
NumberOfDocks	Determines how many docks the cross-dock will feature.	75 (single-sided) 150 (double-sided)
DistanceBetweenDocks	The distance between two docks (heart-to-heart).	3.5-4.0m
LengthCrossRoad	The distance between the two main roads as a starting point for rearward docking or moving between the two main roads.	Depends on angle between main road and crossroad. In case of 90°: ~18m.
LenghtDockingArea	The distance between the dock and the top main road. Measured from the front of the dock to the heart of the main road.	20-26m
CurveLength	The length of a curved road. Quarter of a circle.	$\frac{1}{4}\pi$*(DistanceBetweenDocks^2)
LengthParkingArea	The distance between the parking area opposite to the cross-dock and the bottom main road. Measured from bottom of parking to heart of bottom main road.	Equal to LengthDockingArea

Using the *Init* we can easily assign values to these parameters at the start of every simulation run and reset the entire track system using *Reset*. An example of initiating the track system with the value of *NumberOfDocks* being two and some arbitrary values for the other variables, is shown in Figure VII-2.

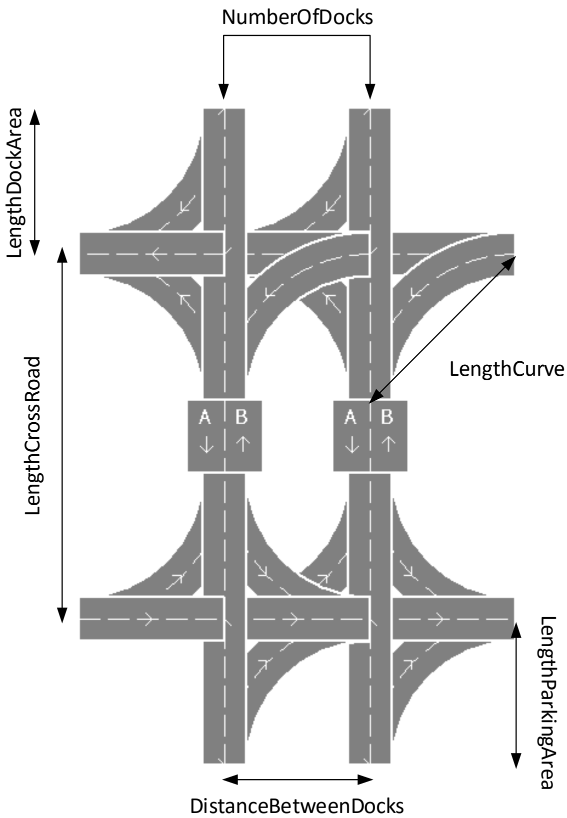


FIGURE VII-1 ILLUSTRATION OF CREATING TRACKS USING VARIABLES

7.1 Technical description

With the few variables thus far we can create tracks resembling the guide-path design introduced in Section 5.3.1 in front of the cross-dock. AGVs are able to drive on these tracks to find their way to pick-up and drop-off locations. Figure VII-2 shows how the variables together define the guide-path. The figure also shows the driving directions of each track, east-to-west on the top main road and west-to-east on the bottom main road. The crossroads in the middle are bidirectional tracks as AGVs should also be able to drive backwards to dock or park a semi-trailer. In our particular case these tracks consist of two lanes, A (north-to-south) and B (south-to-north). The curves connecting the main road with the crossroad are connected to the lane which corresponds to the driving direction (e.g. the bottom curve going upwards is connected to lane B) shows three different configurations of these variables, all featuring five docks. The left layout is our initial configuration. The middle configuration increases the value of LengthCrossRoad, resulting in more space between the two horizontal main roads. The right configuration resets LengthCrossRoad to our initial value and increases DistanceBetweenDocks as well as LenghtDockingArea and LengthParkingArea. This results in more space between the docks, longer curved segments and a longer docking- and parking road.

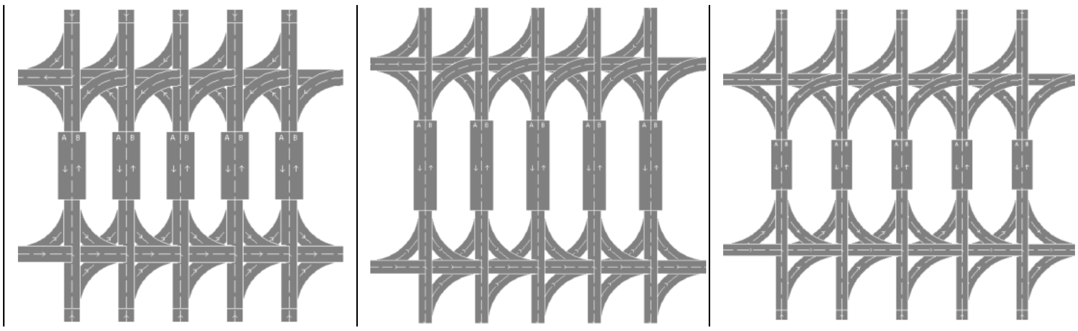


FIGURE VII-2 THREE DIFFERENT CONFIGURATIONS OF THE TRACK SYSTEM

As one of our goals is to build a flexible model, these changes we can make to the track layout is very helpful. This however also imposes some modeling challenges. When we change the values of the variables responsible for the shape and size of the guide-path, we also need to make sure all track segments are positioned at the right place in our model and aligned correctly. This can be solved to create a reference point for all pieces of track to be inserted in the model. For every dock in our model we need a fixed amount of tracks (i.e., the dock area, the crossroad, the parking area, a top main road, a bottom main road and six curved segments to connect everything). We can thus loop over the number of docks defined in our initialization and create the tracks required per dock in every iteration. Within every iteration it creates the track segments one by one, taking into account the variables discussed before, and adjusting the x- and y-coordinates for the next track segment.

We created a method called *CreateTracks* to build the guide-path at the cross-dock area and is called by *Init.* The following pseudocode shows a small snippet of the entire method, focusing only on creating the '*TopTracks*', which are the track segments of the top main road.

```
Initialize StartXPos and StartYPos

for i = 1 to NumberOfDocks do
    Create TrackSegment named TopTrack[i] at coordinates
(StartXPos,StartYPos)
    Set length TopTrack[i] to DistanceBetweenDocks
    Rotate TopTrack to set the direction westwards
    Write name and coordinates of TopTrack[i] to a table inserted at
row i

    StartXPos = StartXPos + DistanceBetweenDocks
```

This code will create all the *TopTrack* segments, making sure that the next *TopTrack* created, is located at the appropriate place. Note that the y-dimension does not change. Similarly, all other tracks can be created by increasing or decreasing the value of the starting coordinates and changing the orientation and length as needed. We write all track segments and their coordinates to a table file, which will be of use later on in the model. In every row the track segments of the corresponding dock are stored (dock area, cross road, parking area, … of dock 4 are stores in row 4). This enables us to quickly find the track segments belonging to a certain dock as they are all stored within the same row and all have the row number in their name (e.g. TopTrack4, CrossRoad4, DockRoad4). This is especially useful when we want to identify the tracks that need to be connected to other track segments, such that the AGV can move from track to track. Most of the tracks we want to connect are all in the same row of the table. Exceptions are tracks that need to be connected to a track in an adjacent row (e.g. connect TopTrack1 with TopTrack2) and connecting the outer ends of the main roads. After all tracks are created, some extra tracks are created and connected to close the left side of the loop. An example of the result of *CreateTracks* using five docks is shown in Figure VII-3. The tracks on the right side, connect the cross-dock with the Arrival/Departure parking and the AGV parking.

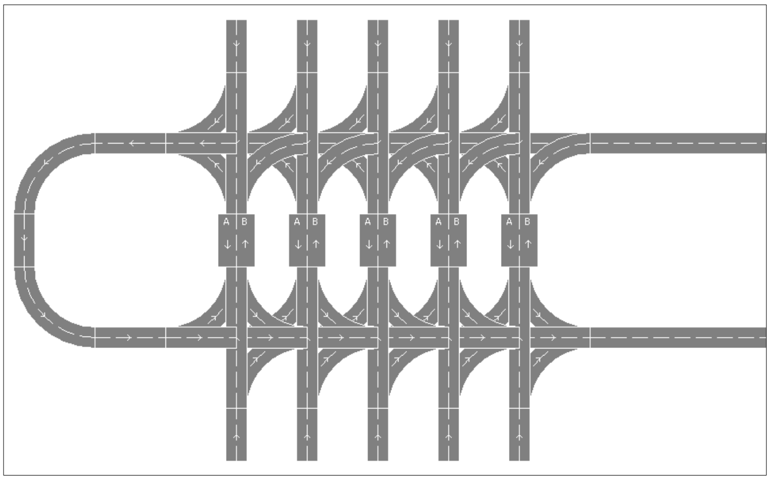
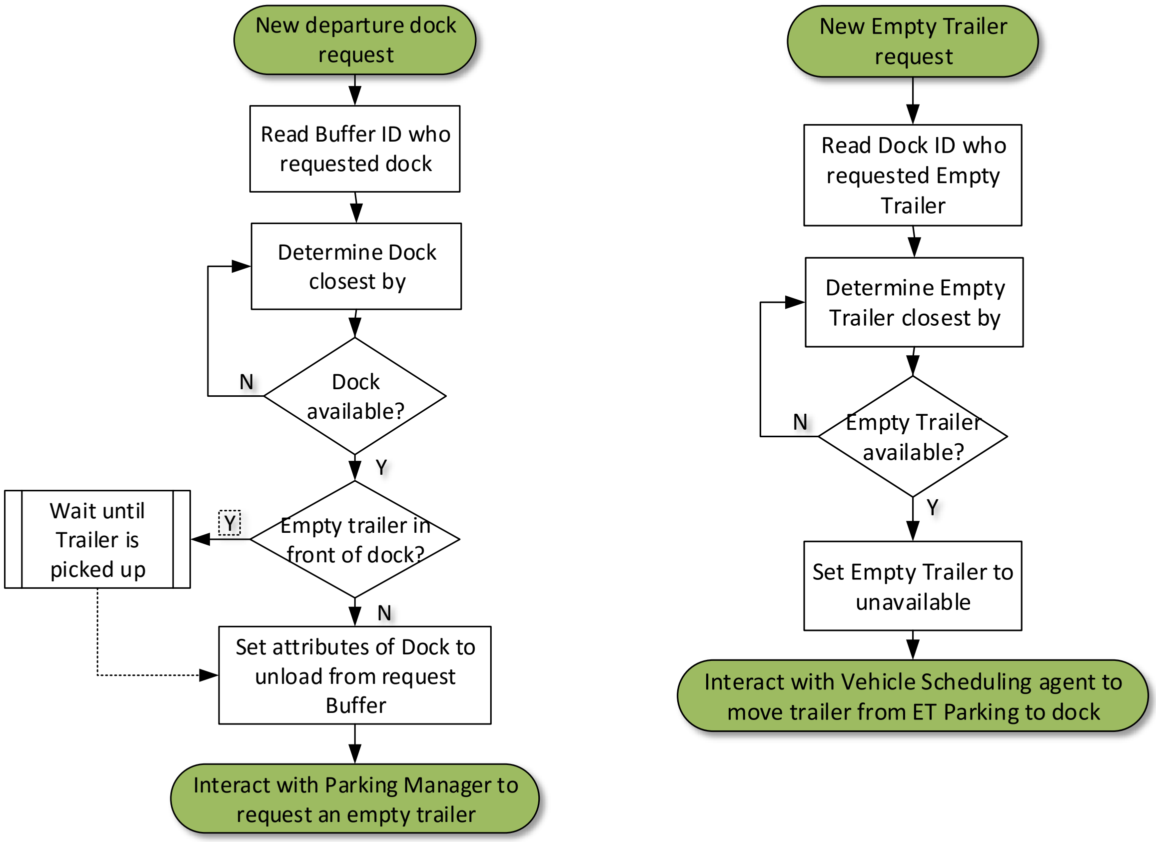


FIGURE VII-3 RESULT OF THE *CREATETRACKS* METHOD

8 FLOWCHARTS CONCEPTUAL MODEL

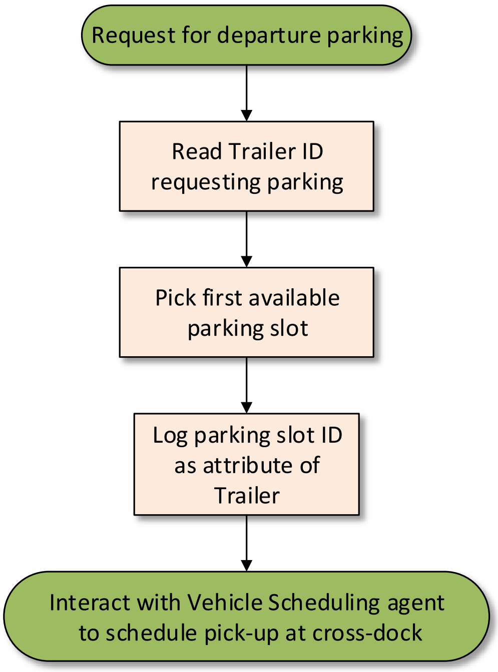
8.1 Determining the nearest available dock and empty trailer

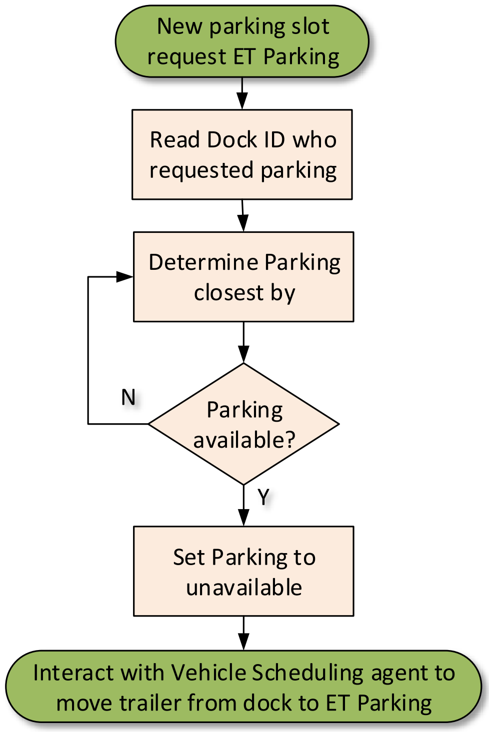
The flowchart on the left shows how the nearest dock available is determined. The flowchart on the right shows the succeeding actions required to pick an empty trailer from the parking area opposite of the cross-dock. Both inhabit the same logic. When the dock/trailer closest by is not available, it will check whether the dock/trailer second closest by is available and so on. Picking the dock closest by, minimizes the time between the trigger for departure and the actual start of the loading the cargo and picking the trailer closest by minimizes the travel distance between the empty trailer parking and the cross-dock.



8.2 Assigning parking slots to departing transports and empty trailers

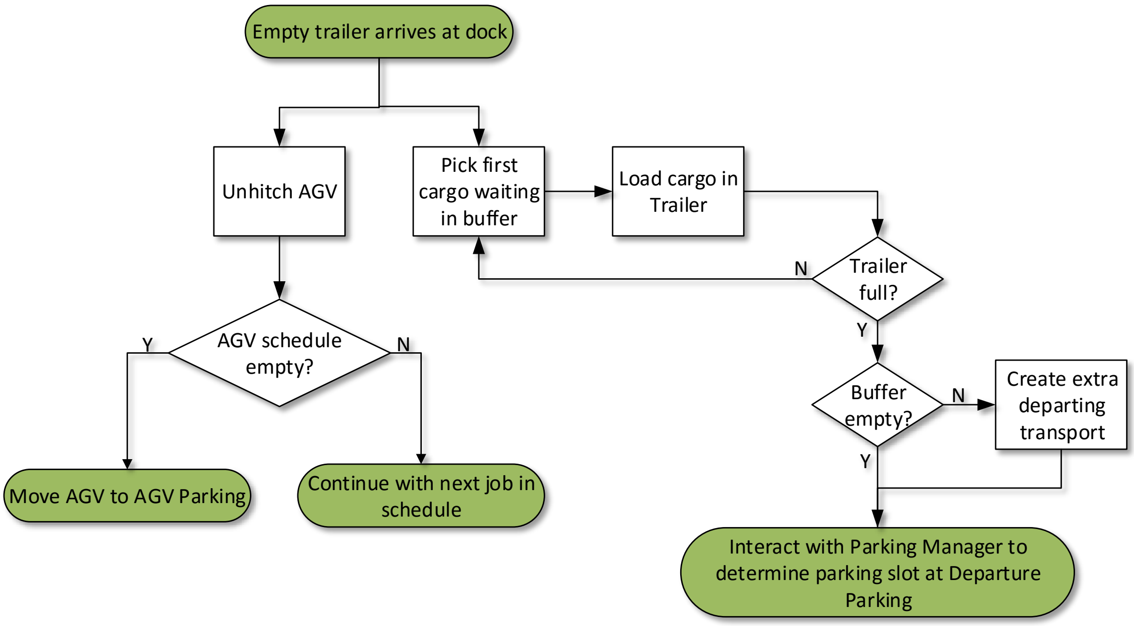
The left flowchart shows what happens in the event of an empty trailer parking slot request, which is triggered when a trailer is fully unloaded. On the other hand, the event in the right flowchart is triggered when a trailer is fully loaded and thus requiring a parking slot at the Arrival/Departure Parking.

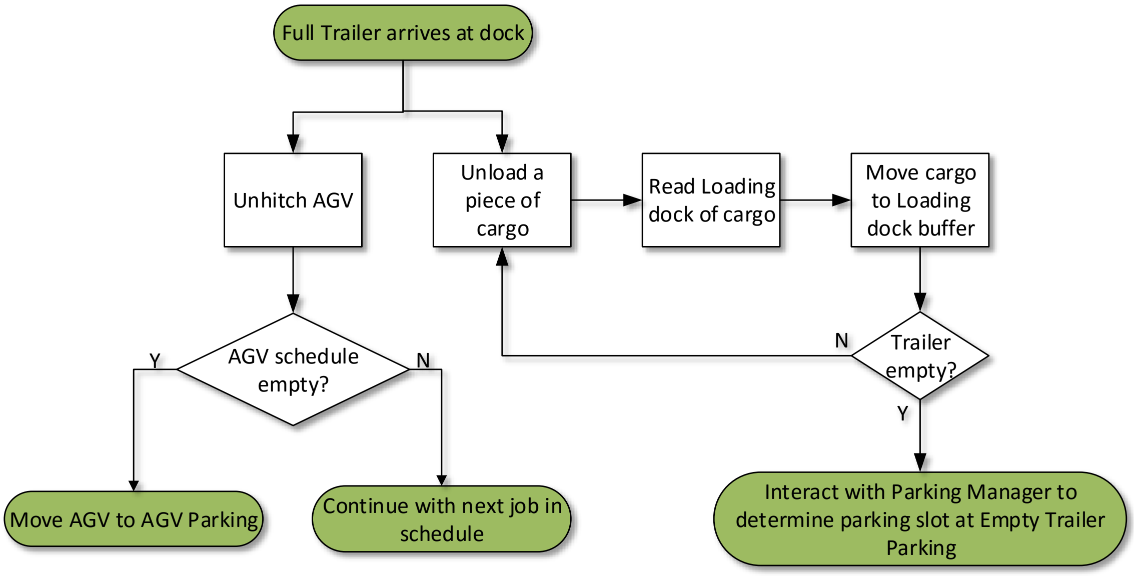




8.3 Arriving and departing transports at dock

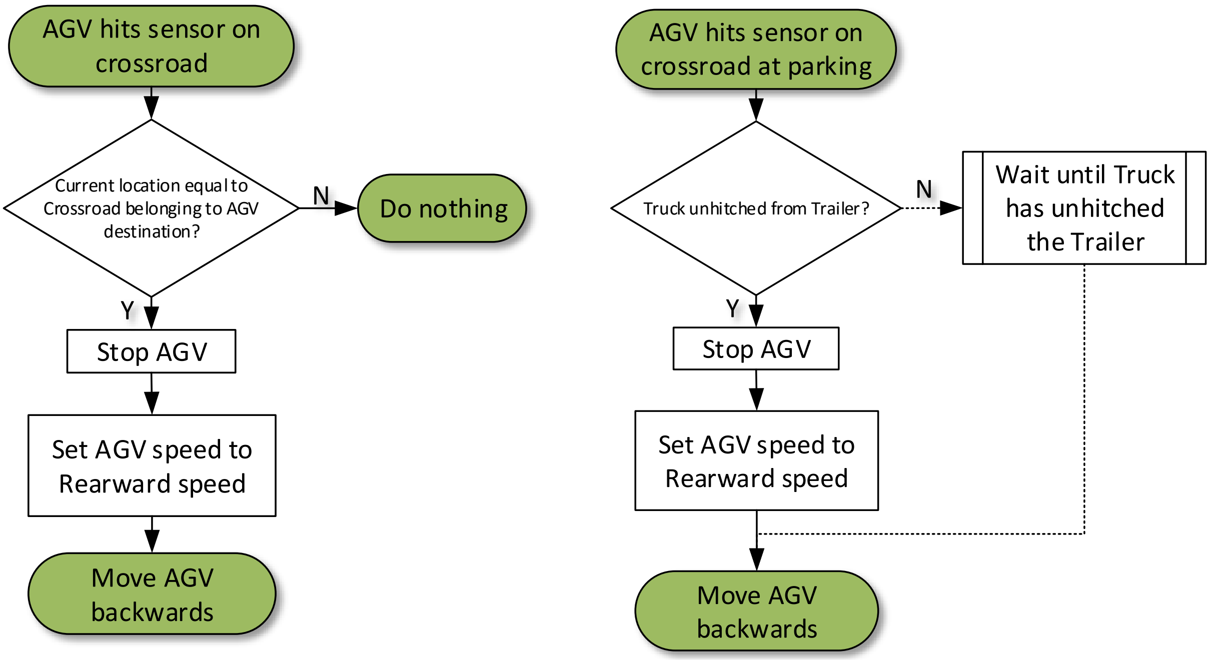
The flowcharts below show how the system responds when an empty trailer arrives at a dock (and thus needs to be loaded) and when a full trailer arrives at a dock (and thus needs to be unloaded).





8.4 Rearward driving control of AGV

When an AGV drives on a crossroad which is the crossroad corresponding to the dock/parking where the semi-trailer hitched to the AGV needs to be, the AGV needs to stop and switch its gear to rearwards such that the docking/parking maneuver can start. The flowcharts below show how these events are handled in the simulation.



9 SIMULATION MODEL OUTPUT

9.1 Simulation output (3 AGVs)

Run	Avg Arrival Parking	Avg Arrival Dock	Avg ETParking	Avg Departure Dock	Avg Departure Parking	Avg CycleTime [hh:mm:ss]	Avg Travel distance [m]	Avg Difference Travel distance [%]	Avg. Traveltime [mm:ss]	Avg Diff. Traveltime [mm:ss.0]
1	15.2	37.9	38.1	38.2	18.5	7:28:51	1129	-0.86%	04:22.5	00:05.8
2	14.6	37.0	38.1	38.4	26.0	7:40:23	1153	-0.68%	04:28.2	00:05.6
3	13.1	39.6	38.4	36.8	27.2	7:35:56	1159	-1.96%	04:29.4	00:05.9
4	13.9	37.9	37.8	38.4	25.5	7:49:02	1153	-1.41%	04:28.2	00:06.1
5	14.0	37.8	38.1	37.9	24.2	7:36:49	1141	-0.88%	04:25.3	00:05.8
6	12.8	38.5	37.4	37.4	22.4	7:38:17	1121	-1.20%	04:20.6	00:05.8
7	12.4	35.2	39.2	39.5	22.1	7:31:50	1124	-0.25%	04:21.5	00:05.6
8	13.8	39.1	37.7	37.0	20.3	7:25:24	1132	-1.83%	04:23.3	00:05.7
9	13.3	39.3	37.7	37.9	27.9	7:32:53	1155	-1.59%	04:28.6	00:06.4
10	13.2	37.9	36.9	37.8	27.2	7:38:26	1147	-1.13%	04:26.9	00:05.7
Avg	**13.6**	**38.0**	**37.9**	**37.9**	**24.1**	**07:35:47**	**1141.5**	**-1.18%**	**04:25.5**	**00:05.8**

* Avg = average, Diff = difference.

9.2 Simulation output (2 AGVs)

Run	Avg Arrival Parking	Avg Arrival Dock	Avg ETParking	Avg Departure Dock	Avg Departure Parking	Avg CycleTime [hh:mm:ss]	Avg Travel distance [m]	Avg Difference Travel distance [%]	Avg. Traveltime [mm:ss]	Avg Diff. Traveltime [mm:ss.0]
1	33	38	40	39	32	13:15:04	1336	-1.33%	0:05:11	00:09.4
2	32	40	39	38	29	12:47:53	1317	-2.09%	0:05:06	00:08.3
3	32	40	38	36	33	12:57:19	1332	-2.82%	0:05:10	00:08.8
4	32	38	38	38	28	12:44:51	1307	-1.68%	0:05:04	00:09.1
5	33	37	37	38	30	12:54:06	1317	-1.08%	0:05:06	00:08.8
6	32	38	38	38	31	12:34:21	1316	-1.79%	0:05:06	00:08.4
7	33	38	39	39	32	12:49:24	1332	-1.20%	0:05:10	00:09.6
8	34	39	38	37	33	12:55:08	1344	-2.11%	0:05:12	00:08.7
9	33	40	38	38	29	12:41:22	1327	-2.19%	0:05:09	00:09.1
10	33	39	38	38	33	12:53:29	1341	-2.12%	0:05:12	00:08.8
Avg	**32.6**	**38.7**	**38.1**	**37.8**	**31.0**	**12:51:18**	**1327**	**-1.84%**	**0:05:09**	**00:08.9**

* Avg = average, Diff = difference.

9.3 Simulation output (1 AGV)

Run	Avg Arrival Parking	Avg Arrival Dock	Avg ETParking	Avg Departure Dock	Avg Departure Parking]	Avg CycleTime [hh:mm:ss]	Avg Travel distance [m]	Avg Difference Travel Distance [%]	Avg. Traveltime [mm:ss]	Avg Diff. Traveltime [mm:ss.0]
1	43	38	38	39	52	31:05:51	1490	-1.39%	0:05:46	00:12.0
2	43	37	38	38	51	30:46:42	1482	-1.10%	0:05:45	00:12.1
3	43	39	40	38	52	30:55:54	1489	-1.34%	0:05:46	00:11.7
4	43	40	40	37	50	30:47:55	1499	-1.84%	0:05:49	00:12.4
5	42	36	36	39	52	31:03:06	1485	-1.03%	0:05:45	00:12.0
6	43	38	38	37	53	31:20:26	1492	-1.85%	0:05:47	00:11.6
7	42	38	38	38	49	30:43:31	1481	-1.56%	0:05:44	00:11.9
8	43	38	39	38	54	31:19:40	1506	-1.63%	0:05:50	00:12.6
9	43	38	38	38	53	30:52:39	1487	-1.82%	0:05:46	00:11.6
10	43	38	37	37	52	31:03:32	1485	-2.01%	0:05:45	00:11.7
Avg	**42.7**	**38.1**	**38.2**	**37.8**	**51.8**	**30:59:56**	**1489.5**	**-1.56%**	**0:05:46**	**00:12.0**

* Avg = average, Diff = difference.