Information integration and intelligent control of port logistics system

Feng, Fan

DOI
10.4233/uuid:88fbd0ea-7998-4d6f-a6ae-ac65114e6d95

Publication date
2019

Document Version
Final published version

Citation (APA)
Feng, F. (2019). *Information integration and intelligent control of port logistics system*. [Dissertation (TU Delft), Delft University of Technology]. TRAIL Research School. https://doi.org/10.4233/uuid:88fbd0ea-7998-4d6f-a6ae-ac65114e6d95

Information Integration and Intelligent Control of Port Logistics System

Fan Feng

Cover design: XiaoJie Luan

Information Integration and Intelligent Control of Port Logistics System

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof. dr. ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op dinsdag 22 januari 2019 om 10:00 uur

door

Fan FENG

Master of Science in Embedded System,
Technische Universiteit Eindhoven, Nederland
geboren te TaiYuan, ShanXi, China

Dit proefschrift is goedgekeurd door de promotoren:
Promotor: Prof. dr. ir. G. Lodewijks
Copromotor: Dr. ir. Y. Pang

Samenstelling promotiecommissie bestaat uit:

Rector Magnificus	voorzitter
Prof. dr. ir. G. Lodewijks	Technische Universiteit Delft, promotor
Dr. ir. Y. Pang	Technische Universiteit Delft, copromotor

Onafhankelijke leden:

Prof. dr. H. Hu	Shang Hai JiaoTong University, China
Prof. dr. J. van Hillegersberg	University of Twente
Prof. dr. ir. L. A. Tavasszy	Technische Universiteit Delft
Prof. dr. ir. M. van Koningsveld	Technische Universiteit Delft
Prof. dr. R. R. Negenborn	Technische Universiteit Delft

This thesis is the result from a project funded by the China Scholarship Council [Grant number 201307720072], and the section of Transportation Engineering and Logistics in Department of Maritime and Transport Technology, Delft University of Technology in the Netherlands.

TRAIL Thesis Series no. T2019/1, The Netherlands TRAIL Research School
P.O. Box 5017
2600 GA Delft, The Netherlands
E-mail: info@rsTRAIL.nl

Published and distributed by: Fan Feng
E-mail: fengfan0607@gmail.com

ISBN 978-90-5584-244-5

Printed in the Netherlands

To my family

Acknowledgements

When I started my college study at the age of 18, pursuing a Ph.D. abroad seems to be a dream which is far beyond my imagination. Surprisingly, I got the chance to continue my master in the Netherlands and even further for a Ph.D. Well, this is the moment I know that this long adventure towards a Ph.D. will be over in a few weeks. This makes me firmly believe that, sometimes, life is much more exciting than imagination. Of cause, those people who around me make this long adventure more colourful and joyful. I want to express my gratefulness to them.

First, I would like to express my gratitude to my promotor Prof. Gabriel Lodewijks for giving me an opportunity to conduct research under your supervision. I would not have finalised this thesis without your trust, commitment, professional guidance, and wisdom. Thanks for your inspiring discussions and innovative ideas that keep motivating my research. Of cause, your sense of humour makes me feel less anxiety when facing difficulties. Second, I would like to express my gratitude to my supervisor Dr. Yusong. Dear Yusong, thanks for your assistance and support during the past years. Your help is definitely beyond what this thesis covers. Your encouragement and trust give me courage during my hardest time in both work and life. I am grateful for your way of supervision which gives me enough room for conducting independent research. Our weekly meetings and leisure talks are the most valuable resources I have during the past years.

In addition, I would like to thank all the committee members for your time and your comments on the early version of the thesis. Your comments definitely improve this thesis significantly. Furthermore, my sincere thanks go to the people who provided support for my work. Thank Dick for helping me to translate the summary into Dutch. Many thanks to Jozephina, Patty, and Dineke for always being kind and helpful to my research.

I would also like to express my gratitude to my colleges in TUDelft. Many thanks to the colleges in the Transportation Engineering and Logistics section for sharing ideas and stories regarding research and life. It has been an honour for me to work in such an inspiring and friendly section.

I would like to thank my friends in the Netherlands for their supports and accompany for the past years. Thanks for my roommates Yang, Jia and XiaoCheng for their kind accompany. Many thanks to my friends Tian, Gang, Chuang, Minghe, Zixuan, DaiJie

and WenJie for making this Ph.D. life colourful. Also thanks to all small basketball team members. It was a great fun to play with you guys. Moreover, I would like to thank Ma and HuBiao for being good friends with me for the past eight years. Also, thanks to my friends in China, though we might not see each other often, I know I have all your support.

I would thank my parents for your unconditional love and support. You are truly a pillar of strength for me. Our weekly video talks feed me with the strength and power I need to continue this adventure. For my sister and brother in law, thanks for taking care of our family when I was absent. Thanks for my lovely nephew, your sweet smile makes life colourful. Finally, I would like to thank my wife, XiaoJie. Surely, complete this Ph.D. journey is a huge step, but meet you and in love with you is definitely the most important thing for me during this journey. Thanks for your accompany, your support, your understanding, your encouragement, and your smile.

Fan Feng

Delft, December 2018

Contents

Chapter 1

Introduction

The main focus of this thesis is to develop an ICT framework and apply it to support decision-making processes in a port logistics system. This chapter introduces the thesis by providing a contextual background and outlining the main research questions. In light of this, this chapter is broken down into four main sections. The background and definition of port logistics are discussed in section 1.1. The purpose of this research is discussed in section 1.2. The research questions are outlined in section 1.3. Finally, section 1.4 presents a roadmap of this thesis by outlining the overall structure and content of the thesis as a whole.

1.1 Background

A port provides an essential interface between sea and land transportation. It is responsible for receiving imported goods into a country and sending exported goods out of a country. In 2013, it was estimated that over 80% volume of all globally traded freights were carried by sea and handled by seaports in (UNCTAD, 2014). As a result of the high carrying capacity and the low cost of seaborne shipping, port logistics has become a key player in modern global freight transport (Tseng et al., 2005).

Port Logistics (PL) is defined as the process of planning, implementing and controlling the flow of goods and information from seaside to inland side via ports and vice versa (Herz & Flämig, 2014). PL systems concern the development of functions to support activities include seaside and landside transportation, cargo storage, order processing, and distribution. The complexity of a PL system is illustrated in Figure 1.1. It involves different types of terminals, a variety of equipment and infrastructures, a diverse number of stakeholders, and information systems used to support operations. As shown in Figure 1.1, a vessel comes along the berth for loading and unloading cargoes with the assistance of terminal crews and on-site facilities. Before tranship to its destination, the unloaded cargoes will go through several processes which include, stevedore, transit, storage, and inland transport. To execute PL operations, a plan is required to ensure all resources, including facilities, and equipment, are available and properly used. A control system is needed to provide a robust operation. A comprehensive information system is required to ensure the parties involved are properly coordinated and the plans are executed punctually.

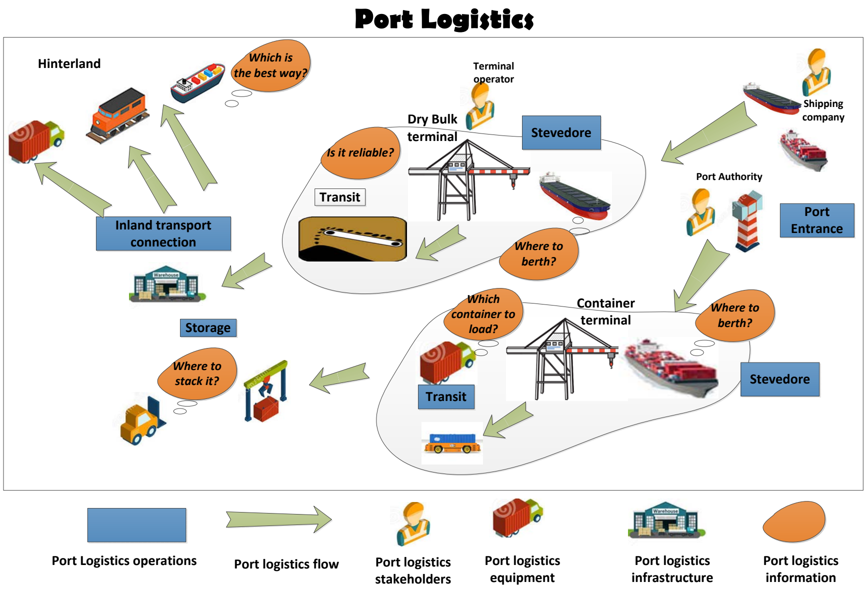

Figure 1.1: A simplified schematic of Port Logistics

On-time delivery and cargo tracking are only part of the objectives pursued by a PL system. These goals not only require proper management of the flow of goods but also rely heavily on a timely and accurate flow of information (Langley Jr, 1985). Obtaining the right information at the right time and the right place is crucial for executing PL processes effectively. To improve the management of information flow, the application of information communication technology (ICT) is indispensable (Lai et al., 2006). The term ICT encompasses a variety of different systems, devices, and services used for data processing. This also includes telecommunications equipment and services for data transmission and communication (European Commission, 2008a). An ICT-enabled system plays an important role in a PL system. It is responsible for the functioning of cargo tracking, custom automation, resource planning, and coordination. For example, the Port of Hong Kong designed and established its terminal operating system known as ”3P” in 1997. The introduction of this system results in a significant enhancement of the overall performance of a port. This included a 20% increase in crane productivity and a cost reduction of handling per TEU by more than 35% (Murty, Wan, et al., 2005). Both Langley Jr (1985) and Introna (1991) inferred that the use of ICT can enhance logistical effectiveness, efficiency and flexibility. Consequently, adoption and implementation of an advanced ICT system are universally recognised as a prerequisite for PL success.

1.2 Motivation and aim of the thesis

1.2.1 ICT advancements in PL systems

Increasing demands and highly competitive markets have forced PL systems to improve their performance. This includes improvement of handling capacity and quality of services. Physical expansion can be costly and may be constrained by a port geographic position within an urban centre. In response to this, UNCTAD (2006) proposed an alternative to innovate existing processes and operations, enabled by ICTs, to improve the performance of existing operations and infrastructures. Furthermore, Cepolina and Ghiara (2013) have highlighted that over the past ten years, more than 30 European research projects have been financed by European Union (EU), all of which have focused on the ICT solutions in European ports. Also, they noted that this would be the main target of EU funding in the near future. Accordingly, it can be expected that ICT advancements could potentially improve the performance of PL systems. Several potential improvements are considered below.

(1) **Use of ICT to enhance communication and information sharing between different actors**. There are numerous stakeholders involved in the PL chain, and each has different needs and expectations. For example, the terminal operators usually concern about the terminal resource utilisation level while the road and rail carriers pay careful attention to service punctuality. The overall competitiveness of the PL chain depends on the effectiveness of coordination and the synergy created among those stakeholders. Therefore, establishing effective coordination and information sharing is essential (UNCTAD, 2006). However, in current practice, the degree of coordination and synergy is far from optimal due to a low level of ICT adoption. For example, enhancing the port-hinterland connection is a significant concern in the Port of Rotterdam (Franc & Van der Horst, 2010). It is suggested that ICT is a critical factor for this challenge (S. Li et al., 2014). However, most works remain on a conceptual or simulation level. There are three main reasons for this. First, concerns about privacy mean that stakeholders prefer to stay autonomous and are reluctant to hand over their control. Second, stakeholders may have a conflict of interests. This means that establishing a trusted platform through which to coordinate all the operations is a challenge faced by both researchers and policymakers (Moonen, 2009). As such, the benefits from ICT-enabled decision-making system are so far unknown yet.

(2) **Use of ICT to improve interoperability of PL system**. Interoperability is defined as the ability of two (or more) systems to exchange information and use information (D. Chen & Daclin, 2007). It concentrates on the interactions between human-human, human-machine, and machine-machine. This not only refers to the standard of data exchange but also to a common understanding of the exchanged information. The hybrid information system and intelligent machines are primarily applied to support PL processes. This includes, for example, automated guided vehicles and smart sensors. The immense data generated from these systems (e.g., the location of containers and

vehicles, priorities of containers, vehicle arrival times, equipment conditions) needs to be made accessible in an appropriate way. Moreover, the information consumer (e.g., the terminal operator) has to understand the current situation or to be aware of the context in which it happened in order to understand the real meaning of the information. For example, the condition of the equipment can be monitored by intelligent sensors. It is essential to understand how to extract system condition from sensor data and ensure the information is given at the right place and at the right time (Lodewijks et al., 2016). To fulfil these requirements, interoperability has to be improved. This is because the number of intelligent machines and devices is expected to be increased significantly as a result of the current trend towards automated terminal design in future PL development. Consequently, interoperability will be of vital importance for PL systems. However, both scientific and practical projects rarely address how to improve interoperability in the PL domain.

(3) **Design an integrated ICT framework which could combine ICT with methodologies from other domains**. The focuses of practical projects and scientific outputs are not universal. On the one hand, recent EU projects have focused extensively on implementing ICT tools from a pure software perspective. This means they concentrate on the system connectivity, user acceptance, and interoperability of the developed system (UNCTAD, 2006). On the other hand, outputs from the scientific domain are more concerned about solution quality from a pure algorithmic perspective. For example, applying optimisation methods to improve specific objectives (Hartmann, 2004). In practice, most decision-making problems simultaneously require consideration of the communication, system flexibility, and decision quality. To achieve this, integrating ICT tools with methodologies from other domains may be beneficial from both theoretical and practical aspects. However, such integration receives little attention in the logistics and transport domain (Perego et al., 2011). As a result, the benefits are yet to be presented.

(4) **Adapt new ICT initiatives**. In recent years, the ICT innovations have advanced significantly. Technologies, such as distributed agent systems, context-aware systems, the Internet of things, big data and cloud computing, have already been successfully integrated into various industrial domains (Yılmaz & Erdur, 2012; Maturana & Norrie, 1996; Gao et al., 2015). However, the usability of, and potential for, applying new ICT technologies have not been thoroughly investigated regarding improving the performance of the PL system (Harris et al., 2015).

The potential benefits of developing a new ICT framework for PL systems can be briefly summarised as follows: (1) improve coordination and collaboration between various stakeholders participating in a PL system; (2) enhance interoperability of a PL system with respect to information exchange and processing; (3) achieve sustainable and reliable operations via real-time monitoring and accurate decision making; (4) promote operational efficiency by delivering an integrated solution.

1.2.2 Aim of the thesis

In light with the potential benefits discussed above, this thesis aims to develop a new ICT framework to improve the performance of PL systems. More specifically, this thesis aims to use ICT to support PL decision-making (DM) at different levels. Figure 1.2 presents a framework of decision-making problems in a PL system. Three levels can be identified: a strategic, a tactical, and an operational level based on the time horizon involved (Hendriks, 2009). Decision-making problems at the strategic level concern the layout design of port and the handling capacity design in the long-term (years). This is out of the scope of this research. The decision-making problems at the tactical and operational level are the main focuses of this thesis. A brief introduction of theses two DM level problems, and the specific problems this thesis addressed are presented as below.

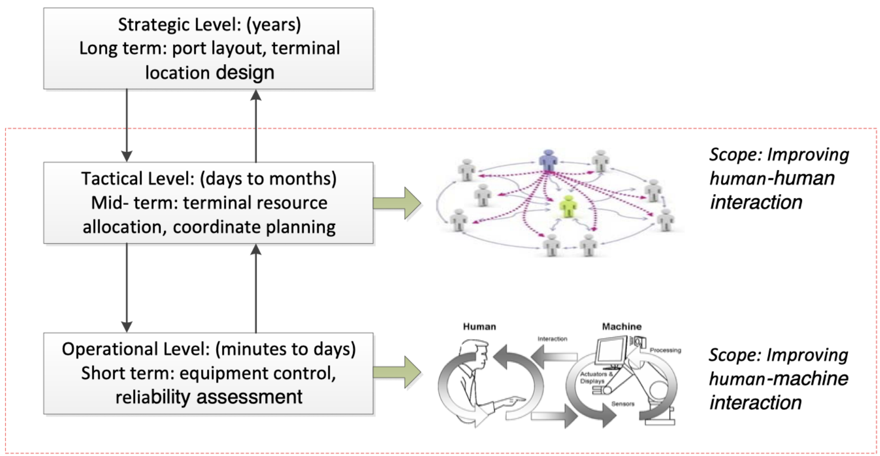

Figure 1.2: The decision-making problems of PL system at different DM levels. The red dashed block presents the research scope of this thesis.

- **Tactical level decision-making**: A tactical decision covers a period of between one day and several months. This level focuses on issuing decisions concerning vessel routing, resource allocation, and transport planning. At sea, ship routing and planning problems are categorised as typical tactical decision-making problems (Christiansen et al., 2004, 2007). In particulate, there is a focus on minimising total cost while ensuring that the cargo is delivered from their loading port to their port of discharge on time. At the terminal, a tactical level decision-making considers the resource allocation and transport planning problems, such as berth allocation (Moorthy & Teo, 2006) and storage yard assignment (Leech, 2010). At the hinterland, a a tactical level decision-making considers the planning and coordination of different hinterland transport infrastructure, such as trains (Hansen, 2004), waterway transport (Moonen, 2009) and road transport (Guan & Liu, 2009; Phan & Kim, 2015).

At the tactical level, this thesis addresses a hinterland barge transport planning problem. The problem faces several challenges. First, the participants may have different objectives regarding the planning process. This means information sharing is difficult to achieve. Second, there is insufficient support for decision-making. As a result, the average rotation time and the terminal resource utilisation level are unsatisfactory. In light of those challenges, the aim is to design an ICT system to provide ICT support to improve coordination and collaboration between participants, and optimise planning quality which include barge turn-around time and the terminal utilisation level.

- **Operational level decision-making**: Operational decisions are made in the short-term and concern scheduling and execution of the daily operations by considering possible disturbances (e.g., weather) and uncertainties (e.g., equipment breakdown). A typical operational level decision-making is usually based on deciding which equipment to use for which operations, such as quay crane scheduling (Duinkerken et al., 2002; Lee et al., 2008) and equipment control (Xin et al., 2014).

  At the operational level, this thesis addresses the challenge posed by reliability assessments of large-scale equipment in a PL system. The difficulties associated with achieving sound reliability assessments include (1) insufficient support of data mining and information integration; (2) lack of automated decision making. Therefore, the aim is to develop an ICT system which focuses on data acquisition and exchange, information integration, and decision making. The goal is to improve the interoperability of the reliability assessment process by analysing and delivering equipment condition assessment in an integrated way.

In sum, this thesis aims to take advantage of ICT technologies to improve the performance of a PL system. By designing and developing a new ICT system, it seeks to show the benefits of effective use of ICT by first improving human-human coordination at the tactical level, and second, by improving human-machine interaction at the operational level.

1.3 Research questions

To achieve the research goal, two main research questions have been formulated. Furthermore, each research question also contains several sub-questions.

1. How can a new ICT framework be implemented and integrated into a Port Logistics system?

To answer this research question, three sub-questions need to be examined:

- **1.1 What is the research status of decision-making problems and the development status of ICT applications in a PL system?**

  To answer this sub-question, a thorough literature review will be carried out on two aspects. First, the current literature on PL decision-making problems will be examined. Second, the current ICT applications that have been developed to support the PL decision-making will be assessed. The goal is to identify the existing challenges at the tactical and operational decision-making level and analyse the potential improvements that can be made from an ICT perspective.

- **1.2 What are the potential ICT technologies that can be used to address decision-making problems?**

  To answer this sub-question, different ICT technologies will be assessed based on the requirements derived from the first sub-question. In short, two major ICT components need to be selected, namely, (1) a middleware, which will be used to support information integration and exchanges and sharing among different entities in a PL system; (2) an intelligent decision-making approach which will be used to support information processing and decision making.

- **1.3 How can ICT technologies be integrated in a PL system?**

  To answer this sub-question, an ICT framework will be designed to integrate the selected ICT technologies into a generic architecture. It should be flexible and scalable in order to cope with the identified decision-making challenges.

2. What are the benefits of using a new ICT framework in Port Logistics?

The second question will be answered by conducting two case studies. The case studies consider two specific decision-making problems at the tactical and operational levels respectively. To answer this research question, two sub-questions need to be examined:

- **2.1 What are the benefits of using a new ICT framework at the tactical decision-making level in a PL system?**

  To answer this sub-question, the problem of hinterland barge transport planning in a container terminal is addressed. The proposed ICT technologies and the framework will be applied to show the improvement of the planning performance at the tactical decision-making level. More specifically, a mediator-based agent system and a hybrid meta-heuristics algorithm are proposed and designed. The designed ICT system aims to achieve coordination and collaboration between participants and improve the quality of rotation plans concerning total turn around time and terminal capacity utilisation.

- **2.2 What are the benefits of using a new ICT framework at the operational decision-making level in a PL system?**

  To answer this sub-question, the problem of reliability assessments in a large-scale belt conveyor system in a dry bulk terminal is studied. The proposed ICT technologies and the framework will be implemented as a context-aware system to demonstrate the improvement of the reliability assessment at the operational decision-making level. The developed context-aware system includes a mediator-based agent system and an ontology knowledge. The ontology is developed to model the domain of belt conveyor reliability assessment and used to transfer heterogeneous data into linked through data association and logic reasoning. The agent system is developed to support condition assessment. The agent system interacts with the ontology to obtain assessment context and knowledge. To assess the usability and applicability of the system, a case study of belt tear condition supervision is studied.

1.4 Thesis outline

Figure 1.3 presents an overall structure of this thesis. The thesis is split into two parts. Each part answers one main research question respectively.

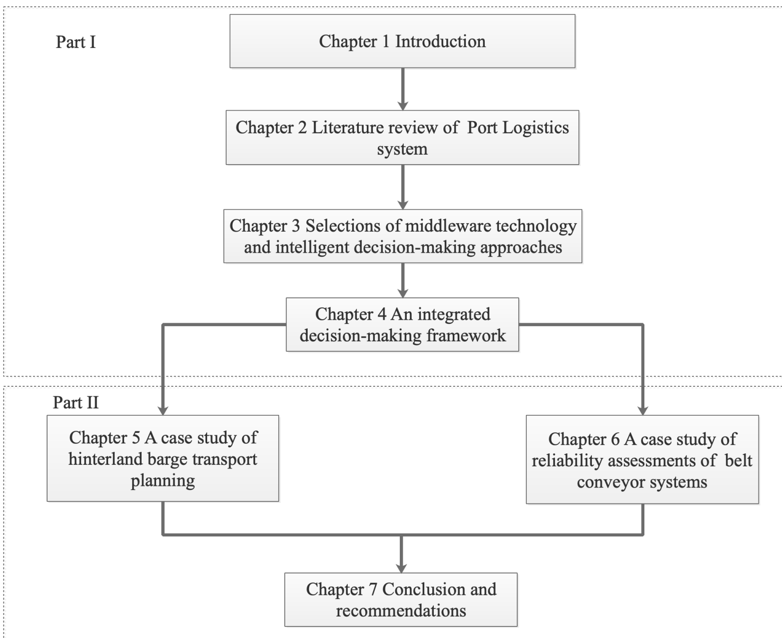

Figure 1.3: The structure of this thesis

The first part answers the first research question by clarifying the challenges in the current PL practice, select ICT technologies used in this research, and implement an integrated ICT framework.

- In Chapter 2, the current literature on PL systems is reviewed. In the first part of this chapter, two decision-making problems are discussed. These are the hinterland barge transport planning and the equipment reliability assessment. Relevant studies are reviewed, and the gaps in the current research regarding solving these two problems are identified. In the second part, a review of ICT applications in a PL system is presented. It explores the role of ICT applications in a PL system and investigates the ICT developments in addressing decision-making problems. Based on the review, the essential elements of implementing a new ICT system are identified, namely, middleware and an intelligent decision-making approach. This chapter answers the research question 1.1.
- Based on the requirements proposed in Chapter 2, Chapter 3 selects the elements of the ICT system. This includes the selection of middleware and intelligent decision-making approaches.
  - Middleware: The multi-agent system is chosen as the middleware tool to support information exchange, sharing, and coordination. The result is derived by comparing the agent-based system with an object-based system and an expert system.
  - Intelligent DM approach: This selection depends on the nature of the problems. Therefore, two different solution approaches are selected for each DM problem. The meta-heuristics approach is chosen to support barge transport planning, and the context-aware system and ontology are chosen to support reliability assessment.
- In Chapter 4, an integrated ICT framework is proposed. By comparing different architectural types, a hierarchical framework is chosen as the architecture of the integrated ICT framework. The goal of designing such an ICT framework is to integrate the technologies proposed in Chapter 3. It combines the middleware and the intelligent DM approaches into a hierarchical framework and subsequently applied it to different DM problems. This chapter answers research question 1.3.

The second part of this thesis focuses on assessing the proposed ICT framework by solving decision-making problems in a PL system. This part answers the second main research question.

- In Chapter 5, the ICT framework is assessed by solving a tactical level DM problem. More specifically, an integrated planning system is designed to assist the hinterland barge transport planning. A bi-level programming model is constructed, and a multi-agent system (MAS) combined with meta-heuristics is implemented. This chapter defines how planning activities are executed, which level of collaborations are incorporated, and how benefits are achieved. This chapter answers research question 2.1.

- In Chapter 6, the ICT framework is assessed by solving an operational level DM problem. A context-aware supervision system is designed for the reliability assessment of a large-scale belt conveyor system. The proposed framework adapts the semantic web technology to capture the general concepts and knowledge of a supervision system. A multi-agent system is designed to assist major supervision tasks through agent intelligence and collaboration. This chapter designs the how the ontology is built, how the agent system is organized, and how to achieve agent-ontology integration. This chapter answers research question 2.2.
- In Chapter 7, the conclusion of this thesis and the recommendations of future research directions are put forward.

According to the outline of this thesis presented in Figure 1.3, readers interested in the hinterland barge transport planning problem for PL tactical level DM could read the dissertation in the following order: Chapter 1, sections 2.1, 2.2, 2.3, 2.6 and 2.7 of Chapter 2, sections of 3.1, 3.2 and 3.3 of Chapter 3, Chapter 4, Chapter 5, and Chapter 7.

Those interested in the reliability assessments of large-scale belt conveyor system problem for PL operational level DM could read the dissertation in the following order: Chapter 1, sections 2.1, 2.4, 2.5, 2.6 and 2.7 of Chapter 2, sections 3.1, 3.2 and 3.4 of Chapter 3, Chapter 4, Chapter 6, and Chapter 7.

Chapter 2

Literature review of Port Logistics system

This chapter presents a literature review of PL systems. It contains two parts, as shown in Figure 2.1 below. The first part of the review aims to explore the studies related to DM problems in PL systems. Based on the review, the DM challenges addressed in this thesis and their research gaps will be identified. Section 2.1 introduces the general operations in PL systems. In section 2.2, tactical DM problems raised in PL systems are reviewed. Besides, the challenge of hinterland barge transport problem is further introduced in section 2.3. In section 2.4, operational DM problems raised in PL systems will be reviewed. Furthermore, section 2.5 introduces the challenge of reliability assessment for large-scale belt conveyor systems.

The second part of the review aims to explore existing ICT applications in PL systems. Based on this review, the research gaps and requirements for future PL developments will be clarified from an ICT perspective, specifically for addressing DM challenges. Section 2.6 provides a review on the recent ICT developments in PL systems. The current challenges and their potential solutions are discussed in section 2.7. Finally, a conclusion is drawn in section 2.8.

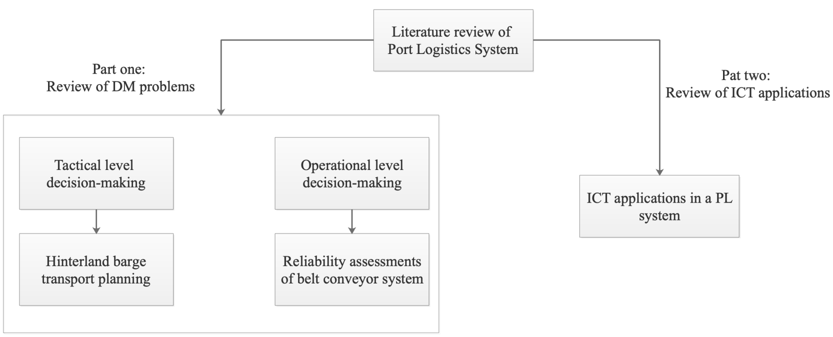

Figure 2.1: An outline of chapter 2

2.1 An overview of Port Logistics processes

The concept of PL mainly focuses on planning and controlling the flow of goods and information between sea and inland via ports. After arrival at the terminal, the inbound goods need to be unloaded and transshipped to their destination. Outbounds goods will then be loaded onto the vessel. The flow of goods will go through several processes. This includes stevedore, transit, storage, and inland connections. Efficient PL processes require the execution of different operations in a coordinative and collaborative way. In order to achieve this, information must be integrated and shared. This leads to a further focal point of the PL system, namely, information flow management. Figure 2.2 presents the major PL processes. Each process will be briefly explained below.

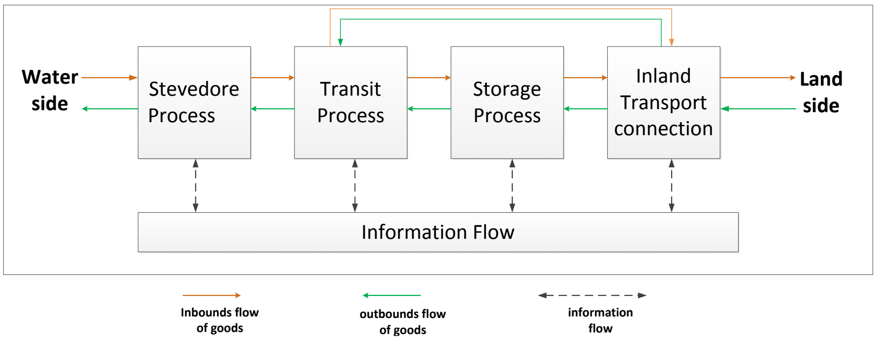

Figure 2.2: Main Port Logistics processes

- The stevedore process includes the loading or unloading of cargo to and from a vessel at a sea or inland terminal. After arrival at a port, a vessel is assigned to a berth location and equipped with different types of equipment for loading and discharging cargo. For example, a trolley crane can be used by container terminal, and a grab crane can be used by dry bulk terminal.

- The transit process involves transporting cargo between the quayside and the storage facility. This is done using different types of equipment. For example, at a container terminal, different types of trucks and vehicles are used. At a dry bulk terminal, continues transport equipment such as a belt conveyor system are employed.

- The storage process is used to temporarily store cargoes until they are ready to be shipped to their destination by different transport modes. The equipment used in this process also varies depending on the type of cargo. For a container terminal, the containers are stacked in ties, which can be made up of several containers. A yard crane performs stacking and retrieving operations. At a bulk terminal, bulk materials are stored in piles where a stacker-reclaimer is widely used.

- The term inland transport connections is used to describe the connection between port and cargo destinations. Depending on their destination, cargo may be transshipped to another vessel, or dispatched to the terminal gate for transport by trucks or trains. For countries with a well-developed waterway network, barge delivery is preferred because more cost-efficient. If the cargo needs to arrive more quickly, however, rail transport is preferred.

The performance of PL systems can be determined by examining several factors. This includes the turn-around time of vessels, the resources deployed for handling loading/discharge operations, the waiting time for inter-terminal transport trucks and trains, and the congestion on the roads and at the storage yard.

In order to improve of these factors, effective DM systems are essential. As highlighted in Figure 2.3, the DM problems can be categorised into 3 levels: strategic, tactical ,and operational level. Each decision plays a role in the overall performance of the PL system. When plans to build a new terminal are put in place, strategic decisions are very important for example, consideration of equipment capacity and the size of the storage area. When considering how to improve the performance of the existing systems, more attention is paid to optimise tactical and operational decisions. Sections 2.2 and 2.4 will discuss the DM problems at these two levels.

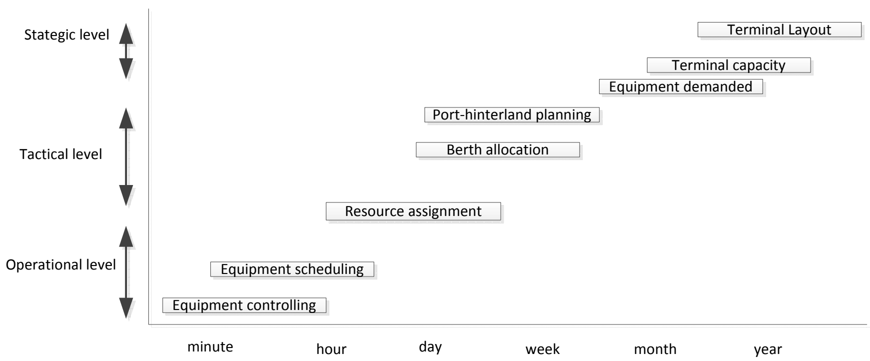

Figure 2.3: Overview of DM problems in PL, (adapted by Van Zijverden and Negenborn (2012))

2.2 A categorisation of tactical decision-making problems

A tactical decision covers a time-span between a day and several months. This level focuses on issuing decisions concerning vessel routing, resource allocation and transport planning. In this section, the tactical decisions made in different areas (e.g., seaside, terminal side, hinterland) are reviewed.

2.2.1 Sea side

At seaside, a typical tactical decision concerns a ship routing and planning problem (SRSP). Ship routing is defined as the assignment of sequences of ports to be visited by the ships and scheduling implements temporal requirements into routing. In general, the SRSP consists of three types of shipping modes: liner shipping, industrial shipping and tramp shipping. In linear shipping, the routes of vessels are fixed. The objective is to maximise the profits through fleet deployment. For industrial shipping, the operators have full control of their cargo and ships. Given the planning horizon and service start time, the primary objective is to minimise total cost while ensuring that the cargo is delivered from their loading port to their port of discharge on time. This is done via cargo routing and planning. Compared with industrial shipping, the scale of tramp shipping is small and random. The vessels are deployed where cargo is available. The primary goal is to maximise the profit by making decisions as associated with ship and cargo routes. An overview of ship routing and scheduling problems can be found in Christiansen et al. (2004, 2007).

2.2.2 Terminal side

Inside the terminal, tactical decisions typically consider resource allocation and transport planning. At the quayside, berth allocation and quay crane assignment are considered. For each vessel, information includes the physical parameters of the vessel (length and scale), expected arrival time, and projected handling time are given to the terminal operator before the arrival of the vessel. This is done in order to minimise the total berthing time. A tactical berth allocation problem (TBAP) is formulated and solved with Tabu search in (Giallombardo et al., 2010). Moorthy and Teo (2006) also addressed the TBAP problem by formulating a bi-criteria optimisation problem and solved through a heuristics algorithm. At the land side, storage space allocation considers the use of the yard space for stacking and efficiently retrieving cargo. At a container terminal, containers can be stacked in ties made up to several containers. Decisions about storage allocation aim to find the best position in the storage yard in order to minimise the operation cycle time for storing (Expósito-Izquierdo et al., 2012), retrieving (Vis & Roodbergen, 2009) and reshuffling containers (Carlo et al., 2014; K. H. Kim & Hong, 2006). At a bulk terminal, different policies for assigning storage location to piles were introduced by (Leech, 2010). It should be noted that the decisions made at a tactical level (e.g., berth allocation problem) may be modified when the ship arrives at the port. Operational decisions are invoked to avoiding traffic congestion at both the quay and yard side, and to reduce ship loading and unloading time (Stahlbock & Voß, 2008).

In addition to single DM processes, integrated DM has also received attention from a variety of scholars. For example, Zheng et al. (2017) proposed a closed-loop energy-efficient scheduling method and control of an autonomous Inter-Terminal Transport

(ITT) system using waterborne Autonomous Guided Vessels (waterborne AGVs). The developed system considers both tactical ship scheduling and operational control simultaneously. Giallombardo et al. (2010) studied the integration, at a tactical level. It focuses on the berth allocation with the assignment of quay cranes, aims to maximise utilisation of the chosen quay crane and to minimise housekeeping costs. Meisel and Bierwirth (2013) considered the berth allocation, quay crane assignment, and quay crane scheduling problem simultaneously to improve resource utilisation rate and service quality using heuristics. Murty, Liu, et al. (2005) proposed an integrated decision system to optimise storage space, vessel berthing time, external truck waiting time, and landside congestion. This resulted in a 30% reduction in turn-around time and a 35% reduction in handling cost. A detailed survey can be found in Bierwirth and Meisel (2010).

2.2.3 Hinterland side

Delivering cargoes to destination relies on a well-developed connection between the terminal and the hinterland (Notteboom, 1997). Figure 2.4 depicts a general view of the hinterland transport chain. Various modes of transport, such as trucks, trains, and barges, are used to access the hinterland. Different parties are involved in this process. They include shipping lines, terminal operating companies, forwarders, hinterland transport providers, and inland terminal operators. The tactical DM problem at the hinterland side concerns the transport planning of trucks, ships and trains. To enable efficient hinterland transport, attention should be focused on the time and cost (M. Van der Horst & Van der Lugt, 2009).

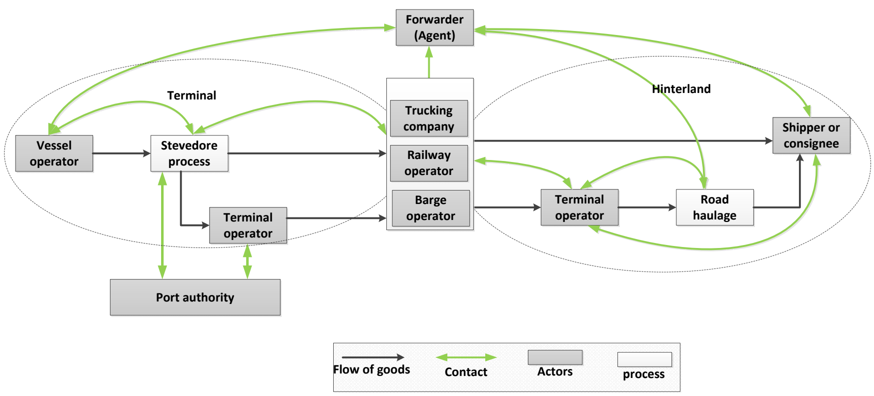

Figure 2.4: A hinterland container transport logistics chain, derived from (Van Der Horst & De Langen, 2008)

- Road transport: Transportation by truck is a major method of delivering cargo to its customer via road. Due to the lack of coordination between the truck companies and terminal operating companies, congestion can occur during peak hours at the terminal gate area. In turn, this can lead to an unpredictable turn around times and insufficient capacity utilisation of trucks. An appointment system is used to support negotiation between truck companies and terminal operators by predetermining the maximum number of trucks that can pass through the gate during a specified time window and rejects applications after the maximum available capacity is reached (Guan & Liu, 2009; Phan & Kim, 2015). A time windows system has been used to keep truck operators informed on the vessel arrival information, and allocates a time window for truck operations (G. Chen et al., 2013).

- Rail transport: Compared the truck transportation, the coordination problems in rail transport is more complicated. On the one hand, railway operations are inflexible, slow and expensive (Hansen, 2004). Therefore different resources, such as the availability of the train path, should be planned in order to maximise profits. This requires close cooperation between terminal company, train operators and railway companies (M. R. Van der Horst & Van der Lugt, 2014). On the other hand, because there is no contractual relationship between the terminal and the train company, information about the arrival of cargo is unavailable to the train operator. This could cause regular delays as a result of the late arrival of vessels. In order to address this, Nozick and Morlok (1997) proposed a mixed-integer model to support decisions such as train lengths, engine allocations, traffic routines and work allocation for the terminal. Furthermore, Corry and Kozan (2006) developed an analytical tool to optimise the operation strategy in the planning of container trains.

- Waterway transport: Barge transportation is an important mode of transport for the hinterland connection. Similar to rail transport, the barge company has no contractual relationship with the terminal operating company. Thus a reliable operation at the terminal cannot be guaranteed (Moonen, 2009). Moreover, where rail transport has dedicated infrastructure(e.g., rail terminal), barges have to share the same infrastructure with sea-vessels which are given higher priority. As a result, any delays in sea-vessel operations will propagate to barges operations (Van Der Horst & De Langen, 2008).

In this section, a categorisation of different tactical level DM problems is presented. Based on the review, it can be found that tactical DM involves multiple actors, different equipment, and a variety of processes. The tactical level DM complexities are explained by the coordination between different actors, the balancing of different interests, and the planning of various resource. The following section will focus on a specific DM problem in the current PL practice. This problem is associated with coordination problem of hinterland barge transport planning.

2.3 Tactical DM challenge: hinterland barge transport planning

2.3.1 Hinterland barge transport planning practice

In general, hinterland transport consists of truck, barge, and rail transport. Rotterdam is located at the delta of the rivers Rhine and Meuse. This makes inland barges the ideal method for reliable and cost-effective transportation of containers. It is estimated that the use of barges in the Port of Rotterdam will have a share of 45 percent in hinterland transport by 2035, and this number is expected to continue growing beyond that time (De Langen et al., 2012). Today, a barge rotation consists of several terminal visits with loading and unloading operations at each terminal. The lack of coordination between terminal and barge operators (Van Der Horst & De Langen, 2008) and insufficient coordination (A. Douma et al., 2009) leads to long waiting times for barges in the port region and low utilisation of terminal resources (Konings, 2007). These problems have already been highlighted as a significant issue in the Port of Rotterdam.

In current practice, the HBT planning process is conducted manually (S. Li, 2016; A. M. Douma, 2008; Moonen et al., 2005). The barge operator makes appointments with assigned terminal operators individually. This is followed by a Yes-No strategy. This strategy is outlined in Figure 2.5. For example, a barge needs to visit two terminals to load and unload containers at each terminal. If the appointment at terminal two is granted and the appointment at terminal one is declined, the barge operator has to initiate negotiations by re-sequencing the terminal visit orders. This means the barge would visit terminal two first, then terminal one. This process will be executed repeatedly until all terminal operators reply with a positive answer.

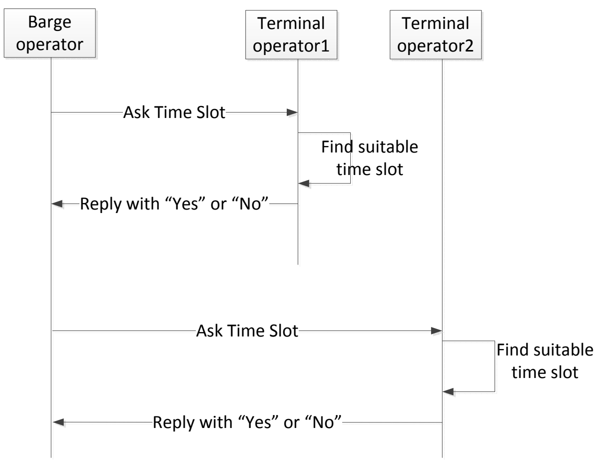

Figure 2.5: The current HBT planning process: a Yes-No strategy

2.3.2 Challenges in current HBT planning process

A challenge of information sharing

As discussed above, barge rotation plans are carried out manually. This process leads to ineffective information sharing. Consequently, infeasible plans are often created at the time of construction. A typical issue that arises is known as double-booked issue (Moonen et al., 2005). The double-booked issue occurs when a barge is expected to call at multiple terminals at the same time. A barge cannot call at two terminals at the same time, meaning the use of terminal capacity is wasted. As depicted in Figure 2.6, the barge is expected to call at terminal four and five in the period of 45 and 55, which is impossible. As a result, the call to either terminal four or five cannot be executed. Information sharing is a critical intrinsic factor that influences the interaction between barge and terminal operators. In turn, it leads to infeasible plans.

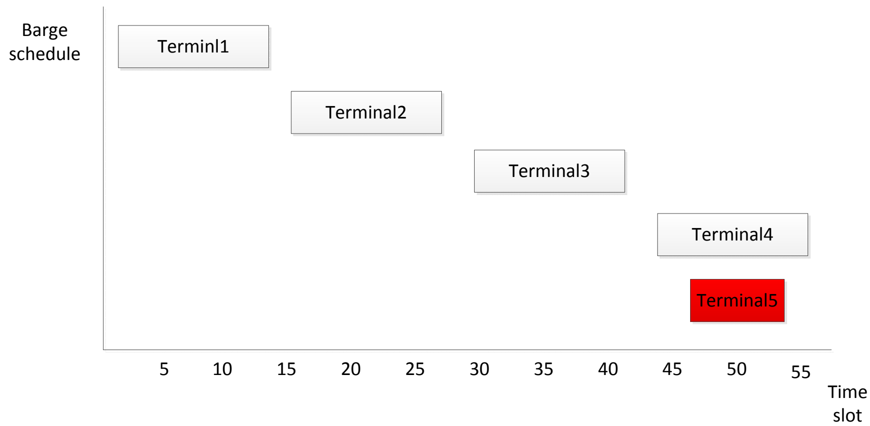

Figure 2.6: The double booking scenario

A challenge of rotation plan quality

Groningen (2006) introduced several key performance indicators of the HBT planning process. The primary objective of terminal operators is to make decisions about when to serve a barge call by considering its resource availability and efficiency. Resource utilisation is identified as the most important key performance indicator (KPI). The resource utilisation rate can be improved by increasing crane productivity and employing new workforce. In addition, Groningen (2006) also indicated that terminal operators prefer to have a queue in their berth location. This is because it can prevent idle times of equipment. As a result, an improvement of the resource utilisation level is expected. As shown in Figure 2.7, the operator of terminal one plans the calls from barges one to three directly after a sea vessel operation. This is a preferable scenario

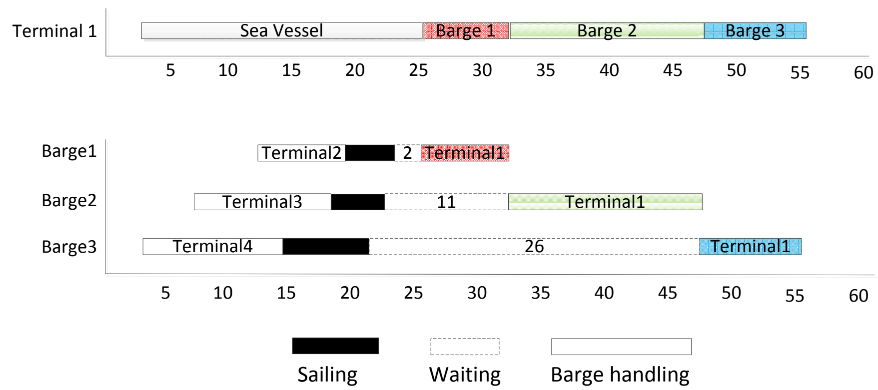

Figure 2.7: The barge planning from a terminal operator perspective

from the perspective of the terminal operator because it means resource utilisation is high. However, it can be found on the barge plan board that all barges will be expected to wait. Despite the sailing time from the previous terminal to terminal one, the waiting time is 2,11 and 26 for barges one to three. In this example barges have no control over the terminal decision-making process, and this can cause long waiting times for barges. However, to ensure barges do not have to wait before berthing usually leads to low terminal resource utilisation (UNCTAD, 1985). Similarly, as Groningen (2006) pointed out, if the terminal is more flexible, barge operators may take advantage of the terminals flexibility to adapt to its preference. For an efficient plan, a level of compromise between these two conditions is required.

In current HBT planning practice, barge operators have no information regarding the occupation status of terminals. Therefore, the rotation sequence is fully determined by terminal operators. Since the turnaround time of a barges are not the primary concern of terminal operators. As a result, long waiting times can be expected for barge operators. Moreover, neither barge operators nor terminal operators are willing to share their daily schedule or to competitors, due to privacy reasons. As a result, achieve effective coordination and collaboration among terminals and barge operators are difficult. (A. Douma et al., 2009).

2.3.3 Literature review of HBT planning problem

The HBT planning issue has received attention from the scientific community. A so-called barge hub terminal in or near the port of Rotterdam was proposed by Pielage et al. (2007). This will serve as a collection and distribution point for containers to and from the hinterland. The main goal of introducing the barge hub terminal (BHT) was to reduce the average terminal calls of barges by moving all required containers to a dedicated position. The same concept was also discussed in research conducted by Visser et al. (2007) which explored the concept of hinterland transport with an additional barge hub terminals by focusing on the type of transport system to operate the barges. These authors conducted an overview of new transport technologies such as automated trucks, multi-trailer system, automated trains and automated barge handling systems. They concluded that the innovative transport technologies would likely promote the efficiency of hinterland transport and support the concept of additional inland terminals with commercial viability. In the work of A. M. Douma et al. (2011), HBT alignment issues were addressed using a fully distributed MAS architecture .A. M. Douma et al. (2011) investigated how different levels of information exchange can affect the tardiness of barge rotations. It also introduced the concept of 'slack time' to increase the flexibility of terminal planning. A simulation game A. Douma et al. (2012) was developed to validate the effectiveness of the proposed system which helped the potential users to get an overview of the solution it provided. The system inherits the concept of MAS decentralised coordination between different parties. The final rotation plan will be made by barge operators according to the information(waiting profile) provided by terminal operators. The works of S. Li et al. (2014, 2016b, 2016a, 2017, 2018) proposed a solution to solve the barge rotation planning problem using distributed constraints optimisation in a distributed agent environment. The research aimed to generate an optimal plan for barge operators concerning minimum waiting time.

In addition to efforts from academic researches, several industrial projects have been developed, or are under development to improve the overall HBT performance in the industry. The first important initiative towards HBT planning problem in the port of Rotterdam was established in 2003. This is known as APPROACH (Schut et al., 2004). The project aimed to establish a decentralised system that generates a rotation plan off-line. The design of the system created a multi-agent communication environment where terminal and barge agents could negotiate to achieve a feasible plan. The system feasibility was validated in the work of (Moonen et al., 2005). In 2007, A barge traffic system web application [1] was put into practice in the port of Antwerp. It supported barge operators to make appointments with terminals according to their announcements about call size and estimated arrival time. The terminal operators then provide feedback regarding terminal planning and either refuse or agree to the plan. In addition, terminal operators can acquire real-time information regarding the position of the barge within the terminal. The supported functions included barge planning requests, consulting about lock planing, barge position tracking minimal planning inland barge. A new project called "Next Logic" [2] was launched in September 2013. This involved different parties including the port of Rotterdam authority, barge companies, terminal companies, shipper and the Dutch government. The project focused on reducing inefficiencies in inland container shipping by providing a centralised integral platform in which all of the related parties would join. Currently, the project is undergoing a feasibility study. It aims to develop a neutral and integrated planning platform where all relevant data and information is gathered together and then fed to a central decision

[1]BTS: HTTP://www.portofantwerp.com/apcs/en/BTS
[2]NextLogic: https://www.nextlogic.nl/

module called the 'brain'. The proposed platform represented a positive move towards centralised planning, although great efforts need to be taken to harmonise the interests for different parties.

In sum, the related literature about HBT planning problems can be categorised into two main areas. They are outlined below.

- To develop an information system to assist automatic information exchange and coordination. This stream of work focuses on preserving the autonomy and privacy of users by using a distributed system. However, the decision-making process is not fully optimised. In other words, the developed system does not consider planning performance as a primary goal.
- To design an analytical approach to optimise the objectives of the barge transport process by minimising barge turn around time. No research explores how to improve the terminal objective and the barge objective together, especially when a conflict of interests are involved during the planning process. Moreover, this stream of researches does not consider how data should be exchanged and how information flow should be coordinated in practice. As a result, this limits its applicability.

2.3.4 Summary

In light of the discussion above, to further improve the HBT planning process, this thesis addresses the DM process from two perspectives,

- First it examines the provision of an information platform for information exchange and sharing between barge and terminal operators. Section 3.2 introduces a MAS system as a tool to support information exchange and sharing.
- Second, it examines the provision of an intelligent decision-making approach to optimise the barge transport planning process. This leads to a more balanced result by taking into account the interests of both barge and terminal operators. Section 3.3 highlights that a meta-heuristics approach was selected to achieve intelligent decision making.

Chapter 4 proposes an integrated decision-making framework. This framework will be further implemented and assessed in Chapter 5.

2.4 A categorisation of operational decision-making problems

Operational decisions are short-term decisions. They concern the scheduling and execution of daily operations by considering possible disturbances such as weather, and uncertainties, such as equipment breakdown. A brief review of the decision-making problems at the operational level will be categorised and discussed in this section.

2.4.1 Scheduling and controlling of transport equipment

Transport scheduling aims to minimise the task completion time by assigning equipment as efficiently as possible. At the quayside, quay crane scheduling problems need to be addressed by finding appropriate moving sequence thus maximising crane productivity (Duinkerken et al., 2002; Lee et al., 2008). In addition, extensive work devoted to the routing and controlling of automated guided vehicles has considered ways to reduce both idle time and energy consumption (Xin et al., 2014; Duinkerken et al., 2006). In a dry bulk terminal, attentions are focused on efficient use of the belt conveyor network (Lodewijks et al., 2009). At the yard side, efficient work schedules are developed to ensure a high throughput of yard cranes by minimising the sum of the job waiting times (Ng & Mak, 2005; W. Li et al., 2009). Reduced traffic congestion is also considered in terminals with limited space and a busy schedule (Ng & Mak, 2004). Similarly, work can also be found in bulk terminals, such as stack-reclaimer scheduling (Hu & Yao, 2012) and quay crane scheduling (Bugaric et al., 2012).In addition to scheduling, equipment control is also essential. Van Vianen et al. (2012) developed a simulation model to evaluate the operational control strategy of equipment route selection in a dry bulk terminal. As modern terminals start to replace manual equipment with automated equipment such as automated lifting vehicle (ALV), and automated guided vehicles (AGV), proper control must be implemented. Typical problems include trajectory scheduling (Grunow et al., 2006), deadlock prediction and prevention of AGV (K. H. Kim & Hong, 2006). Other topics such as energy-aware control (Xin et al., 2014) and speed control of equipment (He et al., 2016) have also been studied. A comprehensive review can be found in Steenken et al. (2004); Stahlbock and Voß (2008).

Integrated scheduling of different types of equipment in multiple areas is also addressed at the operational level. Cao et al. (2010) proposed an integrated mixed-integer programming model for yard truck and yard cranes scheduling problems using a bender decomposition method. Xin et al. (2015) studied the interactions between crane handling and horizontal truck transportation inside container terminals.

At the operational level, performance is determined by how efficiently equipment can be utilised. To achieve that, most of the studies discussed above concentrated on assigning the equipment to a task. This leads to minimum completion time and maximum productivity using an operations research method. However, ensuring the optimal performance of PL equipment not only depends on operational planning and control of deployed equipment but also the maintenance of each asset involved in various processes (e.g., material handling). Therefore, equipment reliability should not be overlooked because proper maintenance is of vital importance.

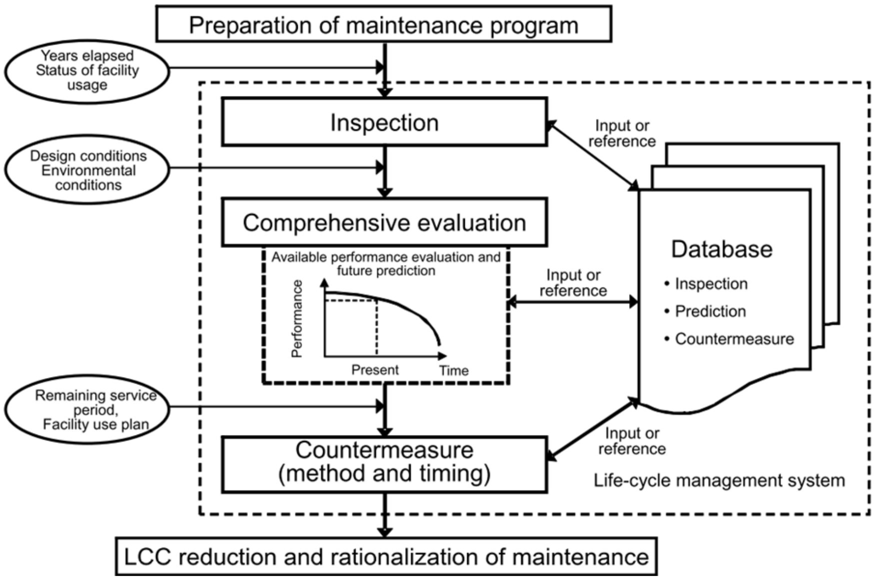

Figure 2.8: A life cycle management of port equipment, adapted from (ASEAN, 2011)

2.4.2 Equipment reliability assessment

Equipment should remain in service for a certain period, to properly maintain its functions. It is essential to conduct proper maintenance from the time their service starts. UNCTAD (1990) conducted a review on port maintenance practices. The review claimed that maintenance deficiencies were evident in all port activities. This was attributed to several factors such as lack of reliable diagnosis and inadequate data. In response, ASEAN (2011) presented a guide on strategic maintenance for port infrastructures by taking the concept of life cycle management into account. As can be seen in Figure 2.8, the maintenance task should follow a series of actions: preparation of maintenance, proper inspection, comprehensive evaluation of the inspection result and the creation of a maintenance plan. Mokhtari et al. (2012) addressed the issue of port risk management and highlighted the importance of proper maintenance of equipment.

As the related studies of equipment reliability assessment mainly focus on establishing maintenance policy on a strategic level, limited attention is given to the design of a system to support activities such as inspection, analysing and planning for specific equipment. Most monitoring and maintenance tasks are performed reactively. At the operational level, such a system is required to provide timely and accurate information about the condition of equipment and to propose an effective maintenance plan. In this thesis, the issue of reliability assessments of large-scale belt conveyor systems is considered. This is outlined in the following section.

2.5 Operational DM challenge: reliability assessments of belt conveyor system

Belt conveyor system (BCS) is widely recognised as an important mode of transport in continuous material handling. Compared with other modes of transport, such as trucks and trains, belt conveyors are capable of transporting large volumes of materials rapidly and efficiently in areas where road or railway do not exist. Figure 2.9 depicts the belt conveyor systems that are deployed in the EMO bulk terminal. At the EMO bulk terminal, nearly 50 km of conveyor belt provides the capacity to handle 140,000 tons of dry bulk materials every day. Since MHS is intrinsically associated with production flow, a reliable MHS operation would have a positive impact on transit time, resources usage, and service level (Vieira et al., 2011).

Figure 2.9: Belt conveyor system in a bulk terminal, (Courtesy from EMO bulk terminal)

2.5.1 Reliability assessments of BCS

Reliability of BCS is defined as the percentage of operational time the system's continuous performance is guaranteed (Lodewijks & Ottjes, 2005a). Since a BCS is deployed in an open and harsh environment, consequently critical components are permitted to deteriorate as system ages. As a result, the user should expect problems and costly delays. System failures such as component degradation (a tear on the belt) or operational abnormalities (misalignment) can cause costly downtime and cease whole operations if not detected quickly. Figure 2.10 depicts some typical failures of a BCS, such as a worn condition of an idler and a tear condition on the surface of a belt. To enable a better understanding of the BCS conditions and prevent these failures, an effective reliability assessment is needed. The functions of BCS reliability assessment include assessing the status of the conditions of component and system and proposing maintenance actions and plan based on the assessments.

Figure 2.10: The typical failures of a belt conveyor system. Figures from left to right represents a idler worn condition, a belt tear condition, a belt high temperature condition respectively

Effective reliability assessment relies on the accuracy of gathered data and the assessment method and knowledge. Traditionally, inspection of belt conveyors is carried out by companies which use belt conveyors. Followed by the expert advice on the condition of BCS, maintenance activities are planned such as replacement or repair of BCS components. This way of reliability assessment has several disadvantages. First, inspection of BCS replies on human-effort which is time-consuming. Also, an accurate assessment can only be made from an experienced human-expert who should be well-trained personnel. There is no automated intelligence involved during the assessment process. If the inspector is not well trained or different inspector performs adjacent inspections on the same part, the chance of an inconsistent result is significantly increased.

Nowadays, the advances of sensor technologies enable an automated BCS monitoring. Compared with the traditional way, BCS monitoring is automated. Table 2.1 presents a list of parameters, components, and their corresponding sensor technologies applied for BCS monitoring. ICT applications can be developed to support reliability assessment of different components. It can be achieved by parsing the obtained measurements, enhance the quality of the information, a predict the component status based on expert knowledge.

Table 2.1: Parameters, components, and corresponding sensor technologies for BCS measurement, derived from Pang (2010)

Parameter	Component	Sensor
Belt condition	Surface Steel cable	visual inspection conductive detection
Speed	Belt Brake disk Motor	Optical/magnetic encoder Magnetic RPM pickup sensor
Torque	Motor shaft Brake shaft Pulley shaft	Torque meter
Force and tension	Take-up Belt Frame	Strain gauge
Vibration	Pulley Idler roll Rotation drive	Acoustic sensor, Accelerometer
Position	Belt misalignment Take-up displacement	Alignment switch Optical encoder
Temperature	Ambient Material Brake disk Pulley shaft Motor	Thermocouple, Infrared temperature sensor

2.5.2 Challenges of reliability assessments of BCS

According to Pang (2010), current reliability assessments of BCS are carried out in response to the abnormality and failure of individual components. The assessments, such as failure alarms, are straightforward. Reliability assessments involve little or no predictive assessment. Meanwhile, Pang (2010) also emphasised that effective reliability assessments of BCS should be able to identify subtle changes in operation that may indicate a mechanical or electrical problem is starting to develop. To achieve the goal, reliability assessments should be extended from a component level to a system level, including integration of measurement data and integration of condition assessment methods.

Effective reliability assessment of BCS demands automated decision-making. Currently, the BCS is still diagnosed by experienced people with in-depth training (Pang, 2010). The monitored data and corresponding knowledge for deciding what maintenance activities should be carried out are mostly not stored. As a result, they are not reusable. Moreover, the combination of individual component assessment requires the integration of different assessment knowledge and data. There is no ICT support to allow a system level reliability assessments of BCS.

The data used for diagnosis can come from different sources. For example, this can include visual inspection, sensor measurement, and expert judgement. Some of the traditional monitoring systems present the monitored data in a manner of series lines on a length of chart paper or in a visual graphics format, e.g., conductive monitoring and camera scanning. There is a lacks information integration to convert different data sources into meaningful information on which the monitoring process can be based (Lodewijks & Ottjes, 2005a). For instance, an inspection of a breath tear (e.g., tear width, depth and position) on a belt is meaningless unless it clearly identifies the damage level (the impact of the damage on the belt conveyor's performance) and results in proper maintenance decisions (whether the belt should be replaced or not).

Complex systems will generate large amounts of data. Different categories of data need to be associated with each other in order to make integrated decisions. So far,there is no such support available.

In sum, to further improve reliability assessments of BCS, an integrated DM system is needed. The system needs to (1) integrate different data sources and transform into meaningful information; (2) enable information exchange and sharing to achieve coordinated DM.

2.5.3 Literature review of reliability assessments of BCS

Component monitoring and diagnosis

A stream of researches is conducted to improve the reliability assessment of a single component. One perspective is to explore condition measurement technologies. Fourie et al. (2005) developed an X-ray scanning system for imaging defects in the cords of steel-cord-reinforced belting. Pang (2010) proposed to using a matrix of Hall effect magnetic sensors to measure main operational parameters of BCS, such as speed, belt tension, and wear. De La Cruz et al. (2006) designed a RFID-based system to measure the temperature of BCS idlers. X. Liu et al. (2016) proposed to use a tactile pressure sensor to measure the pressure distribution on a running conveyor belt. Another perspective is to develop diagnostic methods to support assessment decision-making of a single component. Lodewijks and Ottjes (2004) designed a fuzzy-logic approach to assess the damage level of a belt tear condition and proposed maintenance actions by considering the current tear damage level and its propagation speed. Pang and Lodewijks (2005) developed a Bayesian inference method to assess a BCS long braking time condition. Jeinsch et al. (2002) developed a method to detect a seeded fault of a drive unit by using a model-based approach through measurements of driving torque and belt speed.

Integrated monitoring and maintenance

In 2005, Lodewijks and Ottjes (2005b) introduced the concept of automated maintenance and control of BCS using intelligent monitoring systems and ICT technology. With the support of an expert and knowledge system, it strives to optimise the operational performance and reliability of BCS at three levels: component, equipment and system. Followed by the proposal of automated maintenance, Pang (2010) put forward the concept of intelligent and integration control of BCS which enables the understanding of the operational and component condition. It utilises different artificial intelligence (AI) technologies and MAS to facilitate the monitoring and diagnosis of a BCS (Pang & Lodewijks, 2005, 2012). Figure 2.11 presents a MAS architecture developed by Pang and Lodewijks (2012). With the MAS architecture, agents that monitor and diagnose individual component are organised into cooperative communities to achieve optimum reliability control of BCS.

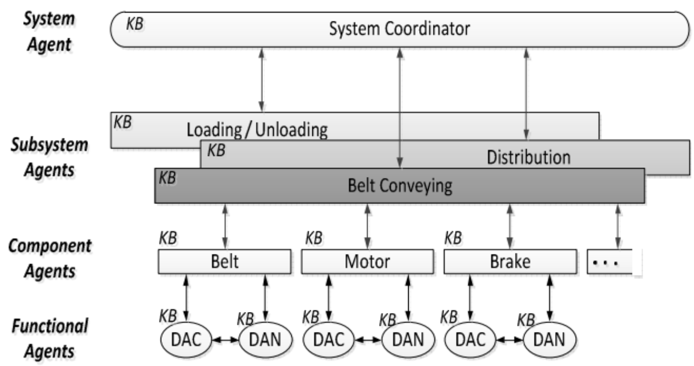

Figure 2.11: A multi-agent system architecture for BCS condition monitoring, derived from Pang and Lodewijks (2012)

In sum, existing researches can be divided into two parts:

- Develop measurement technologies and assessment methods for a single component. This steam of researches focused on a specific component of BCS. As a result, no integration aspects are considered, such as information integration, coordinated decision-making to achieve a system-level assessment.

- Propose an integrated system to achieve reliability assessment at a system level. This stream of research take efforts to develop an integrated system towards system level assessment. The system developed by Pang (2010) aims to provide a generic framework that can be used to assess different conditions of BCS. However, in the proposed framework, the interoperability issues are not properly considered. For example, how to mine different data sources and transform it to meaningful information, in which way information and knowledge can be integrated, how information is exchanged between knowledge-based and business logic (e.g., agent intelligence). Consequently, the challenges discussed in section 2.5.2 has not been fully addressed.

2.5.4 Summary

In light of the discussion above, to further improve reliability assessments of BCS, this thesis addresses BCS reliability assessment problem from two perspectives,

- First, it will provide an information integration platform. This platform should be able to automatically acquire and integrate system data into a generic platform for decision making. Section 3.2 introduces a MAS as the tool to support information exchange and integration.
- Second, it will provide an intelligent decision-making approach to improve the interoperability of the decision-making process.This should be able to mine different information sources to find internal relationships and generate an understandable diagnostic of the systems condition. Sections 3.4 introduces a context-aware system and ontology as the tool for supporting information mining and intelligent decision making.

Chapter 4 proposes an integrated decision-making framework. This framework will be further implemented and assessed in Chapter 6.

2.6 ICT applications in Port Logistics system

In section 2.2 and 2.4, the decision-making problems at the tactical and operational level are reviewed. To put the decisions into practice, the development of an effective ICT system is indispensable. In this section, the developments of ICT in the current PL system are reviewed. First, a brief review of the technologies concerning information acquisition is given in section 2.6.1. Second, section 2.6.2 provides the current development of information systems in PL from both academic and industry standpoint. Finally, a discussion of existing architectures and future developments are outlined in section 2.7.

2.6.1 Key enabling technologies

The development of the PL system relies on the use extensively on advanced technologies. The introduction of new ICT technology has enabled a high degree of automation in ports. This has led to more efficient and flexible operations. The following section gives a brief review of some key available technologies that could be beneficial for the development of the PL system.

Radio frequency identification

An RFID is a contactless automatic identification technology that uses radio frequency waves to transfer data between a reader and a tag. It consists of an antenna, tag and radio frequency module with a decoder. The RFID is widely used in the transport and logistics industry. Typical tasks include identifying, tracking, locating, and monitoring cargo flow and vehicles movement (De La Cruz et al., 2006; Ngai et al., 2008). Although the use of RFID in the PL domain is still in its infancy phase (Heilig & Voß, 2016), several developments started to merge. Typical usages include cargo tracking (Shi et al., 2011), operational control (Siror et al., 2011; Min et al., 2013), and security and safety (Pang & Lodewijks, 2011; Kadir et al., 2016).

Smart devices

Smart devices encompass a wide spectrum of intelligent systems. These include ubiquitous sensors, mobile phones, global navigation system and so on. The use of smart devices enables users to receive the necessary information that monitors the surrounding environment (Lun et al., 2008). Wang (2012) designed a wireless sensor network for safety monitoring of a seaport over the IEEE 802.15.4 Zigbee network. Abbate et al. (2009) used a wireless sensor network (WSN) to keep track of container location in real time. Xu et al. (2014) provided a comprehensive review of the use of WSN to assist environment monitoring in the marine field. The mobile devices, such as smartphones and tablet, are normally equipped with powerful processors and Internet access. This provides various opportunities to enhance the mobility and flexibility of the information system. For example, Heilig et al. (2016) proposed an architecture for truck routing in a container terminal, which mobile devices act as a gateway for data forwarding. Pang and Lodewijks (2013) developed a mobile inspection tool for condition monitoring of large-scale equipment in a dry bulk terminal. Despite the design and using of smart devices, other studies have focused on how to integrate information gathered from these devices. Ngai et al. (2011) proposed a context-aware middleware for integrating data generated by a ubiquitous sensor network for tracking the movement of trucks in a container terminal.

Automated identification and positioning

Enabled by advanced identification and positioning technologies, movable objects (i.e., vessels, cargo, vehicles, and equipment) can be tracked, monitored and positioned. Such systems enhance the transparency and visibility of the flow of goods and equipment. At sea, global positioning systems (GPS) are widely used for tracking vessel movement in the port. The position data is essential for forecasting the estimated arrival time of a vessel. As such the operations in the terminal can be efficiently coordinated. An optical character recognition system is used to provide automatic recognition of container numbers and truck license plates at the gate of a terminal (Chao & Lin, 2010). A voice recognition system is used to transform the instructions of terminal operators into electronic data stored in computers for message recording (Kia et al., 2000).

Communication standards and protocol

Despite using advanced technologies for real-time data acquisition, how data is exchanged is also essential. This is known as communication standards. Electronic data exchange is recognised as the most important standard to structure and exchange data for commercial administration and commercial transactions. UN/EDIFACT [3] defined EDI message types for data exchange related to port operations such as berth management. Since its introduction in the 1980s, EDI is mostly used in major ports. With the development of Internet technology, another standard known as Extensible Markup Language (XML) has become popular because it is both flexible and cost effective. Another important aspect is the communication protocol used for data routing and relaying in a sensor network, such as 6LoWPAN, WPAN and so on. A comprehensive review can be found in Lodewijks et al. (2016).

In this section, key enabling technology such as sensor technology and communication standards are briefly reviewed. A more in-depth review can be found in (Kia et al., 2000; Heilig & Voß, 2016).

2.6.2 Information systems in PL

Enabled by the different technologies discussed in section 2.6.1, data generated from PL operations can be measured, collected, and transmitted in real-time. To transform data into useful information, an integrated information system is required. In this section, a broad review of information systems, which are widely used in current PL practice, is presented. Figure 2.12 categorises the information system based on the scope of applicability, i.e., the user group, the addressed operations, and the area of usage. The following section reviews information systems based on this categorisation. Efforts from both academia and industry practice are considered.

Port community system

A port community system (PCS) is defined as an information portal or hub that seamlessly connects various port-related actors electronically into large-scale global transportation networks. It is an inter-organisational centralised information system that integrates heterogeneous information, across multiple actors, systems, and processes, to a single location. Its major role is to enable an efficient flow of information that

[3] UN/EDIFACT: https://www.unece.org/cefact/edifact/welcome.html

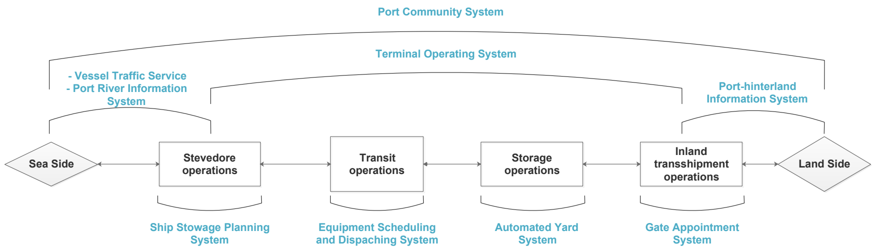

Figure 2.12: A categorisation of information systems in a PL system

relates to PL activities such as customs, import and export declarations, and dangerous goods declarations, thus enhancing administrative and logistics processes. The usage of a PCS can be found in major ports around the world, a review of current developments of PCS is given in Table 2.2. In the academic domain, research mainly focused on the applicability and potential benefits of adopting a PCS in a port. Table 2.3 lists several research initiatives related with PCS. A comprehensive review of the PCS and its main functions can be found in Posti et al. (2011); Carlan et al. (2016).

Table 2.2: Port community system in industry

PCS	**User**	**Established**	**Main purpose**
PortBase [4]	Port of Rotterdam	2004	Access coordination
Dakosy [5]	Port of Hamburg	1981	Access and port process coordination
TradeXchange [6]	Port of Singapore	2007	Access coordination and paperless communication
TradeLink [7]	Port of Hong Kong	1988	Port process coordination and paperless government application
Fright information real time system for transport (FIRST) [8]	Port of New York and New Jersey	2001	Reduce the number of data exchange among multiple participants

Terminal operating system

While a PCS considers information exchange and integration at the logistics level, the actual operations inside a terminal are managed by a terminal operating system (TOS). A TOS normally consists of several subsystems. Each subsystem is in charge

[4] PortBase: http://www.portbase.com/en/
[5] Dajosy: http://www.dakosy.de/en/
[6] Tradexchange: http://www.tradexchange.gov.sg/tradexchange/index.html/
[7] Tradelink: http://www.tradelink.com/hk/eng/index.html/
[8] FIRST: http://www.firstnynj.com

Table 2.3: Studies on the port community system

Authors	**Aim of study**	**Application level**	**Industrial background**
Gordon et al. (2005)	Evaluate how IT contribute to the competitiveness of Port logistics	Strategic level	Port of Singapore
Posti et al. (2011)	Evaluate the suitability of a PCS to the Finnish Port	Strategic level	Port of Finland
Srour et al. (2008)	Lift cycle development of a PCS	Strategic & Technical level	Generic
Aydogdu and Aksoy (2015)	Analysing the benefits and the cost of deploying a PCS	Strategic level	Port of Turkish
Keceli et al. (2008); Milà (2007)	PCS adoption analysis	Strategic level	Multiple port
Aksentijević et al. (2009)	Information security of PCS	Technical level	Generic

of specific decision-making processes. For example, a ship stowage planning system is responsible for assigning berths to ships and planning the stowage of containers. The TOS exhibits the characteristics of an ERP system which provides a set of applications to collect, store, process, analyse and distribute information in order to optimise the operational efficiency and handling capacity. The functions of a TOS can be categorised into two areas. First, the tactical level is planning systems. The planning systems include berth planning, yard planning, resource planning, rail operation planning, and ship operation planning. Second, at the operational level, Real-time operation scheduling takes places.This includes real-time scheduling, transport equipment dispatches, and yard resource allocation and positioning. The TOS has been extensively deployed in major ports. Table 2.4 lists some major TOS manufactures in the market. As shown in Table 2.4, the commercial TOS is widely used in major ports around the world. It can be seen that alongside the development of ICT technology, a trend of migrating the TOS into the cloud by providing the TOS software as a service has emerged (Heilig & Voß, 2016).

Table 2.4: Commercial terminal operating system providers

TOS	**Company**	**Platform**
CATOS [9]	Total Soft Bank (Korea)	J2EE; .Net
Navis SPARCS N4 [10]	NAVIS (USA)	J2EE
CITOS [11]	Port of Singapore (Singapore)	Unix/Linux
nGen [12]	Port of Hong Kong (Hong Kong)	J2EE
XVELA [13]	XVELA (USA)	Cloud

As shown in Figure 2.12, there are several subsystems executed under the umbrella of TOS. A ship stowage planning system is used to determine the berth position when a ship arrives. An equipment scheduling and dispatching system is used to assign trucks and AGVs to transport cargoes from quayside to yard side for storage. An automated yard system considers the operations in yard area which include yard crane allocation, yard location allocation, yard monitoring and so on. A gate appointment system is used to control the number of vesicles entering into the port during a certain period of time. Overall, different subsystems are working in parallel to support complex PL operations. A comprehensive review of the TOS developments can be found in (K. H. Kim & Lee, 2015).

Vessel traffic service

A Vessel traffic service (VTS) is a maritime traffic monitoring system that has been used in port or congested waters. The primary task of a VTS system is to monitor the movements of vessels within the VTS area and provide vessel operators with real-time information regarding current and future events, which ensures the safe passage of vessels. The VTS is a generic concept, its implementations involve various technologies (GPA, IOT), and consist of multiple modules such as automatic identification system and AIS. Trained officers monitor the displayed information for safety and security risks to pre-empt suspect vessels and activities. The research topics of VTS include vessel tracking, collision avoidance and onboard communication. A summary is provided in Table 2.5.

Table 2.5: Researcher on the vessel traffic system

Scope of research	**Publications**
Vessel monitoring and tracking	S.-J. Chang (2004); Tu et al. (2017); Xie et al. (2011)
Collision avoidance	Simsir et al. (2014); Kao et al. (2007); Q. Li and Fan (2011)
On-board communication	Y. Kim et al. (2009); Manoufali et al. (2013)
Human factors	Xie et al. (2011); Darbra et al. (2007)
System architecture	Park and Bang (2016)

With the availability of more accurate information about vessel location and port traffic, the activities such as vessel scheduling and terminal operations can be further improved (Heilig & Voß, 2016). The port river information system (RIS) is a specific type of VTS. The RIS is used to improve communication and coordination between actors

[9]CATOS: http://www.tsb.co.kr/RBS
[10]Navis N4: http://navis.com/get-more-n4
[11]CITOS: https://www.singaporepsa.com/our-commitment/innovation
[12] nGen: https://www.singaporepsa.com/our-commitment/innovation
[13]XVELA: https://www.xvela.com/

involved in the process of transport over the sea. A brief review of the available RIS is presented in Table 2.6. Its major task is to collect, analyse and distribute information to navigate vessels in a busy and confined waterway area both efficiently and safely. Information is provided to relevant actors include the geographic position of vessels, status of mooring service and tugboats, water level prediction and weather conditions, and docking and berthing planning.

Table 2.6: River information system in industry

RIS	Users	Establish	Major purposes
Port river information system Elbe (PRISE)	Port of Hamburg	2014	Berth planning and registration at the terminal; status information tracking of ship position on the river Elbe
Harbor master management information system (HamMis)	Port of Amsterdam and Rotterdam	2015	Traffic guidance; traffic planning, safety and incident control
Barge traffic service (BTS)	Port of Antwerp	2007	Consult locking plan; position request; terminal planning

Port-hinterland information system

The port-hinterland information system is used to improve the efficiency and visibility of cargo movement and transhipment between terminals and inland destinations. This involves the development of specific information systems to improve the exchange of information between different actors, reduce administrative burden, and coordinate operations related to hinterland-transport. Table 2.7 provides a list of ICT applications that have been developed and applied in some port.

Table 2.7: Port-hinterland information system industry use

Information system	Users	Transport mode	Major purposes
transPORTrail [14]	Port of Hamburg	Rail	Cargo tracking; train movement monitoring; loading/unloading planning
Nextlogic [15]	Port of Rotterdam	Barge	Barge rotation planning; call optimisation; actors coordination
Intermodal planner[16]	Port of Zeeland	All modes	Intermodal route planner

[14]transPORTrail:http://www.hamburg-port-authority.de/en/port-customers/port-railway/tpr/Seiten/default.aspx

[15]Nextlogic: http://www.nextlogic.nl/

[16]intermodal planner: https://intermodalplanner.eu/Planner/

2.6.3 Summary

Based on the extensive review presented in this section, it can be concluded that ICT applications play a major role in the development of PL. Different ICT systems have been developed to support tasks including information exchange, centralised planning and control, operation monitoring, and tracing and tracking. Most DM problems discussed in section 2.2 and 2.4 have corresponding ICT support. For example, the tactical and operational decisions related with terminal operations (discussed in sections 2.2.2 and 2.4.1) are made via a TOS (discussed in section 2.6.2). By translating the key findings and algorithms from operations research into a hybrid DM engine, the TOS continuously monitors, integrates terminal conditions and provides optimised decisions for terminal operations.

However, the challenges identified in sections 2.3 and 2.5 have not receive sufficient supports from existing ICT applications.

- Coordination planning of barge transport: A litter IT system developed for hinterland barge transport planning. The only available system is the BTS developed in the Port of Antwerp (see Table 2.6), which is used for vessel tracking and positioning. Currently, a project is under development attempting to solve the coordinated barge planning problem, but the benefits are so far unknown (e.g., NextLogic in the Port of Rotterdam, see Table 2.7).
- Reliability assessment of large-scale equipment: Given the review above, it can be found that most existing IT applications, which are related to equipment management, focus on tracking, navigating and assigning of equipment. The primary goal is to maximise the machine utilisation level. Currently, there is no dedicated application used to serve the purpose of equipment reliability assessment. Human-based inspection and diagnostics is still the method of assessment used in current practice.

2.7 Requirements of ICT developments

Figure 2.13 presents a summary of two DM challenges. As shown in the figure, collaborative planning requires a DM system to support human-human interaction. The DM system is required to answer questions include when to serve which barge for terminal operators, and when to call at which terminal for barge operators. The reliability assessment requires a DM system to support human-machine interaction. The DM system is required to answer questions such as what goes wrong for the assessed equipment, when to repair it and how to perform the maintenance actions.

- At the tactical level, the implementation of collaborative planning has two main requirements. First, it requires an information platform to enable easy information exchange, integration and automated coordination between parties. For

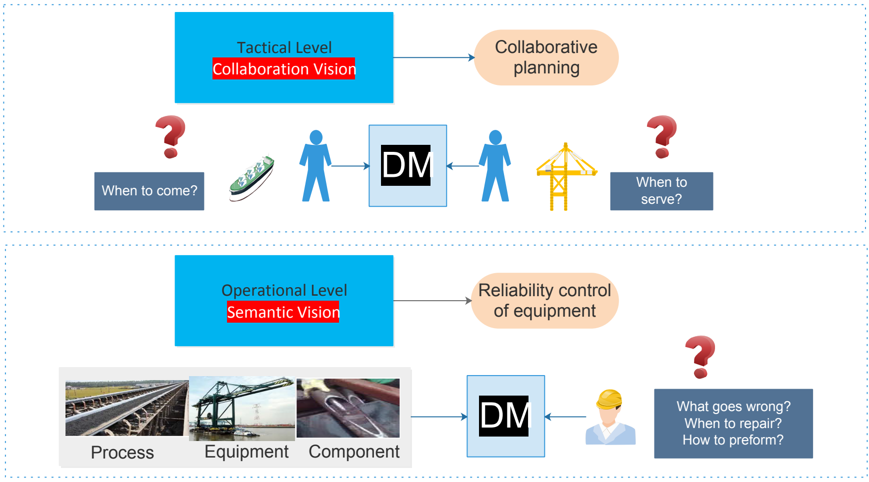

Figure 2.13: A summary of DM challenges at each decision level

example, hinterland barge planning involves terminal and barge operators. Currently, the information exchange is done via telephone and email. These methods are error-prone and inefficient. Secondly, it requires a solution approach to harmonises the conflicts of interests and enables collaborative planning. In hinterland barge transport, the terminal and the barge operators have different interests. The terminal operator is most concerned about the resource utilisation while the barge operator most concerned about total turn around time. Balancing the interests of multiple parties is critical to successful communication. Currently, barge rotation plans are made without fully considering the objectives of each party. As a result, the DM process is unsatisfactory.

- At the operational level, the challenge posed by reliability assessment has two main requirements. First, it requires an information system to process and interpret gathered information and deliver it to the right person at the right time. For example, the traditional BCS monitoring presents the monitored data on a paper. As a result, it is difficult for maintenance to interpret the information and take proper maintenance actions. Second, it requires integration of information that represents the state of components, sub-systems and systems. Mining and integration of large amounts data can be difficult. Currently,there is no DM systems that can store the monitored data and corresponding knowledge and automate the DM process. For example, in current BCS maintenance, a decision is mostly made by human experts via visual inspection. This suffers low accuracy.

By examining the requirements, it can be found both challenges share the same requirements from an ICT perspective. These requirements are summarised as follows:

- Provide an information platform for information exchange and sharing: An information platform is needed to improve information exchange and sharing between multiple parties at a tactical level. Similarly, an information platform is needed to assist information flow from ground measurement to end users so that human-machine interaction is enhanced at the operational level.

- Provide an approach for integrated decision making: A collaborative planning algorithm is required to harmonise the interest of all parties to ensure a global optimum solution can be provided. Similarly, an approach for information integration and knowledge management is needed to enable integrated reliability assessments of PL equipment.

Based on the requirements outlined above, two main elements are specified as follows:

- Middleware: a middleware is defined as a software layer or a set of sub-layer interposed between the technological and the application levels. It forms a bridge between the technology and its application. In this research, a middleware is needed to support information integration, exchange, and sharing among different entities in the PL system.
- Intelligent DM approaches: Ideally, the DM system should act autonomously by gathering and processing information, proposing a possible course of action and be capable of evaluating the consequences of that action. To achieve this, intelligent approaches need to be designed and integrated with the middleware to enable the DM system to behave intelligently. At the tactical level, a DM approach is required to assist barge planning. At the operational level, a DM approach is required to support information mining and equipment condition assessment.

The selection of each element is presented in Chapter 3. In Chapter 4, an integrated DM framework is proposed to integrated the selected elements into a generic framework.

2.8 Conclusion

This chapter provides the answers to sub-question 1.1 listed in Chapter 1. First, it presented an overview of the PL system. The DM processes at the tactical and operational levels were also outlined. Based on the review in this chapter, two DM challenges in the current PL system were identified, hinterland barge transport planning and reliability assessments of BCS. A discussion of tactical decision revealed that the lack of coordinative planning is a challenge faced by PL actors. A case of hinterland barge transport issues was used to illustrate this challenge. The discussion of operational decision identifies a further challenge faced by the current PL system. This is the challenge posed by the maintenance of port equipment. A case of reliability assessment of large-scale belt conveyor system in bulk terminals was used to illustrate this challenge. These challenges are caused by insufficient DM support. Therefore, a more effective use of ICT solutions was suggested as a method to address these challenges.

Second, a review of ICT applications in the current PL system was presented. It was concluded that ICT is applied extensively in practice to support different PL DM processes. In contrast to other DM problems (e.g., DM problems in terminal), the two challenges have so far not been fully solved by ICT solutions.

Finally, by assessing the requirements of both challenges, it can be concluded that two elements are required. First, a middleware is required to support information exchange and sharing between different entities. For example, this can be used to support interaction between terminal and barge operator for barge transport planning, support interaction between maintenance operation and machine for reliability assessment. Also, an intelligent approach is required to support intelligent decision making. For example, an approach is required to assist terminal and barge operator to create barge rotation plans and to derive equipment conditions from measurements.

In Chapter 3, the middleware technology and the intelligent decision-making approaches are selected. Chapter 4 initiates a new conceptual ICT architecture. In Chapter 5 and 6, the proposed framework will be instantiated and applied to address the decision-making challenges discussed in this chapter.

Chapter 3

Selections of middleware technology and intelligent decision-making approaches

Chapter 2 presented an overview of the PL system and identified the present DM challenges. This included challenges faced by hinterland barge transport planning and reliability assessments of large-scale BCS. It was concluded that ICT has the potential to address both challenges. This can be done using a middleware and intelligent decision-making (IDM) approaches. In this chapter, the middleware technology and the intelligent DM approaches are selected. Figure 3.1 presents an overview of the technology selection. A multi-agent system (MAS) will be selected for the middleware technology. A detailed introduction of the MAS is given in section 3.2. The IDM for each challenge is selected in sections 3.3 and 3.4. Section 3.3 outlines the choice of meta-heuristics for supporting the barge transport planning. Section 3.4 introduces the context-aware system and ontology for supporting the BCS reliability assessments. Finally, a conclusion is drawn in section 3.6.

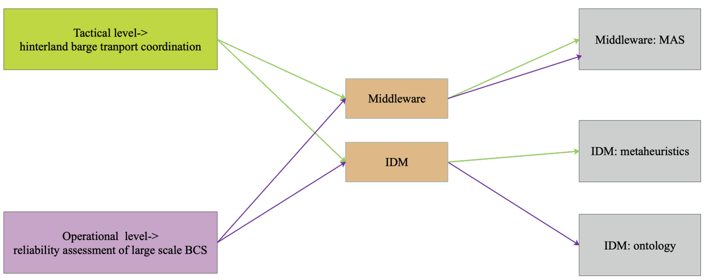

Figure 3.1: A summary of technology selection for each challenge. The green line represents the tool selected for the tactical challenge and the purple line represents the tool selected for the operational challenge

3.1 Selection of middleware technology

In this section, the technology for implementing the middleware is selected. In section 3.1.1, both functional and non-functional requirements are proposed as the selection criteria. Based on these requirements, Section 3.1.2 compares three major technologies. As a result of comparison, the agent based approach will be selected. In section 3.1.3, the applicability of using an agent-based system will be further discussed.

3.1.1 Requirements of middleware selection

The middleware is used to provide a platform to support interactions between different entities. To select the appropriate technology for implementing the middleware, three functional requirements should be fulfilled:

- Collaboration and coordination: Collaboration is an important criterion for efficient DM in the PL practice. For C1[1], the collaboration considers the goal of working together in mutually defined ways in which participants are actively sharing information and jointly making decisions in order to achieve global optimal solutions. For C2[2], the collaboration emphasises on joint DM among individual monitoring/diagnosis system. Given the interdependency of the subsystems in transport equipment, instead of focusing on single components or processes, the reliability assessment considers the overall conditions of the system requires the DM system to establish collaboration between components, functions, and systems.

- Information exchange and sharing: Information exchange and sharing imply an ability to interact with the environment and other entities. For C1, it concentrates on the interactions between joined parties, such as terminal operators and barge operators. C2 focuses on the interactions at functional, components, and system level. Via information exchange and sharing, an integrated decision can be made. To achieve information exchange and sharing effectively, a well-defined and robust communication channel and a clear protocol are required.

- Autonomy: Autonomy emphasises on the control over individual responsibilities without intervention from others. For C1, the core of collaboration is to harmonise the potential conflicts of interest and reach a consensus via the DM system. In the process of negotiation, the DM should be able to consider the interests of each joined party and make decisions. Thus, autonomy can be preserved. For C2, each module (component, system, user) has its own specific function which operated autonomously. To maintain autonomy, the DM system is required to have control based on the modules state and conditions.

[1] C1: represents the collaborative planning challenge in this section
[2] C2: represents the reliability assessment challenge in this section

Besides, several non-functional requirements should be considered for middleware selection. These include usability, flexibility, adaptability, compatibility, modularity and fault tolerance.

3.1.2 A comparison between available technologies

Three technologies can be found in the literature: object-based approach (D. Liu & Stewart, 2004), the agent-based approach (Jennings, 2000) and the expert system (Bohanec & Rajkovic, 1990). A brief introduction of each technology is presented is presented as below.

- The object-based approach (OBA) uses an 'object' to model an entity or a system. Object-based application primarily focuses on services and functional module design. The entity or system modelled by an object is normally passive rather than proactive. The relationship between objects is static and cannot be changed once deployed. In addition, the objects in the OBA are organised centrally under the control of other components.
- The expert system (ES) can be defined as a software mechanism used to store and represent knowledge acquired by experts. It consists of a knowledge base, an inference engine, and an interpreter. The ES is typically isolated and self-contained. Little attention is paid to support interactions with other ES or external systems. Besides, the design of ES largely relies on expert knowledge, meaning the self-intelligence abilities of an ES are low. The knowledge stored in ES cannot be evolved on its own.
- Agent-based approach (ABA)is defined as a component of software which is capable of acting exactly in order to accomplish tasks on behalf of its user. Compared with OBA, the design of ABA is more flexible. The relationship between agents is more complex and dynamic. The agent can adapt to the changing environment as a result of its higher level of self-intelligence. Besides, the agent has control over its behaviour. Therefore, it is known as an autonomous system. Compared with ES, the interactions in ABA are more emphasised.

To select the appropriate approach, the following section will examine these three major functional requirements for each approach as follows.

Collaboration and coordination

The use of ES typically operates at symbolic and knowledge level. It is efficient in solving individual DM issues but is normally isolated and self-contained (Guimaraes, 2007). This could be problematic in the development of collaboration mechanisms.

The ABA is favorable in a situation where collaboration is required (Aylett et al., 1998). For the ABA, agents actively cooperate by sharing information and knowledge. A consensus is achieved via a collaboration mechanism embedded into the intelligent agent. Concerning OBA, the entity or system modelled by an object is normally passive (Khozium et al., 2013). The relationship between objects is static. This means they have been defined and fixed before the deployment. Consequently, enable collaboration in a dynamic and complex environment can be difficult by using this approach.

Information exchange and sharing

ES is recognised as isolated and self-contained software. The communication module provided by an ES normally concentrates on the interactions between the system and end users (e.g., human interface) (Gutierrez & Branch, 2011). Concerning OBA, the messages exchanged among objects can be seen as a basic form of interactions (Aylett et al., 1998). In a dynamic environment where complex and intensive communication exists, a more complex degree of interactions is needed to enhance interoperability and efficiency. Concerning ABA, communication among agents is supported by an agent communication language (ACL). This enriches the level of exchanged messages significantly. The messages exchanged or shared could range from a simple string to complex content (e.g., information can be contained about both the sender and receiver).

Autonomy

The OBA and ES normally act directly to their environment. In this case, actions are based on clear rules/explicit use of knowledge. They are effective in the sense that they can complete individual tasks via pre-defined intelligence. However, the use of ES and OBA lacks the dynamic intelligence to adapt to a changing environment. For a fast-changing and complex environment, it is infeasible to enumerate all cases in the design phase, and autonomy may not be guaranteed in such a complex environment (Guimaraes, 2007). An advantage of using an ABA is that an agent can maximise the preservation of autonomy. As a result of the changing environment, the agent can choose appropriate actions based on its interpretation that maximise its interests and profit. That is a primary reason for the widespread application of the ABA in a negotiation intensive environment where conflicts of interests between the parties involved. For example, this is often the case in such as supply chain negotiation (Hernández et al., 2014).

Given the above discussion, the criteria for selecting the suitable DM approach is summarised in Table 3.1. The result indicates that the properties of ABA are suitable to meet the major requirements discussed previously.

From a system architecture design perspective, two types of system should be considered namely the centralised system (Duchessi & Chengalur-Smith, 1998) and the dis-

Table 3.1: Summary of DM approaches

DM approaches	F1: collaboration	F2: information exchange and sharing	F3: autonomy
OBA	Fair	Fair	Fair
ES	Poor	Poor	Fair
ABA	Good	Good	Good

tributed system (Schneeweiss, 2012). Further, a server-client system and a multi-agent system and are chosen to represent each type of system. Based on the non-functional requirements, the suitability of these two types has been compared. This comparison is summarised in Table 3.2.

Table 3.2: A comparison between MAS and client-server architecture (based on Aylett et al. (1998); Weyns and Georgeff (2010))

	MAS	CS
Scalability	Highly scalable, agents can be deployed in a physically distributed environment	The scalability could be achieved by adding more servers.
Flexibility	Flexible to meet different demands. Functions or services can be added on demands	All services and functions are implemented before deployment, any on-demand changes could be difficult to implement.
Modularity	Highly modular, Function or logic can be spilt and modulated by a single agent.	System functions are intensively integrated. This can be achieved by exposing functions as services to end user.
Compatibility	Highly compatible with different technologies. Can be seamlessly integrated with a legacy system .	Serves to reach specific goals and lacks compatibility to integrate with other systems.
Adaptability	Goal-oriented, context-sensitive. The workflow and individual intelligence can be altered on demand.	Difficulty in adapting to a changing environment or new business rules.
Usability	Distributed, autonomous and highly dynamic environment	Distributed, autonomous and highly dynamic environment
Fault tolerance	Failures only take effect locally	System fails when a server goes down. Require redundancy .

Combined with the above discussion above, the agent-based approach is selected as the middleware technology in this thesis. In addition to the analysis above, the literature also reflects similar results. Fischer et al. (1996) listed four reasons to explain why an agent-based approach is suitable for application in the domain of transport and logistics industry. First, this industry is inherently distributed for example, the physical distribution of cargoes and the large-scale equipment. As a result, it is reasonable to model as a MAS. Secondly, the process of transportation involves a high-level of cooperation and coordination between self-interested parties. This makes ABA extremely suitable through means of negotiation, task decomposition and task allocation. Furthermore, maintaining and persisting knowledge engaged in the transport and logistics activity in a central location is a complex task. This becomes even more difficult when partial knowledge is unavailable. Thus it is reasonable to treat the entities (i.e., people and equipment) as an independent and autonomous agent. Similar findings also can be found in Davidsson et al. (2005). The authors of this research claimed that most transport logistics applications have the characteristics that make them most suited to the application of ABA. These characteristics include modular, decentralised, changeable, ill-structured, and complex.

3.1.3 The applicability of using ABA in PL system

As previously discussed, the current inefficiencies of C1 can be summarised in three aspects: inefficient communication, lack of coordination, and self-interests. From an architectural standpoint, the ABA is well suited to address these issues as actors can be modelled as self-organised and locally interested agents. By using ABA, sensitive information can be preserved, and the objective is achieved by agent intelligence. It also provides a highly flexible and effortless solution that compatible with the existing information system. Several works within the research literature confirm the potential use of ABA as the method for enhancing collaboration. For example, Sprenger and Mönch (2014) described a decision support system for cooperative transportation planning in the German food industry. The overall system was implemented with ABA which provided a distributed hierarchical algorithm for collaborative decision making. Hernández et al. (2014) proposed a multi-tier, negotiation-based MAS to promote the service and profit level of supply chain members using CDM. Panzarasa et al. (2002) presented a formal model for CDM in a multi-agent environment.

Similarly, the use of ABA in C2 is beneficial from several aspects. First, the agent provides system modularity. Each agent can be responsible for each component or condition. As such, the complexity of the BCS can be managed by refining the overall system into subsystems based on BCS components and BCS functions. Second, due to the autonomous nature of agents, agent-based architecture offers system mobility that allows the system to add, modify or replace any module without affecting the rest of the system (Pang, 2010). The use of ABA could be beneficial by providing intelligent and integrated monitoring of the system conditions.

3.2 Multi-agent system

In this section, the multi-agent system (MAS) will be introduced. In section 3.2.1, the concepts of agent and MAS are defined. Following this, different architectures of MAS are discussed in section 3.2.2. Issues associated with the development of MAS such as communication, agent management and agent-based middleware are outlined in section 3.2.3. Finally, a review of MAS applications is given in section 3.2.4.

3.2.1 Introduction of agent and multi-agent system

H. Nwana and Ndumu (1997) defined an agent as referring to a component of software and hardware which is capable of following instructions exactly in order to accomplish tasks on behalf of its users. A commonly accepted view of the agent is its three main features: autonomy, adaptability, and coordination (Guo & Zhang, 2010).

- Autonomy: each agent acts without any control of human beings or other objects.
- Adaptability: an agent is capable of responding to unexpected events and adaptable to a changing environment.
- Coordination: an agent can communicate and coordinate with other agents or entities in order to satisfy a set of the objectives.

An agent can display behaviours and properties. The level of intelligence and autonomy of an agent depends on the behaviours and properties it possesses. Based on different purposes and uses, three types of agent are presented (Russell et al., 2003) as follows:

- Reflex agent: this type of agent has a relatively simple level of intelligence in that can react to different environmental changes based on defined rules. For example, it works by finding a condition that matches the current external conditions and performs the actions that are associated with the rule. This type of agent is useful for the systems that require a fast time response.
- Goal-based agent: this type of agent has more advanced intelligence and complex behaviour. It can combine the current conditions with goal-information (desired actions) to make decisions in order to achieve the goal. The DM process may invoke complex reasoning or space searching processes when the goal cannot be derived from immediate actions. This type of agent is useful for complex DM context.

- Utility-based agent: when a goal can be achieved via multiple actions. This type of agent includes a utility function that will map the action and state to a real number. The agent can use this real number to generate corresponding actions. Meanwhile, the utility function can provide explicit trade-offs which help the agent to take appropriate actions when conflicting goals appear. Similar to the goal-based agent, the utility agent is also suitable for a complex DM context.

A common goal when designing an agent-based system is to enable an agent to communicate and corporate with other agents to complete complex tasks. A system comprised of two or more agents are known as the multi-agent system (MAS). Based on the definition proposed by Weiss (1999), a MAS is 'a system composed of a collection of autonomous and interacting entities called agents, evolving in an environment where they can autonomously perceive and act to satisfy their needs and objectives'. In general, a MAS can be categorised into two types, competitive and cooperative MAS. In a cooperative system, the agents in MAS strive to obtain a global goal. For example, when monitoring a large-scale network such as the power grid, each agent might represent a single infrastructure or asset. In this scenario, the goal of individual agents may be to monitor a single parameter that is not relevant to the global goal. The only goal that all agents are committed to reaching is to maximize the flexibility and reliability of the network. In contrast, a competitive system may include agents with conflicting goals. For instance, in a business negotiation scenario, each agent may represent one business party to negotiate with other agents. Therefore maximising its benefits is the goal for each agent.

3.2.2 MAS architecture

The architecture of a MAS defines how agents are organised. It describes the roles, relationships, and authority structures that govern the agents' behaviours (Horling & Lesser, 2004). In general, there are two types of architectures, namely, the fully distributed architecture and the hierarchical architectures. In a fully distributed architecture (as shown in Figure 3.2 (A)), each agent represents an entity, a job, or a party. Here, agents communicate with each other directly. This type of architecture is suitable for solving problems which can be divided into several subtasks. Each task will be executed by an agent through communication and negotiation with other agents. Results obtained from each agent will be aggregated together for operations management. The hierarchical architecture (see Figure 3.2 (B)) introduces the mediator agent into the MAS. This improves the robustness, stability, and optimality of the system. Maturana and Norrie (1996) defined the mediator agent as a distributed decision-making support system for coordinating the activities of a MAS. Coordination tasks include subtasking, the creation of virtual communities of agents and execution of the process imposed by the tasks. In this type of architecture, the mediator is assumed to be the system coordinators by promoting cooperation among intelligent agents and learning from agents behaviour (Shen & Norrie, Nov,1997).

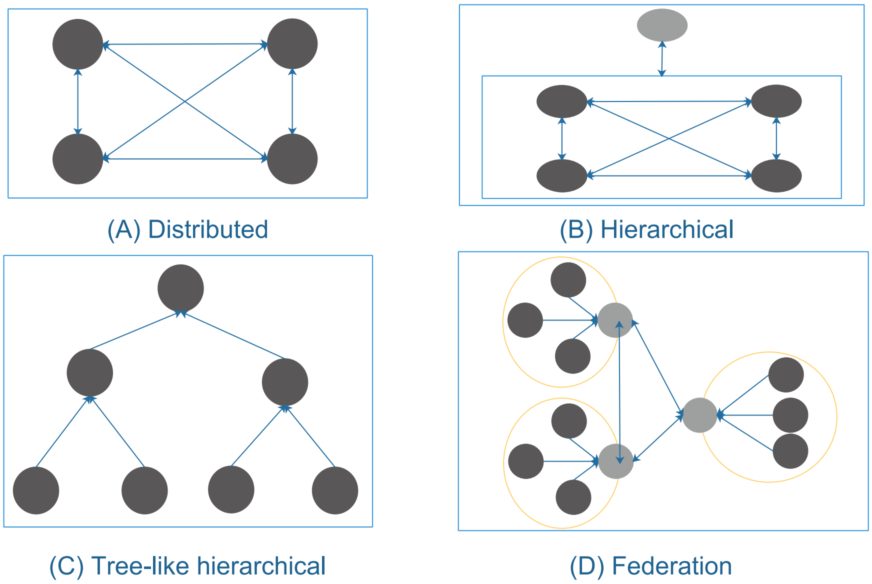

Figure 3.2: A categorisation of the MAS architectures

Other types of architectures are also involved in the development of MAS. Figure 3.2 (C) depicts a tree-like hierarchical architecture where agents in the higher layer have a more global view than those below them. The information generated by lower agents travels upwards to the higher agents and the control command created at higher layers flow downwards to provide directions to lower agents. This type of architecture is suitable in the design of control systems design. In addition, Figure 3.2 (D) outlines a federation. Each federation contains a mediator at the edge of the federate. The federate mediator is responsible for communication and information exchange with other federate. This type is suitable for controlling and monitoring large-scale and complex environment such as manufacturing. A comprehensive review of different types of architecture in the design of MAS, a paper Horling and Lesser (2004) was referred to.

In general, no single type of architecture exists that is suitable for all situations and contexts. The choice of architecture type is a trade-off between different factors such as computation complexity, coordination rules, user acceptance, organisation rules, communication overhead and so on. Thus, a deep understanding of the nature of the problem defines the most suitable MAS architecture.

3.2.3 MAS development

In this section, several key aspects related to MAS development are considered. This includes agent oriented programming, communication and language, agent management, and agent-based middleware.

Communication and language

Ensuring ambiguous communication between agents is an important task in the development of a MAS. The foundation of intelligent physical agents (FIPA) [3] established the standards of agent interactions and communication. A knowledge query and manipulation language (KQML) have been widely used as the language of agent communication language. More recently, however, this has been replaced by ACL (agent communication language) established by FIPA. An ACL message normally consists of several parts. This includes the content of the message, information about the sender and receiver (e,g, name and address), the type of the message (e.g., performative and call-for-proposal), and the used language (e.g., protocol). For complex and information-intensive communication among agents, ontology is applied to define the vocabulary used during conversation. A more detailed explanation of agent communication and FIPA standards can be found in O'Brien and Nicol (1998).

Agents management

In addition to the communication between two agents, the way of how multiple agents are managed is also important during the development phase of a MAS. Similarly, FIPA has established a standard for the agent management. A reference model for agent management is shown in Figure 3.3. The agent module corresponds to a specific agent which defines the life cycle of an agent in five different states. These states include initiated, active, waiting, suspended, and in transit. On top of the agent module, an agent management system is used to provide a run-time environment for agent deployment. A directory facilitator is used to provide yellow page services to agents. In other words, it stores a list of registered agents with their state and capabilities. A message transport system provides the transport protocol for message transfers between agents. The protocol describes the common rules that agent interactions should follow. To enable complex interactions, FIPA has also defined several types of protocols. A commonly used protocol is the contract-net protocol shown in Figure 3.3. Here, the initiator starts an interaction by sending a call for proposal to participants and waits for a response. The participator can either refuse the call or return the proposal. If a proposal is returned by the participants, the initiators will evaluate the proposal and either accept or refuse it.Except for the contract-net protocol, others like brokering, auctions, request, query, and user-defined protocols are also widely applied.

[3]FIPA: http://www.fipa.org

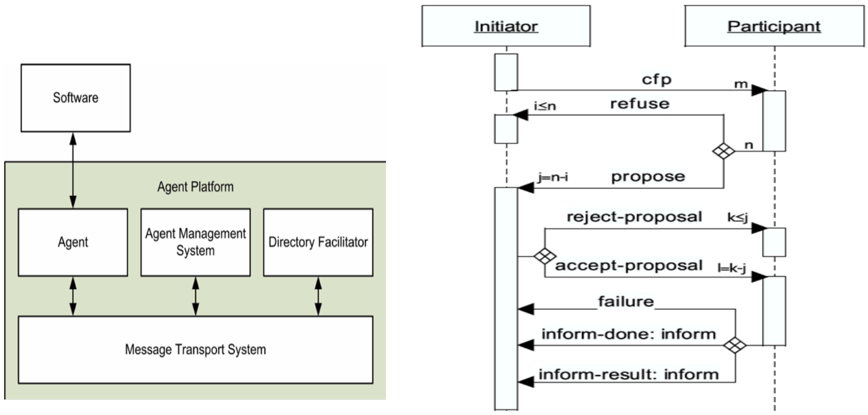

Figure 3.3: A FIPA agent management reference model, and communication protocol specification, derived from F. Bellifemine et al. (2001)

Agent-based platform

An agent-based platform (ABP) provides a generic software infrastructure for the implementation of a MAS. An advantage of adopting an ABP is that it is problem-independent. This enables developers to focus on the core of agent behaviour and intelligence. The ABP considers several common issues during MAS development, such as agent communication and management, and complies with the FIPA standards. Because most ABPs are developed using a popular language (e.g., Java), they also provide rich application programming interfaces and third-party libraries for which the agent capability can be extended and improved. There are numerous general-purposes ABPs available in literature, such as JADE (F. Bellifemine et al., 2001), Cougaar (Snyder et al., 2004), ZEUS (H. S. Nwana et al., 1999), and so on.

Each ABP has its scope of interests. To select the optimum ABP, four main requirements are considered. First, the language should be widely used. Second, it should comply with the FIPA standards. Third, it should be an open source software and been actively supported. Finally, it should be stable and scalable. Based on these requirements, a comparison is performed between several popular ABPs. These include JADE, ZEUS, JACK, and VOLTTRON.

The comparison is summarised in Table 3.3. In light of the comparison, the JADE ABP has been selected as the tool for MAS development in this thesis. Chapters 5 and 6 will further illustrate the implementation details by using JADE. The JADE is an open source agent development platform which is compliant with FIPA standards and fully implemented in Java (F. Bellifemine et al., 2001). It simplifies the MAS implementation through a run-time middleware and a set of tools and libraries that support application-dependent development. Another advantage of using JADE is that it also includes built-in graphical tools that support the debugging and develop-

Table 3.3: A comparison between different ABP platforms

Properties	JADE	ZEUS	JACK	VOLTTRON
Programming language	Java	Java	Java	Language independent
Standards	FIPA-compliant	FIPA-compliant	FIPA-compliant	Not specified
Technical support	Active	Discontinued	Active	Active
Development method	Command line, IDE	GUI based	GUI based	Command line
Advantages	Stable, various third party adds-on, can be implemented on mobile devices	Easy for development, suitable for rapid prototyping	Full components for Java development	Has hardware driver support
Disadvantages	Challenging for beginner	Weak documentation	Weak documentation	Limited usage in industry
Free of use	Open source	Open source	Not free	Open source

ment tasks. Because JADE is fully implemented in Java, it also supports deployments on different machines which do not need to have the same operating system. The JADE agent platform has been applied in multiple domains such as power grid control (Kantamneni et al., 2015), business negotiation (Wang et al., 2014), transportation scheduling (Narayanaswami & Rangaraj, 2015), and system monitoring (Dibley et al., 2015).

3.2.4 A review of MAS applications

In previous sections, different aspects regarding MAS development have been discussed. In light of this discussion, this section conducts a review of MAS applications in the engineering industry. There is a large number of research works whereby applying MAS the complex DM problems in various fields are solved. In response to the scope of this thesis, two typical MAS applications are reviewed. It includes a review of transport planning and a review of equipment control. The result is summarised in Table 3.4. First of all, the MAS is well suited for solving planning scheduling problem, especially in situations where quality decisions require intensive negotiation and information exchange. Moreover, given the high modularity of a MAS, it is also recognised as an important tool for coordination and control of a complex and geographically distributed equipment. A MAS can be designed to divide a complex system/task into subparts where an agent or a group of agents handles each parts and the overall decision is generated by merging all sub-decisions into a universal one with certain criterion. This can potentially leads to improved solution quality or a reduction in the amount of

Table 3.4: An overview of the MAS applications for supporting planning and control

Function	Publications	Problem domain			Approach			Contribution
		Domain	Usage	Purpose	architecture	Agent attitude	Agent model	
Planning and scheduling	Wang et al. (2014)	Virtual enterprise	Automation	Negotiation	Distributed	Selfish	User	Theoretical
	N. Liu et al. (2007)	Manufacture	DSS	Job-shop planning	Distributed	Cooperative	Machine	Theoretical
	Dullaert et al. (2009)	Intermodal transport	Automation	Transport request scheduling	Distributed	Cooperative	User	Software implementation
	Baykasoğlu and Kaplanoğlu (2015)	Transport and logistics	DSS	Truck operations planning	Distributed	Cooperative	Truck	Theoretical and implementation
	Schut et al. (2004)	Transport and logistics	DSS	Barge rotation planning	Distributed	Selfish	User	Software Implementation
	Huang and Liao (2012)	Manufacture	DSS	Job-shop scheduling	Hierarchical	Selfish	User, mahcine	Theoretical
	Yin et al. (2011)	Port Logistics	DSS	Berth and truck allocation	Distributer	Cooperative	resources	Theoretical
	Leung et al. (2010)	Manufacture	DSS	Process and jobshop planning	Hierarchical	Selfish	Machine	Theoretical and implementation
	Y. Zhang et al. (2014)	Manufacture	Automation	Jobshop planning	Hierarchical	Both	Machine	Theoretical
Operational control	Mahdavi et al. (2013)	Cement production	DSS	Production monitoring and control	Hierarchical	Cooperative	Machine	Theoretical and implementation
	McArthur et al. (2005)	Power system	DSS	Condition monitoring and diagnosis	Distributed	Cooperative	Machine	Theoretical and implementation
	L. Liu et al. (2007)	Electric ship	DSS	Fault diagnosis and condition monitoring	Distributed	Cooperative	Machine	Theoretical
	Dawson et al. (2011)	Flood management	DSS	Risk analysis and policy	Distributed	Cooperative	Events and users	Theoretical
	Yu et al. (2003)	Manufacture	DSS	Service maintenance problem	Distributed	Cooperative	Machine	Theoretical and implementation

time spent on computational time. A typical sub-part problem of an agent or a group of agents includes a set of operations it should perform, a set of constrains it should meet, a set of resources it should assign, a set of objectives it should achieve, and also the current state of the agent.

In sum, the given characteristics of a MAS present potentials for improving the performance of PL systems, specifically for the two addressed challenges. in Chapter 5, a MAS will be implemented to assist the DM process of a barge transport planning. in Chapter 6, a MAS will be implemented to support the DM process of reliability assessments of a large-scale BCS.

3.3 Selection of IDM approach for collaborative planning

In section 2.3, the problem of hinterland barge transport was discussed. It was concluded (in section 2.7) that a potential way is to implement an integrated DM system using ICT technologies. For the HBT planning problem, an approach to support planning needs to be selected. Within the literature, there are mainly three types of approaches that can be used for solving a planning problem, namely an exact algorithm, heuristics and meta-heuristics. In this section, three approaches are briefly reviewed, and the choice of using meta-heuristics is justified.

3.3.1 Exact algorithm

The exact algorithm is widely used for finding optimal solutions to classical planning problems. By formulating the problem into a standard model (e.g., integer linear programming and mixed-integer programming), the exact methods can guarantee to find an exact solution in a finite amount of time. Typical exact methods include branch and bound, linear programming, dynamic programming and so on. Use of the exact algorithm has a long history in the domain of operations research. However, its limitations are evidenced by solving large-scale scheduling problem (Jourdan et al., 2009). For example, for a problem that is categorised as NP-complete or NP-hard, the use of the exact algorithm solvers cannot be applied to find the optimal solution within polynomial time. In order To cope with the large-scale optimisation problems, especially for the NP-hard and NP-complete ones, the use of heuristics is preferred.

3.3.2 Heuristics

Compared to the exact algorithm, heuristics is a problem-dependent algorithm which can find an acceptable solution within a reasonable period compared to the exact algorithm. The use of heuristics cannot guarantee an optimum solution. In general, the solutions are normally worse than the optimal one. The effectiveness of a heuristic method depends upon its ability to adapt to a particular problem, avoid solutions been trapped at local optima, and exploit the basic structure of the problem. Amen (2000) proposed a priority rule heuristics to address the assembly line balancing problem. Ruiz and Maroto (2005) surveyed the heuristics methods been applied to the problem of floor shop scheduling. They compared the heuristics ranging from Johnsons algorithm to dispatching rules on computation efficiency accuracy. Due to its problem-dependent nature, there exists no framework exists that the use of heuristics can be based on. In contrast, the meta-heuristics provides a high-level problem-independent algorithmic framework which defines a set of guidelines or strategies to develop heuristic optimisation algorithms.

3.3.3 Meta-heuristics

Osman and Laporte (1996) defined meta-heuristics as 'an iterative generation process that guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space; learning strategies are used to structure information in order to find efficiently near-optimal solutions'. Meta-heuristics can be used to solve NP-hard problems with a large number of variables and non-linear objectives. Unlike heuristics, which requires significant efforts to customise the algorithm to suit specific problems, meta-heuristics can be used to solve any optimisation problems. For a comprehensive review of meta-heuristics and its applications, the work of Osman and Laporte (1996) was referred to. The meta-heuristics that considered in this thesis are introduced below.

Simulated annealing (SA) algorithm is a generic probability method. It is inspired by the process of annealing in metallurgy. This involves heating a substance and then gradually cooling it to obtain a high-strength crystalline structure. The advantage of using SA based optimisation is evidenced that any intermediate non-improving moves can be accepted with a probability which takes the current best optimal away from local maximum (Crossland et al., 2014). Kirkpatrick et al. (1983) first introduced the SA in the optimisation domain. in 1983 Since then, the SA had been used extensively for solving optimisation related engineering problems such as vehicle routing (BañOs et al., 2013), job-shop scheduling (Faccio et al., 2015), terminal berth allocation Lin and Ting (2014) and so on.

A genetic algorithm (GA) is an iterative stochastic algorithm in which natural evolution is used to model a solution search process. It is inspired by population genetics (including heredity and gene frequencies), and evolution at the population level, as well as the Mendelian understanding of the structure (such as chromosomes, genes, alleles) and mechanisms (such as recombination and mutation) (Brownlee, 2011). A GA normally consists of three operators: (1) selections: individuals in the population are selected as parents; (2) crossover: parents are crossbred to generate offerings; (3) mutation: the offspring may be altered according to the mutation rules. For each generation, the objective is to improve the fitness of the individuals (the objective value). Such a result is obtained by simulating the two principal mechanisms (the crossover and mutate operators) which govern the evolution of living beings. The use of a GA-based approach is applied to multiple engineering problems, such as vehicle scheduling (Zuo et al., 2015), ship planning (Bruzzone & Signorile, 1998), parameters optimisation(Bäck & Schwefel, 1993) and so on. In addition to the classic version, the core of GA is also extended to cope with more complex problems, such as multi-objective optimisation using a non-dominated sorting genetic algorithm (Deb et al., 2002).

Furthermore, in addition to the application of single meta-heuristics, the development of hybrid meta-heuristics by combining two or more algorithms has also recently attracted attention in different engineer industry. By integrating the advantages of different approaches, the solution quality and computation efficiency can expect to improve. For example, Junghans and Darde (2015) combined SA with GA for building optimisation. Both computation time and solution quality were improved by applying the hybrid approach compared with the single one. Hybrid meta-heuristics is also used in multi-objective planning. For example, L. Zhang et al. (2013) integrated tabu search with a GA to address a dynamic job shop scheduling problem.

3.3.4 Discussion

In this thesis, a hybrid algorithm combined with a simulated annealing and a genetic algorithm will be used in Chapter 5 to assist the collaborative planning process. This choice has been motivated by two main points as discussed below.

Firstly, the parties (i.e., terminal and barge companies) involved in the hinterland barge transport have their interests. To enable collaborative planning in this context, different interests need to be harmonised. This can lead to a multi-objective planning problem. To solve a multi-objective bi-level problem, the main goal is to obtain the solutions that comprising the Pareto front. Any solution of the Pareto set is optimal in the sense that no improvement can be made on one objective without degradation of at least one other objective. Within the literature, several existing meta-heuristics (e.g., evolutionary algorithms and particle swarm optimisation) are population-based. This means that we can aim to generate several elements of the Pareto optimal set in a single run. Moreover, meta-heuristics seem particularly suitable to solve multi-objective optimisation problems. This is because they are less susceptible to the shape or continuity of the Pareto front (e.g., they can easily deal with discontinuous or concave Pareto fronts), whereas this is a real concern for mathematical programming techniques.

Secondly, the requirements for applying the exact method assumes all the information is collected and aggregated at a single place which is available to process. As discussed in section 2.3, one of the major issues of the HBT problem is the lack of collaboration between terminals and barges party, the root cause is lack of information sharing. Though both terminal and barge party aware of this issue (with discussion from industry expert from Port of Rotterdam), it appears difficult to achieve an acceptable solution because autonomy and privacy are their primary concern. In this sense, centralised planning is not realistic, which means it is difficult to gather all the information from both terminals and barges to the one single location for centralised planning. For this thesis, the condition of adopting a commercial solver (e.g., Cplex) when information is available. In other words, the planning board of all terminals from the port region and all barge sailing plan are available.

3.4 Selection of IDM approach for reliability assessments

As illustrated in section 2.5, the on-going challenges associated with reliability assessment of equipment in PL system can be summarised in three main aspects; (1) unreliable inspection and DM; (2) lack of a system of information integration; (3) uncertainty of problem ownership. Given these deficiencies, the desired functions from the intelligent DM approach are listed as follows:

- Inform users of on-going affairs.
- Determine the consequence that could be potentially induced by a disruption.
- Determine what method or algorithms that could assist in identifying and correcting the condition.
- Proposing suitable actions.
- Identifying, listing and documenting the disruptions and the actions been taken.

The core of the desired functions can be summarised as delivering the right information to the right person at the right time. It not only emphasis on designing specific method to cope with specific faults, but also expects an integrated and intelligent system that incorporates the user interests, system specifications, and expert knowledge. To achieve this, the concept of context-awareness and the context-awareness system is introduced below.

3.4.1 Context-aware system

A widely accepted definition of context is given as: 'any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to be the interaction between a user and an application, including the user and applications themselves' (Krummenacher & Strang, 2007). A system is characterised as context-aware if 'it uses context to provide relevant information and services to the user, where relevancy depends on the user's task' (Dey, 2001). The CAS has been applied extensively in the health care industry (Bricon Souf & Newman, 2007) and education (Rani et al., 2015). Recently, there has been a growing interest in applying CAS to facilitate decision-making in a data-intensive context. Evchina et al. (2015) designed a context-aware information management middleware to address the problem of selective information delivery in data-intensive monitoring systems. Nadoveza and Kiritsis (2014) also considered information delivery in the context of the enterprise information system by context-modelling and reasoning. K. Kim et al. (2015) developed a CAS to assist risk management in the cold chain logistics.

CAS in reliability assessments

To extend the applicability of CAS and fulfil the needs of equipment supervision, several works have already been published. on the subject. For example Pistofidis and Emmanouilidis (2012), introduced context-awareness in engineering asset management. The work delivers a context-aware e-maintenance framework for context capturing and responding in a shop-floor environment. Similar work was conducted by Galar et al. (2015) who proposed a CAS for decision-making associated with asset wear and tear maintenance. Kumar et al. (2015) and Elhdad et al. (2013) attempted to integrate a CAS into system monitoring and maintenance.

3.4.2 Selection of technological solutions of CAS

To implement a CAS, the first step is to choose an appropriate methodology for context modelling. The context model should fulfill several requirements:

- Formal definition and access to knowledge, its storage, maintenance, use and modification
- In-depth analysis, decision-making and control
- Capturing dependencies, relations, inference between components, subsystem, systems, and users.
- Dealing with the complexity of risk identification, analysis, assessment, and knowledge exploitation
- Ease of maintenance and re-use

In response to the requirements outlined above, three major technological solutions are examined, namely complex event processing, linked stream data and ontology.

- Complex event processing (CEP) is a method of analysing large streams of data and inferring patterns and features with pre-defined rules. Its usages are found in multi-sensor data fusions in cars (Terroso-Saenz et al., 2015), security monitoring in a visualised environment (Baumgartner et al. (2015)), and information systems in logistics (Gaunitz et al. (2015)). Several limitations exist when applying CEP: First, the CEP concerns timely detection complex events with a stream of context information which heavily rely on a large number of rules. These rules require consistent checking (Anicic et al., 2010). Second, the given rule is expressed with engine-specific language which could impede the interoperability of the system (Evchina et al., 2015).

- The linked stream data (LSA) approach is also recognised as a tool for context-modelling. It offers a capability for deriving implicit knowledge from explicit statements. It also supports continuous queries with query languages. However, its limitation is given that it mainly focuses on temporal data processing and reasoning over RDF models. This limits its scope of use.

- Ontology is defined as an explicit specification of a conceptualisation, which has been used to describe a specific domain knowledge where concepts and relationships are unambiguously defined and checked (Staab & Studer, 2010). As such, it is perceived as a knowledge base where information can be extracted or inferred from. Its advantages are given as (1) Genericity: the ontology model is about knowledge and information integration which can be designed as an upper model been applied across multiple domains. Therefore, it is considered to have significant expressive power (Ejigu et al., 2007); (2) Reasoning: the reasoning mechanism is considered to be an extraordinary property over other models. It can infer and reason knowledge and new information based on the current context. It is also used to evaluate the consistency and completeness of the current model (Nadoveza & Kiritsis, 2014); (3) Granularity: the ontology model allows concepts to be defined closely to related entities or contexts. This is known as fine-grained ontology. It allows the application of ontology more applicable in different domain either general or specific.

In sum, both methods can be used to model the context. The reasoning capability of CEP relies extensively on the rules. This can be difficult to maintain in a complex environment. Also, implicit knowledge cannot be inferred. Concerning LSD, it is efficient in handling large streams of data and supports temporal reasoning. However, its use is also limited to semantic reasoning. The ontology method provides ideal properties concerning the requirements. Firstly, knowledge from a specific domain can be structured based on taxonomies and relationships in the ontology. It provides a standard and vocabulary on which to model a domain. This is easy to maintain and use. Secondly, the ontology can capture and reason relationship and knowledge through properties, rules and constraints. Thirdly, it is also able to reason new facts or knowledge based on current context.

In light of the discussion above, the selection of ontology as the CAS technological solution is justified. A detailed introduction of the ontology will be provided in the following section, and its use will be discussed in Chapter 6.

3.4.3 Ontology

In the previous section, ontology was selected as the context modelling tool. In this section, an introduction of ontology is presented.

Definition

The term ontology is formally defined as an explicit specification of a conceptualisation (Guarino et al., 2009). The definition originates from the domain of artificial intelligence and computer science industry. To be more specific, ontology is a collection of (Ramos, 2015):

- a set of terms and vocabulary related to a certain domain.
- a set of relationships between terms.
- a set of axioms that define the intended vocabulary.

An ontology is used to describe, represent and model a domain knowledge. It uses a set of concepts to describe the domain knowledge and uses relationships to represent the dependency between concepts. A set of axioms is used to define the scope of interpretation and inference. Every branch of science has its ontology (Hadzic et al., 2009).

Logic for ontology representation

To interpret and represent an ontology in a generic form, formal logic is required. Descriptive logic (DL) is widely recognised as for describing and reasoning knowledge with expressiveness and decidability (Baader, 2003). The DL is defined as a set of knowledge representation languages that can be used to model or represent the knowledge of an application domain in an organised, structured, and easy to understand way (Staab & Studer, 2010). Assuming we want to express a concept of ”A man who has three children, all of whom are students”, the concept can be described by DL as follows:

$$\boxed{Human \sqcap Male \sqcap (>= 3 hasChild) \sqcap hasChild.Student}$$

A DL architecture is shown in Figure 3.4. It includes four parts.

- TBox: A terminological formalism, known as TBox, is used to define descriptions of a complex concept. For example, the statement below represents that a female is a human:

$$\boxed{Female \sqsubseteq Human}$$

- ABox: An assertional formalism, known as ABox, is used to define the property of individuals. For example, the statement of ”Tom has a child whose name is Mary” can be expressed as:

$$\boxed{hasChild(Tom, Mary)}$$

- Reasoning: A DL provides a user with different reasoning capabilities which can be used to infer implicit knowledge from explicitly represented knowledge. For example, it can infer the relationship between concepts and individuals.
- Descriptive language: The DL, by definition, is a knowledge representation formalism. A language is needed in order to apply DL in practice. In the following section, two major languages will be introduced.

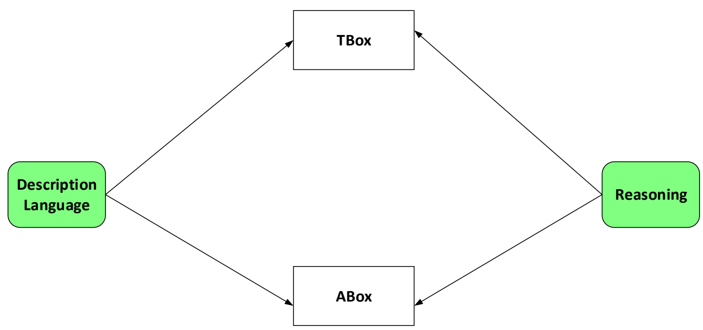

Figure 3.4: A Descriptive Language architecture, derived from Ramos (2015)

Descriptive language

In ontology development, the descriptive language is used to translate data into meaningful representations. In other words, the descriptive language is used to give meaning to data. The resource description framework (RDF) is the first established language to describe the semantics to data. It provides a data model to present the semantics of data. The data model is a triple which includes:

- Subject: the resource to be described.

- Predicate: a predicate of the resource.
- Object: the value of the subject

The RDF is widely used on the internet to link data from different sources together, and enable machines to interpret the data in a meaningful manner to end users.

For example, although RDF is widely used in ontology development, RDF is not expressive enough to provide information about its predicates and the relationship of its predicates with other resources (Ramos, 2015). Consequently, other frameworks are introduced such as RDFS and OWL(ontology web language) . Among these, the OWL is the most used language in ontology development. In general, the OWL is build based on RDF. This significantly improves its expressiveness. In RDF, the expression is straightforward. For example, 'A hasChild B'. In OWL, it can be added to define properties and classes. For example, in OWL, the expression 'A hasChild B' also implies that 'B isChild of A'. Similar to RDF, the OWL also contains three major elements,

- Classes: a collection of individuals such as system and components.
- Individuals: belongs to classes, is related to other objects and data values via properties.
- Properties: a collection of relationships between individuals. This includes data properties (e.g., hasValue) and object properties (e.g., isChild).

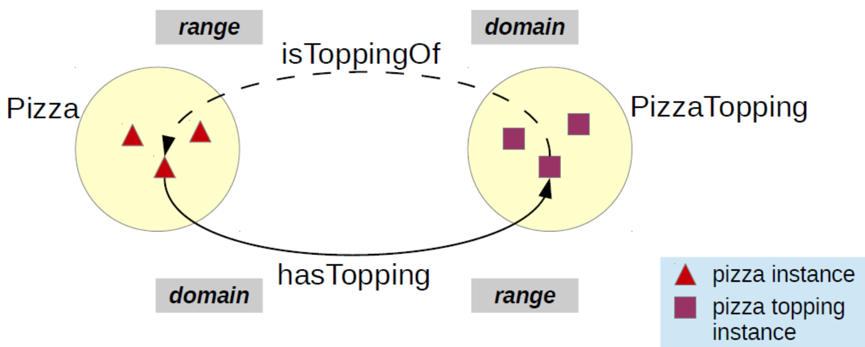

Figure 3.5: An example of OWL elements , derived from Drummond et al. (2007)

An example of pizza ontology (Drummond et al., 2007) using OWL is shown in Figure 3.5. In this example, two core classes are identified, namely, a **Pizza** and a **PizzaTopping**. The individuals of each class are defined under the scope of each class. For example, a **ChessyPizza** is an individual of a **Pizza**. The property defines the relationships between individuals. In this example, two properties are defined, namely **isToppingOf** and **hasTopping**. If a **ChessyPizza** is an individual of Pizza, and **MozzarellaTopping** is an individual of **PizzaTopping**, it is possible to link these two instances by the property **isToppingOf** which can be written as:

$$\boxed{isToppingOf(MozzarellaTopping, ChessyPizza)}$$

The OWL is a language which is used to express descriptive logic. As discussed previously, a DL has two major components which are TBox and ABox. A TBox is a schema that defines the core concept of a certain domain. An ABox includes a set of instances asserting facts based on the concepts defined in TBox. Tables 3.5 and 3.6 presents the mapping between OWL and DL with respects to both TBox and ABox.

Table 3.5: An OWL semantics (TBox)

OWL syntax	DL representation	Example
subClassOf	$C1 \sqsubseteq C2$	$Female \sqsubseteq Human$
equivalentClass	$C1 \equiv C2$	$Man \equiv Human \sqsubseteq Male$
subPropertyOf	$P1 \sqsubseteq P2$	$hasDaugther \sqsubseteq hasChild$
equivalentProperty	$P1 \equiv P2$	$cost \equiv price$

Table 3.6: An OWL semantics (ABox)

OWL syntax	DL representation	Example
type	$a : C$	$John : Male$
property	$< a, b >: R$	$< John, Mary >: hasChild$

Ontology based reasoning

As previously discussed , a DL is used for ontology development. A DL is constructed with four parts as shown in Figure 3.4. The definition of TBox, ABox and description language have already been outlined. The last part is ontology reasoning.

The ontology reasoning is implemented via rule execution. The semantic web rule language (SWRL) is well defined and extensively used in ontology reasoning development. It is an extension of OWL language and serves to enrich expressiveness. A general form of SWRL rules are given as follows:

$$\boxed{antecedent \Rightarrow consequent}$$

Both antecedent and consequent are conjunctions of axioms which can be written by $a1 \wedge a2 ... \wedge an$. Each axioms is composed by variables and predicates. For example, if we want to represent a rule which reasons that if 'a' is the parent of 'b' and 'a' is the brother of 'c', then 'c' is the uncle of 'b'. A SWRL representation is written as follows:

$$\boxed{\quad parent(?a, ?b) \wedge brother(?a, ?c) \Rightarrow uncle(?c, ?b) \quad}.$$

Based on the examples presented above, it can be concluded that the use of rules enriches the expressiveness of ontology by enabling new knowledge generation via reasoning. In a plain ontology, the parent, brother and uncle properties should be defined explicitly. By applying rules, the ontology is capable of reasoning the uncle property based on the parent and brother properties.

Motivation for using an ontology for reliability assessment

In previous sections, the basics of ontology were introduced. To be concluded, ontology is used to model a domain knowledge by defining its entities and specifying the relationship between defined entities in an organised way using descriptive language such as OWL. Ontology can be seen as a knowledge-based database. By using an ontology, the data source is linked together, and new knowledge or fact can be inferred from the ontology. Several major advantages are listed as shown below:

- Structuring knowledge about a specific domain based on taxonomies and class hierarchies
- Defining a standardised and common vocabulary for users and software
- Instantiating data, facts, and knowledge elements
- Capturing complex relationships between knowledge elements through properties, restrictions and rules
- Reasoning about facts and inferring new ones

The ontology has been successfully applied in multiple industries. It has been proven to be effective in supporting information integration and mining in a data-intensive environment. For example, Evchina et al. (2015) developed a context-aware middleware to assist monitoring tasks in a data-intensive environment (e.g., building condition monitoring) where the ontology is chosen as the primary tool for context modelling and information integration. K. Kim et al. (2015) developed an intelligent risk management system for logistics chain monitoring. The domain knowledge is structured by the ontology. Different rules are incorporated in the ontology to support a real-time assessment of potential risks. Different uses of ontology can also be found in disease diagnosis (Y. S. Chang et al., 2015), risk management in logistics (K. Kim et al., 2015), e-maintenance (Chioreanu et al., 2014), building automation (Evchina et al., 2015), and enterprise management (Nadoveza & Kiritsis, 2014).

In section 2.7, several potential improvements regarding reliability assessments of large-scale equipment were proposed. One improvement is evidenced via information integration. Integration should not only emphasise on storing data in a central place but more focus on how to transform data into useful information. It should provide a way of linking different data sources and knowledge together. Given such requirements, the ontology can be seen as a meaningful tool to fulfil the requirements as listed blow.

- Facilitate interoperability and communication between various ICTs;.
- Formalise knowledge about equipment assessment knowledge;
- Define, describe, and standardise vocabulary, concepts and relationships between system activities and components.

Currently, no available work has integrated the ontology technologies in the domain of the PL systems to assist reliability assessment functions. In light of this,this research attempts to introduce ontology to PL and discuss how it can be used, and investigate its potential benefits. A detailed discussion will be given in Chapter 6.

Ontology design approach

According to Uschold and Gruninger (1996), there are three general approaches for adopting an ontology design.

- Top-down approach: this approach begins with the most general concepts in the considered domain and subsequently expands and specialise the concepts.
- Bottom-up approach: this approach begins with the most specific entities without considering the hierarchy. Subsequently, it groups the defined entities into a generalised concept.
- Middle-out approach: this approach is a combination of the previous approaches, and it is more flexible compared with the top-down and bottom-up approaches. This approach emphasises the balance of level of details, namely, entity, concept, relationships, and individuals, only to be captured as necessary. Consequently, the use of the middle-out approach allows for more application-specific ontology.

An ontology development tool: Protege

An ontology tool is needed for ontology development. In this research, the tools named Protege has been chosen. A screen-shot of the Protege development interface is shown in Figure 3.6. Protege is an ontology development tools developed by Stanford University. There are several features enabled Protege to become popular tools for ontology development domain, they are outlined below. A comprehensive review of different tools can be found in Youn and McLeod (2006).

- User interface: Building an ontology is a complex task which requires the engineer to have specific knowledge to construct a complete ontology. The Protege eases the development process by providing a user-friendly interface. The interface allows the user to construct domain ontology, customize data entry and import data.

- Rich built-in tools: Due to its popularity, the Protege is supported and developed by a community of engineers. There are plenty of built-in tools that allow users to design and verify the ontology efficiently. For example, a built-in reasoner allows users to determine class inconsistencies and discover implicit information. A UML visualisation tool allows end-users to visualize the overall ontology.

- Java language compatible: Another feature is its compatibility with Java. The ontology created by Protege can be directly transformed to Java code. Consequently, it allows the developed ontology to be further integrated with a Java-Based software system. It enables any Java-based system to integrate and interact with the ontology knowledge base easily.

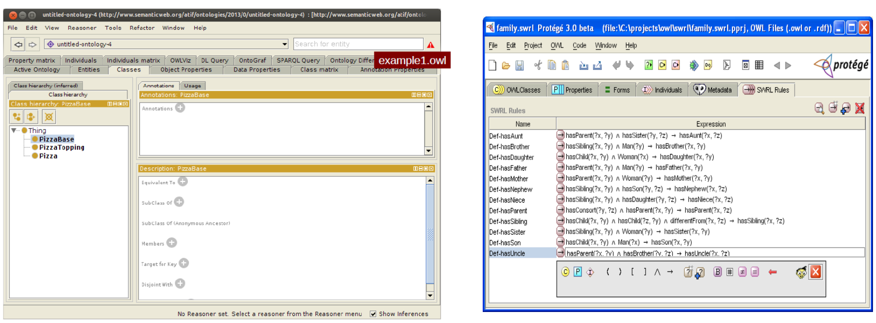

Figure 3.6: A screen shot of Protege development interface. The right side shows a class definition panel. The left side shows a SWRL panel used to define and run SWRL rules.

3.5 Discussion

It can be seen from previous sections that the following two technologies have been selected:

- Middleware technology: a multi-agent system has been selected to support information exchange, coordination and collaboration, and preserve an entity autonomy.
- IDM selection: a meta-heuristics has been selected to assist barge transport planning. An ontology has been selected as the tool to implement a context-aware system and subsequently support reliability assessment.

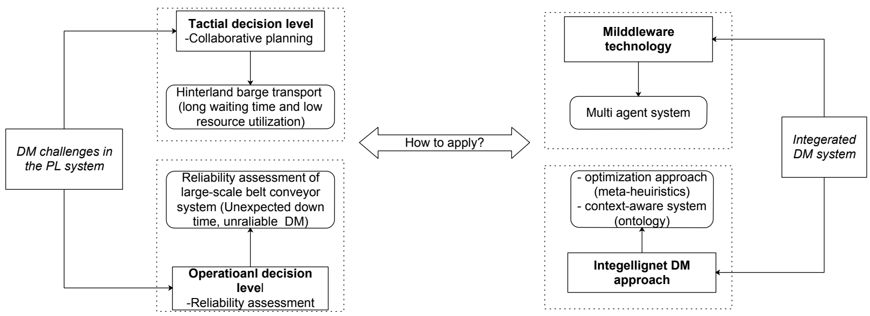

Figure 3.7: A summary of the technology selection and the open questions

However, at this stage , it is still unclear about how to apply the chosen technologies to solve these problems (see Figure 3.7). The questions include how the agent system should be organised and how to integrate the IDM with the agent system. As a result, three steps should be taken:

- Step 1: Design a generic framework to integrate all the components of the system. The first step relates to the design of an information system architecture. The architecture is defined as defining and controlling the interfaces and the integration of all the components of the system (Zachman, 1987). As discussed in section 2.6, different architectural types are available in the current PL ICT applications, e.g., a centralised architecture, a fully distributed architecture. Each architectural type has its scope of applicability, and pros and cons. It is required to generalize an architectural type based on the nature of the addressed problems.
- Step 2: Specify the scope and responsibility of each component. The second step is a follow up of the architectural design. By specifying the architectural type, the responsibility and the function of each component needs to be defined. The connections and relationships between each component need to be defined as well.

- Step 3: Implement the functionalities of each component and assessing the performance and results. The third step is the implementation stage where the architectural type is initialized and adapted to the specific use case.

in Chapter 4, we will design an integrated architecture as described in steps 1 and 2. The step 3 will be executed in Chapters 5 and 6.

3.6 Conclusion

This chapter provides answers to the sub-question 1.2, as listed in Chapter 1. It justified the choice of technologies to support the integrated DM system development. Specifically, two elements have been selected namely the technology for middleware development and the technology for IDM.

The multi-agent system has been selected as the primary tool for meeting the requirements of middleware development for the following reasons. First, a MAS system enables information exchange in an open and dynamic distributed system. Second, a MAS is capable of supporting iterative and hybrid behaviours. This means that agent systems can handle, select and combine multiple behaviours to react appropriately to any requirements. Third, the modularity of a MAS makes it a suitable candidate for establishing monitoring tasks for a large-scale and complex system.

Intelligent DM approaches are selected in sections 3.3 and 3.4. Several theoretical planning approaches are reviewed in section 3.3. As a result, meta-heuristics has been selected as the main approach to support collaborative planning. To enhance reliability assessments of large-scale equipment, the context-aware system was introduced in section 3.4. This will be used to support tasks including knowledge management and information integration. As a technological solution, the ontology is chosen to implement the CAS system to support equipment reliability assessments in a PL system.

The selection of specific technologies has now been determined. Therefore, it is essential to explore how best to apply them to improve PL performance. In Chapter 4, a framework will be proposed to integrate the technologies discussed in this chapter. Based on the framework, intelligent DM systems will be developed to achieve efficient collaborative planning and reliability assessment in Chapters 5 and 6 respectively.

Chapter 4

An integrated decision-making framework

Chapter 3 selected different ICT technologies that could solve the DM challenges, which were previously discussed. Two key elements were identified: middleware technology and an intelligent DM approach. For middleware technology, the agent-based system was chosen as it fulfils the requirements for collaboration, information exchange, and autonomy. Likewise, concerning the intelligent DM approaches, meta-heuristics and context-aware system were introduced, which addressed the collaborative planning and reliability assessment challenges. It was also concluded in Chapter 3 that an architecture is required in order to integrate the selected tools.

Given this, this chapter explores how to integrate different ICT tools to address specific challenges. Section 4.1 introduces the requirements and functionalities required with regards to the architecture design. Section 4.2 provides a classification of different architectural types and offers a comparison between them. Section 4.3 proposes an integrated DM framework, based on which two technological frameworks are designed to address the two challenges highlighted in section 4.4. The conclusion is drawn in section 4.5.

4.1 Introduction

As discussed in Chapter 2, the PL system is facing two DM challenges: collaborative planning of hinterland barge transport and the reliability assessments of large-scale conveyor belt systems. The challenges are caused by two reasons: (1) insufficient information exchange and sharing among different entities (2) insufficient decision-making support. To address these DM challenges, an ICT system is required to meet the following criterion:

- Provide an information platform for efficient information exchange and sharing between different PL entities. For example, communication between joined parties and communication between systems and sub-systems.
- Provide intelligence to support the decision-making process. For example, to achieve effective collaboration and coordination.

Based on these requirements, different ICT tools were introduced in Chapter three. The goal is to use these ICT tools to provide functions that address the requirements. The agent-based system was selected as the primary technology to support middleware development, which is used to assist the exchange of information. The meta-heuristic and ontology-based approaches were selected as tools to enable efficient intelligence.

To bridge the gap between these requirements and ICT technologies, an integrated DM framework is required. The purpose of designing an integrated DM framework is to address the requirements mentioned above and to implement the required functionalities, which are summarised as follows:

- F1: Handle user requests and external environmental changes by providing a platform for information sharing and processing. This function requires an integrated DM system that provides information exchange and sharing environment in order to respond to the challenges. As discussed in section 2.7, collaborative planning involves different parties with different interests. An efficient planning process is determined by the efficiency and the effectiveness of the way that information has been exchanged. For example, in the hinterland barge transport case, the owner of the request (i.e., a barge operator) can indicate the receiver of the information (i.e., a terminal operator). Similarly, the equipment used in a PL system is normally complex and large-scale. In order To enable an efficient reliability assessment, the components, processes, and systems should be integrated. Such integration may require intensive information exchanges between different functional blocks. For instance, the causes of a braking system failure of a belt conveyor system may include pad abrasion and low hydraulic pressure (Lodewijks, 2005). In order to make an accurate diagnosis, information about the condition of the pads and hydraulic pressures should be shared and mined.
- F2: Control the information exchange process and provide positive guidance to achieve a more efficient and optimum DM. Information exchange and sharing are supported by F1. However, without certain controls, the information exchange process can be inefficient. Regarding the collaborative planning process, coordination is required during the negotiations between self-interested parties to ensure a satisfactory planning result. Similarly, interdependency may occur during interactions. For example, diagnosing a braking system fault of BCS may depend on the condition of a pad and the hydraulic pressure. An assessment of the braking system can only take place if the dependent conditions are

met. These type of information exchanges require coordination, which should answer certain questions, such as when to send a message and to where. To conclude, F2 focuses on regulating and controlling the way that information has been exchanged.

- F3: Provide basic services to ensure effective communication and information processing. This function is used to support internal processes, including information exchange and decision-making. It defines the negotiation protocol between the sender and receiver of information.

4.2 A classification of DM systems

Lang et al. (2008) presented a framework to classify logistical planning systems based on their information and decision-making capabilities. The classification spectrum, which is given in Figure 4.1, is inspired by their work, but this research focuses on the decision-making problems within a PL system. The classification is therefore based on information availability and DM objectives. Information on PL systems can be generated by global events and maintained globally, or it may pertain to a local event and maintained locally. As discussed in Chapter 1, decision-making problems include three levels: strategic, tactical, and operational. Based on the information and level of DM, a spectrum of classification is presented, which includes a centralised, fully-distributed, and hierarchical system. The advantages and disadvantages of each type are discussed below.

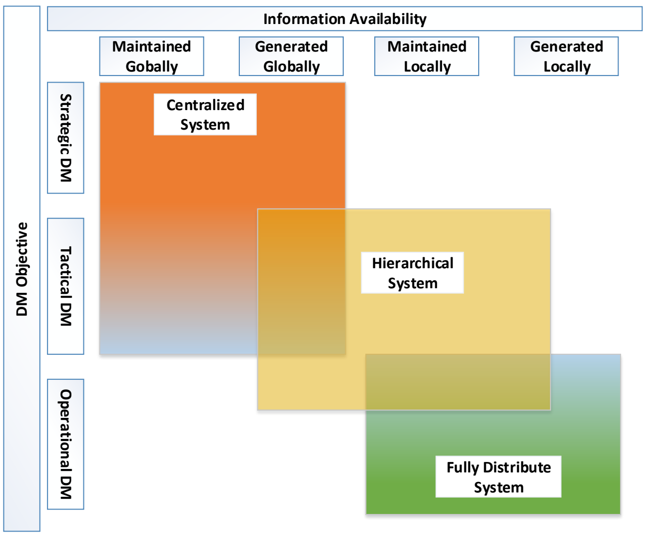

Figure 4.1: A classification of the DM systems based on the role of information availability and the DM objective, derived from Lang et al. (2008)

Centralised system

As shown in Figure 4.1, a centralised system in PL systems is used to facilitate strategic and partial tactical DM problems. In a centralised system, information is held or pertained globally, and all decisions are made at a central level. An example of a strategic level DM within a PL system is the port community system, which was discussed in section 2.6.2. Information is maintained in a PCS and can be accessed by different stakeholders. At a tactical level, a terminal operating system is also characterised as a centralised system, which integrates different kinds of information, such as vessel arrival times and the availability of resources, as well as creating decisions, such as quay planning and yard space allocation. Based on a review of ICT systems in a PL system (see section 2.6.2), most DMs are supported by a centralised ICT system. However, centralised systems have various drawbacks. First, a centralised decision may be infeasible due to a lack of availability of information at a certain point in time (Moonen, 2009) . Second, a centralised DM may lack practicality as centrally integrating information may be limited due to difficulties with information sharing, costs, and time. Third, a centralised system may lack flexibility. A rapidly changing environment and the increasing complexity of operations mean that continuous upgrades and reconfigurations of DM systems are required, but such changes to a centralised system may be prohibitive (Lang et al., 2008). Though, the use of distributed and hierarchical systems can potentially overcome these drawbacks.

Fully distributed system

In the lower left part of Figure 4.1 there is a fully distributed system. In contrast to a centralised system, fully distributed systems generate information that pertains to a local entity. Each entity may have its objectives or goals to achieve. A complex and large-scale DM problem decomposed into sub-problems, with each being solved locally and independently. The final DM results from the integration of sub-problem solutions. A fully distributed system can be implemented by MAS, which enables autonomous agent decision-making locally. The MAS has been used to solve several DM problems within a PL system, such as operational alignment (A. M. Douma, 2008) and equipment control (Van Dam et al., 2006). However, a fully distributed system also has limitations. First, it may difficult to reach a universal solution because each agent is only interested in its local preference, but the sum of local optimal decisions is sometimes unequal to the global optimum. Second, the DM process in an agent may depend on its information and knowledge, as well as the output from another agent. Thus, synchronisation between independent agents becomes a critical issue within fully distributed systems. The key is to improve control and coordination between agents, and so a hierarchical system can be useful.

Hierarchical system

A hierarchical system is in the middle of the spectrum (see Figure 4.1) . It is widely used in situations where (1) the local agent is not independent, its DM relies on internal intelligence, and the outputs from other entities/agents; (2) a universal decision is generated by mining the result from local entities/agents. The hierarchical system is structured as a multi-layer system. The problem is decomposed into sub-problems and passed to each layer where a decision is made locally. Based on Gershwin (1989), the decomposition includes:

- Decomposition of a process into sub-processes;
- Decomposition of a physical system into sub-systems;
- Decomposition of a complex model (mathematical model) into sub-models.

Through decomposition, decisions are made locally. Depending on the interdependencies of the sub-problems, the interactions between local entities and agents can be complex. To address these complexities, synchronous or asynchronous interactions are required. These types of interactions require coordinated support, which is an interactive process that involves continuous interactions between the local layer and centralised layer until a consensus is achieved. As discussed by Lesser (1999), typical coordination tasks include:

- Identifying which agents can solve a specific sub-problem;
- Determining when a consensus has been reached through the interactions between agents;
- Deciding how and when to solve a specific sub-problem.

Another essential task is to integrate the sub-decisions into a universal solution. The integrations include

- Integrating sub-decisions into a unified solution;
- Integrating the conditions of sub-systems;
- Integrating the outputs from sub-models.

Compared with a centralised system, a hierarchical system can be used to cope with problems that are in nature distribute. Decisions are made locally and integrated centrally. Compared with a fully distributed system, a hierarchical system emphasises control and the coordination of interactions between local entities, which can improve the decision quality. A hierarchical system can be treated as a hybrid distribution system, which coordinates and controls distributed entities. As introduced in section 3.2, a mediator-based MAS is a type of hierarchical system. In a PL system, its typical usage includes operational policy evaluations in container terminals (Henesey et al., 2006) , as well as terminal operational planning and scheduling (Yin et al., 2011).

Discussion

A hierarchical system is suitable for addressing the collaborative planning and reliability assessment challenges facing PL systems for two reasons: (1) both challenges can be decomposed into sub-problems; (2) final decisions rely upon coordination and integration.

- Decomposition: the collaborative planning process can be decomposed into sub-processes. Each sub-planning problem can be made locally. Sub-planning is made based on local information and the information provided by other parties. Similarly, large-scale equipment can be decomposed into sub-systems, components, and sub-processes. To implement a reliability assessment, the condition of each sub-system is assessed locally, and an overall assessment is achieved by integrating the sub-systems conditions.
- Coordination and integration: in order to obtain a globally optimum solution when collaborative planning, proper coordination is needed to align the interests between the self-interested parties involved in the planning process. Similarly, reliability assessments of large-scale equipment require collaboration between sub-systems. Local decision-making depends on the results from other sub-systems.

In this research, a mediator-based MAS is selected to implement the hierarchical system. The meta-heuristic and ontology-knowledge base, which was discussed in Chapter 3, are integrated with the mediator-based system. An integrated DM system will be introduced in the following sections.

4.3 An integrated DM framework

An essential role of a framework is to describe subsystems and their components. Due to the complexity of the addressed requirements and the heterogeneity of the functions, it is difficult to structure all desired functionalities into a single, all-encompassing system. Consequently, a layered approach is used to organise different sub-systems or components. A layered architecture is a system approach aimed at dealing with complexity (Stallings & Manna, 1997). It has the following characteristics:

- Each layer performs a set of functions;
- The upper layer uses outputs/services provided by lower layers;
- Layers are loosely coupled to allow changes in one layer without affecting other layers.

A layered integrated DM framework is proposed, as shown in Figure 4.2. It consists of three parts represented by five layers: (1) a human-machine interface (HMI) module, which is represented by a user interface layer; (2) an agent-based DM module, which is represented by an agent model layer, an agent control layer, and an agent management layer; (3) an information source module, which is represented by a data acquisition layer.

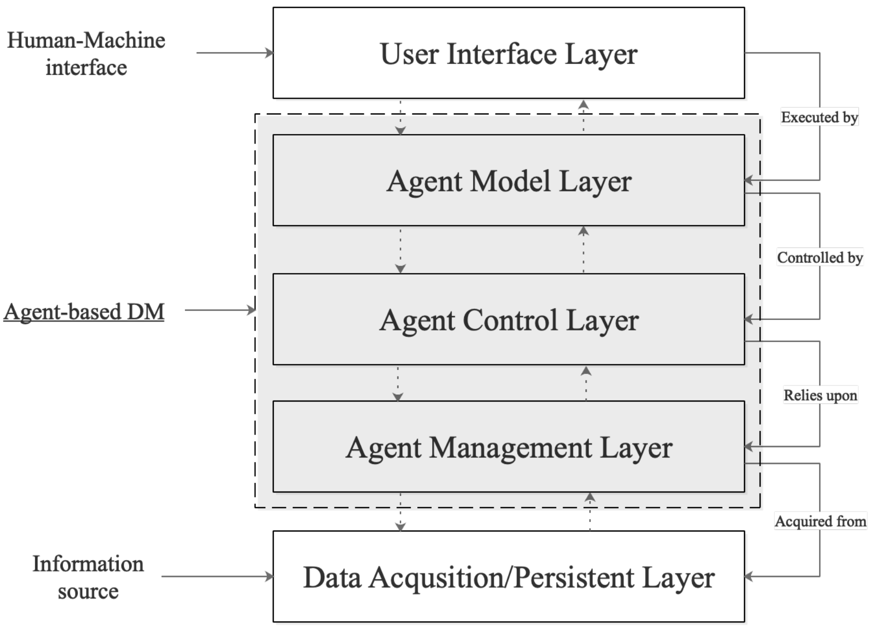

Figure 4.2: A layered integrated framework (a dashed arrow indicates the bidirectional information flow)

The relationship between each layer is also presented in Figure 4.2. The user is interfacing with the integrated DM system through the user interface layer. The requests are executed by the agent-based DM module. Inside the agent-based DM module, the agents are controlled by the agent control layer. The communication and information exchange between different agents is supported by the agent management layer. Additional information is acquired from the data acquisition layer. Therefore, the information flow between different layers is bi-directional.

The functions of HMI and DAC are straightforward as they mainly concern the way that integrated systems interact with the external environment, such as users and physical entities, and how data is persisted in the system. Agent-based DM module is at the core of the system, and each layer implements one of the functions listed in section 4.1. The agent model layer is in charge of request handling through agent communication and interactions (F1). The interactions are controlled and coordinated by the agent control layer (F2). Both agent model and control layer use services provided by the agent management layer. This provides basic services that ensure the efficient execution of the agent system (F3). The functionality of each layer will now be further elaborated in the next section.

4.3.1 Agent-based DM layer

Agent model layer

The agent model layer handles user requests through agent communication and collaboration. In this layer, entities such as users and systems are modelled as the agent. Decisions are made based upon the successful interactions between each agent. Figure 4.3 depicts an agent model and the interaction between two agents. For an agent model, three engines are built to assist dynamic interactions. A computing engine is embedded within an agent to perform the required calculations, such as its current utility. A negotiation engine aims at guiding an agent to interpret incoming information. The intelligent engine is designed to enable an agent with intelligence to make decisions. Besides the three engines, two ontologies are developed to organise the interactions between agents. Personal ontology abstracts the logic and strategic knowledge of an agent. Mutual ontology ensures mutual understanding between agents. It is designed for information sharing and reuse.

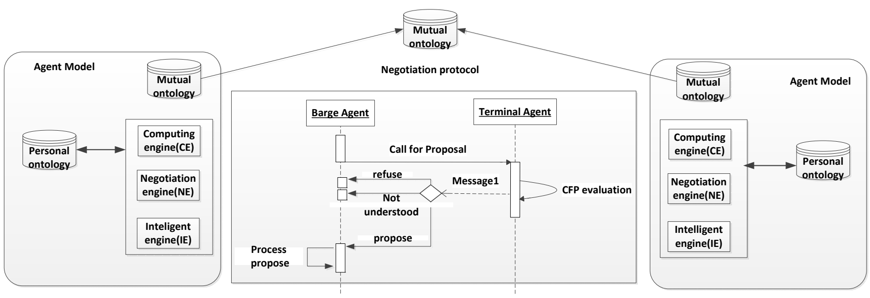

Figure 4.3: Agent generic decision-making model

The user request can be handled by a single agent or a group of agents. When a message is received, the agent will respond by executing its decision functions. The execution is conducted by an agent. Depending on the purpose of the request, the responsibility/functionality of a request can be decomposed into different sub-tasks, where each agent is in charge of one or more tasks. For collaborative planning, it is reasonable to have each joined party modelled as an agent. The planning request is performed by continuous information exchange between agents. Similarly, each component, process, sub-system, and system that requires a reliability assessment can be modelled as an agent. The interactions between each agent can lead to integrated decision-making of the systems conditions.

Depending on the character of the problem, the interactions between agents can be complicated, especially as the interactions at this layer may require a hybrid control strategy to coordinate activities during problem-solving. For example, in the collaborative planning process, each agent may have self-interests, which may conflict with the interests of other agents. Consequently, the output of the solution may not be able to satisfy both parties. Similarly, the reliability assessment task executed by agent A may need the result generated by agent B. Without specific coordination, the execution of agent A may be delayed as the required information may not be delivered by agent B on time.

Agent control layer

The agent control layer is responsible for controlling agent behaviours. It consists of two parts. The first part is a mediator, which is considered to be the system coordinator that controls agent actions in the agent model layer. The need for introducing a mediator agent includes the following: (1) to decide whether an agreement has been reached in a negotiation iterator; (2) to identify which agent can and should solve a specific sub-problem. To perform the coordination task, the mediator agent keeps a list of active agents. It decomposes the incoming tasks into sub-tasks and distributes them to dedicated agents in the agent model layer. Also, it monitors the overall performance of the agents, and coordinates and adjusts the agent strategy such that the final decision is subject to the global optimum.

The second part is an intelligent DM (IDM) approach which provides intelligence support to a mediator agent. The mediator agent interacts with the IDM to acquire knowledge or suggestions. The knowledge or suggestions provided by an IDM can be a choice of assigning which task to which agent, or a decision whether the current negotiation leads to an optimum solution. The reason to set-up an independent IDM rather than embedded with the mediator is to avoid hard-coding complex and dynamic changing logic inside a mediator agent. Different to the agent deployed in the agent model layer which has single responsibility (e.g., diagnosis of a single component), the mediator agent is in charge of controlling all the agent actions. Consequently, its intelligence is complex and subject to change frequently to meet new requirements. The use of an independent IDM can reduce the burden of a mediator agent and enhance system flexibility. Thanks to the development of agent-based middleware (discussed in section 3.2.3) which developed with popular language, i.e., Java , C++, it is possible to achieve such integration from a software perspective. In this thesis, the ICT tools to implement IDM is varied depends on the addressed challenge. For the case of collaborative planning, the meta-heuristics approach are adopted. For the case of reliability assessment, an ontology-based knowledge model is used. Its detailed implementation will be elaborated in the following chapters.

Combined with the role of the mediator agent and IDM, in this research two categories of coordination tasks will be considered at the agent control layer for each challenge:

- Collaborative planning: Resolve potential conflicts of interest by steering agent interactions towards a mutual understanding;
- Reliability assessment: Enable a group of agents to solve a problem efficiently by determining which agents should perform specific tasks and when and to whom they should communicate the result.

Agent management layer

This layer provides fundamental support to the agent execution in the agent model and control layer. For example, a yellow page service F. L. Bellifemine et al. (2007) is used to manage all registered agents. A gateway service F. L. Bellifemine et al. (2007) is designed to route data between different agents. A message transport protocol defines how a message is transported. All of the functionalities as mentioned above are provided by the agent management middleware discussed in section 3.2.3. In this research, the JADE framework is chosen as the agent-based platform. Its usage will be discussed in the following chapters.

4.4 The applicability of Integrated DM framework in PL systems

In this section, the applicability of the proposed integrated DM framework in a PL system will be discussed. The framework, which is shown in Fig 4.2, will be further instantiated by two specific ones that correspond to the challenges discussed in Chapter 2.

4.4.1 An integrated DM framework for collaborative planning

Figure 4.4 depicts an integrated DM framework used to address the collaborative planning challenge, which was discussed in section 2.3. It will be used to solve a practical problem raised by the Port of Rotterdam, namely the hinterland barge transport coordination and planning problem. Figure 4.4 depicts the proposed framework and the functionality of the agent based DM module is given as follows:

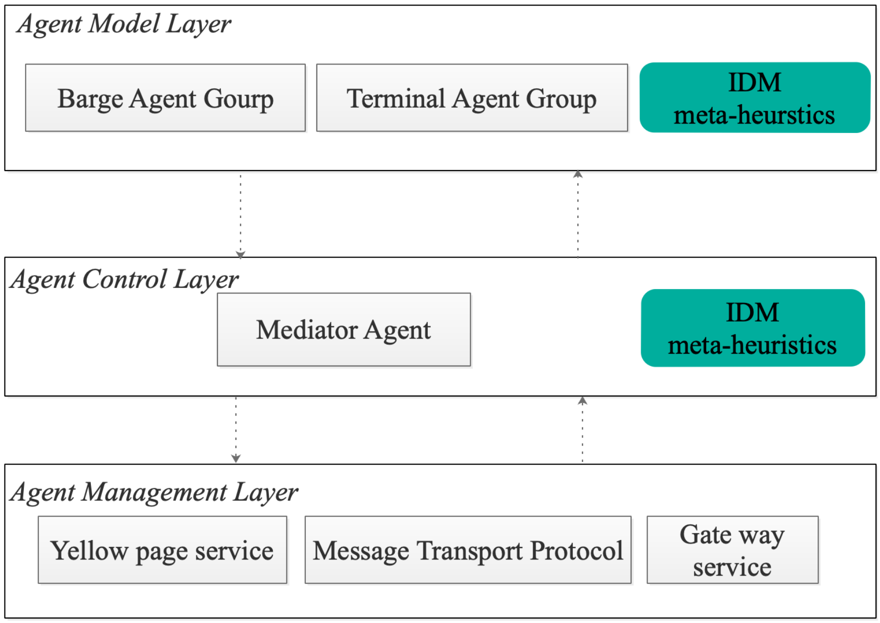

Figure 4.4: An integrated DM framework to achieve collaborative planning: a case of hinterland barge transport

- At the agent model layer, terminal and barge operators are modelled as autonomous agents. A protocol is designed to enable information exchange at this level. Moreover, decision-making algorithms are designed to support agent DM, which aim at maximising agent utilities. Information, such as preferred arrival times of a barge and the possible service time of a terminal, would be continuously exchanged amongst the agents. Upon a successful interaction, a feasible plan can be generated that contains the information, including the barge visit sequence and the terminal service time. The plan will be further assessed by the agent control layer to evaluate quality concerning waiting times and the utilisation of resources.

- At the agent control layer, a mediator is in charge of controlling the negotiation process between each self-interest party. A major role of the mediator agent is to act as a coordinator that accesses and evaluates the plan generated at the agent model layer. If the plan, which is created through agent interaction, does not reach expectations, it generates a command to initiate another round of negotiations. Evaluations and decision-making normally consider a multi-objective problem, and a meta-heuristic optimisation algorithm is developed to resolve the potential conflict of interest.

- At the agent management layer, services include agent message transport protocol and agent management, which are designed to ensure the efficient execution of an agent system.

The proposed framework will be implemented in Chapter 5. Several aspects will be considered to apply the integrated framework. First, the problem will be formulated as a leader-follower model. By analysing the nature and objective of the model, algorithms will be designed. Then, the planning model in combination with the algorithm will be embedded into the agent system. The agent architecture, communication protocol, and the negotiation process will be explained. Finally, the performance of the proposed system will be evaluated, and comparative studies will be presented.

4.4.2 An integrated DM framework for reliability assessment

To address the reliability assessment challenge in a PL system (discussed in section 2.5), an integrated DM framework is proposed in Figure 4.5. It implements a context-aware system, as discussed in section 3.4. The system will be used to facilitate reliability assessments of large-scale equipment by delivering condition assessments, failure predictions, and alarm notifications.

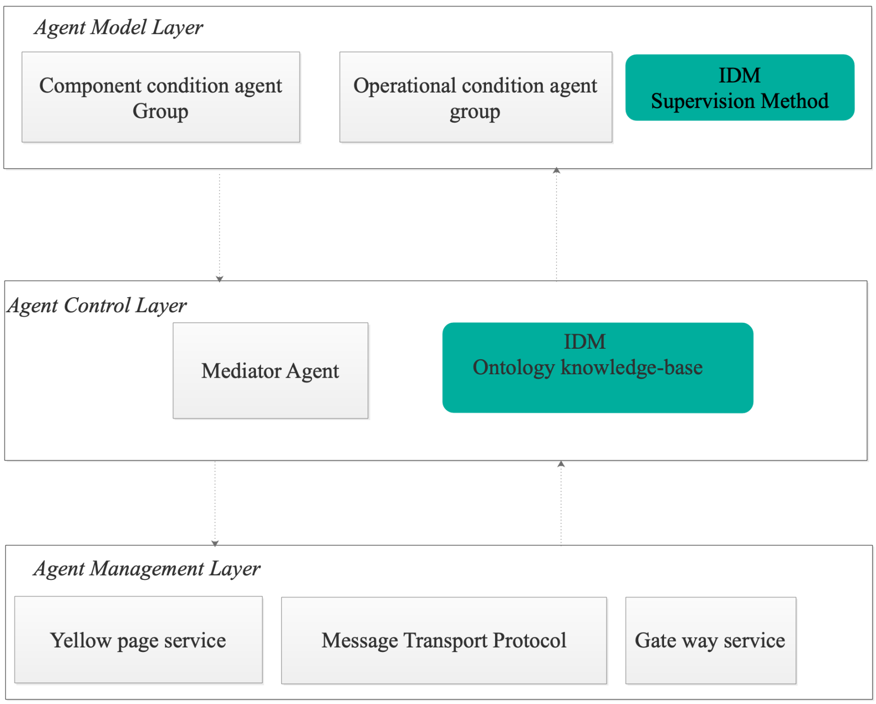

Figure 4.5: An integrated DM framework to achieve reliability assessment: a case of reliability assessments of large scale equipment.

- At the agent model layer, each agent can be designed to represent a process or critical component. Diagnostic and prognostic knowledge is designed and coupled with the agent in order to provide reliability assessment intelligence. Moreover, a condition may relate to several components or processes. In such a case, agents are actively interacting and collaborating to obtain an integrated decision.

- At that agent control layer, a mediator agent is designed as the central coordinator, which bridges the physical object to the agent group. Information is first delivered to the mediator and is then distributed to a dedicated agent for processing. Ontology is used to organise the knowledge that is anticipated in the reliability assessment process. The knowledge base abstracts a specific domain into a context, which is composed of agents and relationships. It connects various aspects, such as fault conditions, diagnosed methods, relevant data, agents and the end-user. Meanwhile, it also possesses reasoning capabilities to facilitate the assessment process. In this research, the designed ontology serves as a knowledge base to provide a common vocabulary of supervision definitions. It is also in charge of controlling the behaviour of an independent agent. The mediator agent has access to the knowledge stored/reasoned from ontology in the forms of data, rules, and restrictions.

- At the agent management layer, services include agent message transport protocol and agent management, which are designed to ensure the efficient execution of an agent system.

In Chapter 6, the reliability assessment challenge is explored by developing a context-aware supervision system based on that depicted in Figure 4.5. The proposed framework will be applied to facilitate the reliability assessments of large-scale conveyor belt systems, which are widely used in bulk terminals. There are four steps for implementing the framework. First, a mediator-based MAS will be designed by decomposing the system into different levels, such as components, sub-systems, and systems. Through agent intelligence and collaboration, integrated decision-making can be achieved. Second, an ontology knowledge base will be developed by considering the entities, relationships, and knowledge anticipated during the reliability assessment process. Third, the ontology knowledge base and MAS should be integrated. Finally, the system evaluation will be achieved using several case studies.

4.5 Conclusion

This chapter answers sub-question 1.3, as listed in Chapter 1 It first assesses different architecture types and concludes that both the centralised and fully distributed systems have drawbacks. The centralised system lacks flexibility, while the fully distributed system lacks the ability for information integration and coordination. A hierarchical system can be adapted to address these challenges, especially as collaborative planning and reliability assessment issues can both be decomposed into sub-problems. Then, decisions can be made to establish the proper coordination and integration of the decomposed sub-problem. Thus, a hierarchical system, namely an integrated DM framework, is proposed.

A layered architecture is also designed to integrate the ICT technologies (introduced in Chapter 3) into a unified framework. It implements a range of functions, including information sharing and exchange, coordination, control over communication processes, as well as supporting decision-making. Specifically, an agent system is used as middleware to facilitate information exchange, negotiation, and collaboration. An agent is represented by different entities in a PL system, such as joined parties and deployed equipment. Different ICT tools are designed based on the nature of the problem, which will assist with agent control and DM processes.

The framework is further instantiated by two specific ones that correspond to the collaborative planning and reliability assessment challenges. For the collaborative planning challenge, a meta-heuristic optimisation engine is developed to resolve planning difficulties during the collaboration stage. For the reliability assessment challenge, an ontology-based knowledge base is developed to assist with knowledge management and the condition assessment process. The implementation details and potential benefits will be further elaborated upon in Chapters 5 and 6.

Chapter 5

A case study of hinterland barge transport planning *

To address the DM challenges in current PL systems, an integrated DM framework was proposed in Chapter 4. In light of this, this chapter assesses the applicability and effectiveness of the proposed framework. This is done by applying it to solve a hinterland barge transport (HBT) planning problem. An introduction of the HBT planning is given in section 5.1. An integrated DM framework is explained in section 5.2 and a decision model is explained in section 5.3. Following this, two collaboration mechanisms are developed in section 5.4. A performance evaluation and comparative study is performed in section 5.5. Section 5.6 presents the system implementation process. A discussion of the contributions of this case study is reflected in section 5.7 and a conclusion is drawn in section 5.8.

5.1 Introduction

In section 2.3, the challenge of HBT planning was introduced. As summarised in section 2.3, the HBT planning process suffers a low level of coordination due to a lack of ICT support. This lead to an ineffective barge rotation plan which results in a long barge waiting time and a low terminal utilization level. To further improve the HBT planning performance, two requirements were identified in section 2.7. First, an information sharing platform is required to enable automatic information exchange and sharing between barge operators and terminal operators. Second, DM approaches are required to support the planning process. The DM approaches should focus on harmonizing the conflict of interests between barge operators and terminal operators.

Based on the requirements, the ICT technologies were selected in Chapter 3. In section 3.1, the MAS technology was selected to support information exchange between

*The content discussed in this chapter has been published in Feng et al. (2014); Feng, Pang, Lodewijks, and Li (2015); Feng, Pang, and Lodewijks (2015a); Feng et al. (2017)

terminal and barge operator. Section 3.3 selected the meta-heuristics as the DM approach to support rotation planning. Further, a hierarchical framework was proposed in section 4.4.1 aims to integrate the selected technology.

Figure 5.1 depicts the integrated DM framework that generalised from the conceptual design in Figure 4.4.

- In the agent model layer, barge and terminal operators are modelled as agents. The agents model the interaction between barge and terminal operators. A simulated annealing algorithm is used by barge agents. It is used to explore the available time windows and generate an optimum plan.
- In the agent control layer, a mediator agent is designed. It is responsible for coordinating and controlling the interactions in the agent model layer, namely, the interactions between barge and terminal agents. A non-dominant sorting generic algorithms (NSGA) is used by the mediator agent. It is used to assess the key performance indicators of the created plan and control the negotiations accordingly.
- The agent management layer provides the facilities to support agent communication. In this case study, the open source software JADE is used. An introduction of JADE has been given in 3.1.2.

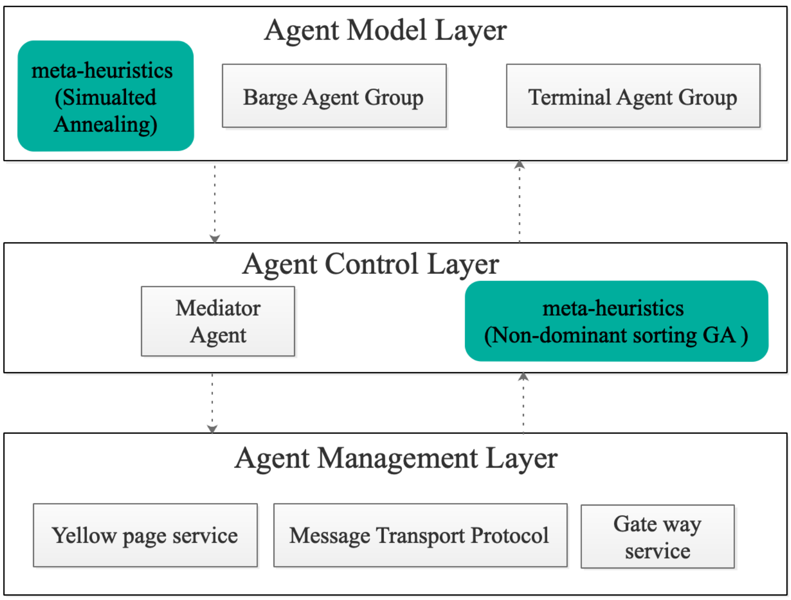

Figure 5.1: The integrated DM framework for the HBT planning, a generalization of the proposed framework depicted in Figure 4.4.

In the following sections, each element will be further discussed. First, the design of the MAS and the meta-heuristics are elaborated in section 5.2. Second, a HBT planning model is presented in section 5.3. Based on the model, two collaboration mechanisms are proposed in section 5.4. The collaboration mechanisms define the way of interaction between barge operators and terminal operators. They further define the role and usage of the DM approach during the interaction. Third, the collaboration mechanisms are integrated with the agent system. A simulation study is performed to evaluate the performance of designed collaboration mechanisms. Finally, several implementation details are explained, and a prototype system is implemented to demonstrate its applicability.

5.2 Implementations of the integrated DM framework

5.2.1 Multi-agent system

Figure 5.2 illustrates a framework of how a mediator-based MAS is used in HBT planning problem. As depicted in the figure, there are three kinds of agent:

- Barge agent (BA): it represents the barge operator to communicate with the terminal operator. The major behaviours include (1) communicate with terminal operators for available service time windows; (2) assess plan performance; (3) confirm the plan with the terminal operator.
- Terminal agent (TA): it is responsible for receiving and processing the request from the barge agent. The primary behaviours include (1) find time slot and communicate with barge agent; (2) assess current berth occupancy level; (3) update its berth plan.
- mediator agent (MA): it is the mediator agent that responsible for monitoring and coordinating the interactions between the barge and the terminal agent. The primary behaviours include (1) collect information regards the barge and terminal utilities which include barge turn around time and terminal occupancy level. (2) evaluate the utilities; (3) determine whether the current plan reaches a satisfactory level.

As shown in Figure 5.2, a barge operator prepares a rotation plan which includes several terminal visits. A barge operator is in charge of communicating with terminal agents for available time windows. A feasible plan is created by aggregating all time windows in a sequence which no two visits are planned at the same period. The barge operator continually assesses the quality of the plan concerning the feasibility and turn around time. If necessary, the terminal visit sequence will be altered to enable better performance. After a better plan is found, the plan will be sent to the coordinator

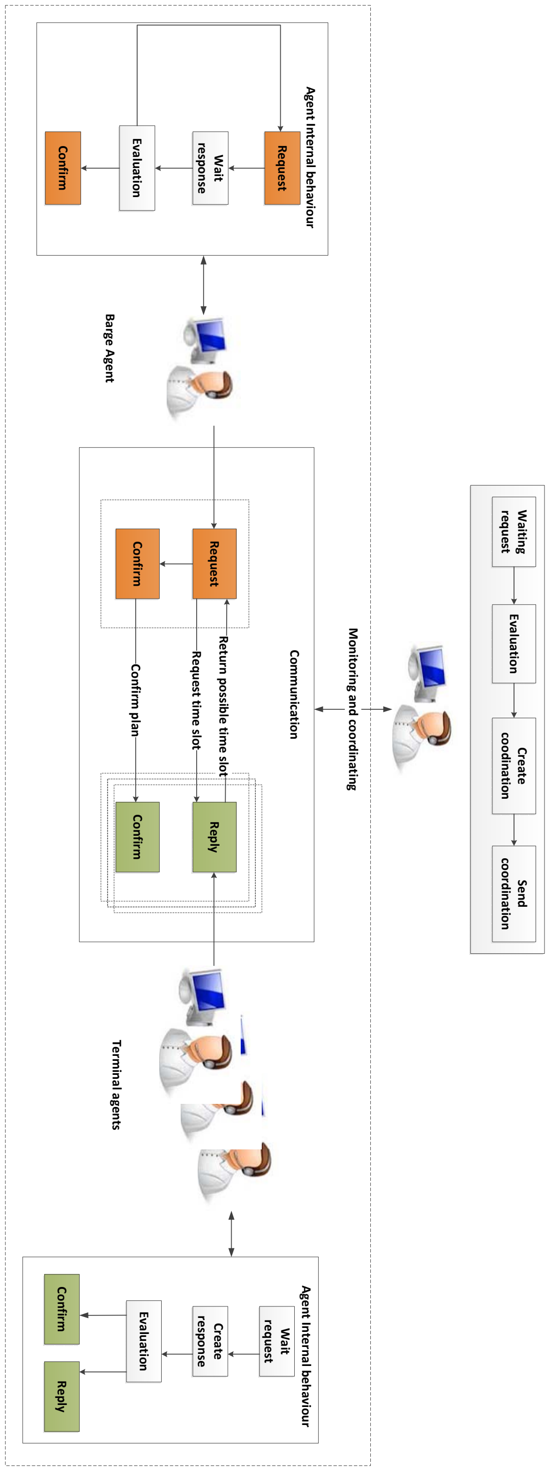

Figure 5.2: An illustration of the agent system

where a higher level evaluation is performed. The coordinator will assess the barge rotation plan from both barge and terminal perspective and may initiate a new round of communication if the current plan is not satisfied, i.e., the plan has long rotation time or low resource utilisation rate.

The purpose of use the agent system is to improve the communication between barge and terminal operator by automating the information exchange process. On top of it, the agent also provides an environment for decision-making. In the next section, the algorithms which will be used to assist agent decision-making will be further introduced.

5.2.2 Decision-making algorithms

In this section, two algorithms, a simulated annealing algorithm and a genetic algorithm, will be introduced. These algorithms will be used to assist the decision-making process. Detailed information about the use of these algorithms will be discussed in section 5.4.

Simulated annealing algorithm

Metropolis et al. (1953) first introduced the simulated annealing process in 1953. Kirkpatrick et al. (1983) further generalised the simulated annealing (SA) algorithm in 1983. This algorithm is inspired by the process of annealing in metallurgy. This involves heating a substance and then gradually cooling it to obtain a high-strength crystalline structure. The SA algorithm is widely used to search for feasible solutions that converge to optimal solutions. For example, it is often used to solve problems associated with vehicle routing, shop flow scheduling, and crew scheduling.

Three terms need to be introduced in order to understand the mechanism of SA, namely, the energy (E), the temperature (T), and the cooling rate (CR).

- Energy: The energy E represents the fitness value or the objective of the problem. In our case, the energy E is the total rotation time of a barge.
- Temperature: The T is used to simulate the heating process. Initially, the T is set high and allowed to slowly decrease or cool as the SA process runs. When T is high, the algorithm is allowed or encouraged to accept less than optimal solutions (compared with the current solution). This enables the algorithm to jump out of local optimums. As the temperature decreases, the chances of accepting less than optimal solutions also decreases. Consequently, it allows the algorithm to focus on a potential search area where an optimum solution can be found.
- Cooling rate: The CR is used to control the convergence speed of running the SA. When a state is generated, the temperature will be decreased based on the

cooling rate. A sizeable cooling rate will lead to a quick convergence but may become stuck at the local optimum. A small number will lead to a long running time. Thus, finding the correct value of the cooling rate is vital for the successful performance of the SA.

The three terms outlined above are summarised in Table 5.1. The energy E is a problem specific value. The energy can be calculated based on the use case, for example, the model for rotation time calculation. The temperature and the cooling rate are algorithm settings variables. To find the best values of T and CR, a parameter tuning is required (see section 5.5.2).

Table 5.1: The terminology for a simulated annealing algorithm

Terminology	Definition	Symbol
Temperature	The temperature is responsible for the correlation between generated solutions and the original solution.	T
Cooling rate	A variable determine how fast the temperature decrease during each iteration.	CR
Energy	A variable represents the state.	E

The process of the SA algorithm is presented as follows:

- Basic iterations: The running of the SA algorithm consists of a number of iterations. The number of iteration is determined by a predefined condition (e.g., whether the system reaches the desired state) or a given time limit. At each iteration, the algorithm will decide to either move the system to a neighbouring state or remain in the current state. The decision is based on the calculation of an acceptance probability.

- Acceptance probability: The acceptance probability is the probability of making the transition from the current state to a new state. The calculation depends on the energy of the current best state E, the energy of the current state E', and the current temperature T. In general, if the energy of the current state is better than the current optimum one ($E' < E$ in a minimization problem), then the system will remain in the current state. If $E' \geq E$, the probability of accepting a worse state is calculated by $e^{(E^{'}-E)/T}$

- Neighbours generation: In order to change a system state from one to another, a new state is required. This is called neighbours generation. The way of altering the current state is termed 'move'. Different moves generate different neighbouring states. The neighbours generation is problem specific. For example, in a vehicle routing problem, the neighbour can be generated by randomly reordering the sequence of two visited nodes.

- Annealing: In SA, whenever a neighbour is generated, the temperature is decreased by a cooling rate. The cooling rate is a parameter which is used to control the annealing process. It is used to update the temperature in every iteration. It can be a fixed value or change adaptively during algorithm runs.

Based on the explanation of the SA algorithm provided above, a schematic of SA is given in Figure 5.3. The left figure shows the process of finding the global optimum by mimicking the annealing process, and the right figure shows the temperature cooling process. At an initial state, high temperature leads to a high acceptance probability. It allows less than optimal solutions to be accepted. This is the state transition from state one to state two as shown in the figure on the left. Alongside the cooling process, the state moves towards the optimum state to eventually find the global optimum which is state four.

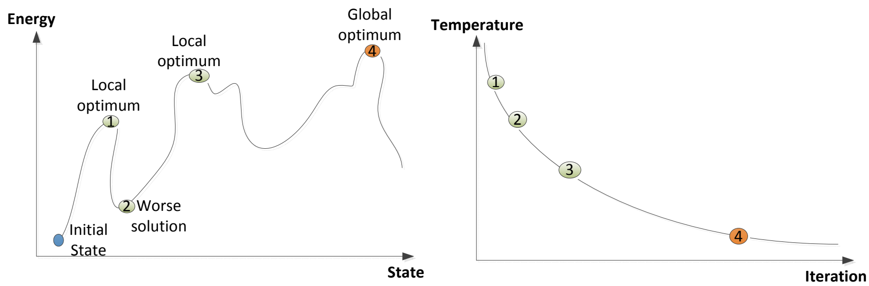

Figure 5.3: A schematic of the SA algorithm

In this case study, the SA will be developed to assist barge operators in finding an optimum terminal visit sequence. It will be programmed as a DM engine that is used by barge agents. The pseudo code of the SA is given in Algorithms 1. Its detailed implementation will be elaborated in section 5.4.

Algorithm 1 Pseudo code of the SA algorithm

Let $s = s_0$
if $n < NumMax$ and $e > emin$ and $T < T_{threshold}$ **then**
  Update T
  Pick a random neighbour, $S_{new} \leftarrow neighbor(s)$
  $eNew \leftarrow E(s_new)$;//compute new sequence energy
  if $P(e, eNew, T) > random(0, 1)$ **then**
    $s \leftarrow s1; e \leftarrow eNew$; //accept and change state
  end if
end if
Output: final state s;

Non-dominant sorting Genetic algorithm II

As introduced in section 3.3, a genetic algorithm (GA) is another meta-heuristic widely used in the optimisation field. The objective of a GA is to maximise or minimise the payoffs of potential solutions in a population against a cost function from the addressed problem (Brownlee, 2011). The general strategy for using a GA is to repeatedly recombine and alternate the candidate solutions, in order to find the optimum fitness value or cost.

In this case study, a specific type of GA, known as non-dominant sorting GA II (NSGA-II), is used. The motivation for this, as discussed in section 2.3, is that the HBT planning problem is by nature a multi-objective problem. The terminal and barge operators have their interests concerning the planning process. The terminal party desires high resource utilisation while the barge party is most concerned about total rotation time. To improve planning quality, both interests should be considered. The goal of a multi-objective problem is to obtain solutions that comprise the Pareto front. The NSGA-II is a multi-objective optimisation (MOO) algorithm that optimises several objectives simultaneously by searching for a set of non-dominated solutions, or the Pareto optimal set (Deb et al., 2002). It is a new version of a multi-objective evolution algorithm (MOEA) which improves computational efficiency and enhances elitism and diversity compared with the non-dominated sorting genetic algorithm (NSGA) proposed by Srinivas and Deb (1994). Recently, the use of NSGA-II to solve MOO problems in transport and logistics has attracted an increasing amount of attentions. For example, Villegas et al. (2006) applied NSGA-II to minimise the cost and maximise coverage for the uncapacitated facility location problem. Ding et al. (2006) applied NGSA-II to assist the MO decision-making process in supporting the design of supply chain networks. Chan et al. (2016) introduced NSGA-II to solve the bi-level truck selection and loading problem to improve company responsiveness.

Typical NSGA-II operations are specified as follows:

(1) Chromosome representation and decoding: The decision variables are encoded as a chromosome which is a float number ranging between $[0,1]$.

(2) Crossover: crossover operation is involved in creating offspring Q_t by selecting and crossing over individuals from parents P_t. In the implementation of NSGA-II, the tournament selection (Madavan, 2002) is used to pick parent chromosome from P_t with a tournament size of 5 and single point crossover is used to generate offspring. During crossover, a random number between [0,1] is generated. If the number is smaller than the predefined probability P_c, the genes from the first parent are copied. Otherwise, the genes from parent 2 are copied.

(3) Mutation: mutation is another fundamental process for evolutionary algorithms. It aims to enhance diversity by allowing a gene to change to a different value. If a randomly generated number is less than the predefined probability P_m, the value of this gene is replaced with a different feasible one. Afterwards, the mutation stops.

(4) Fast-non-dominate sorting: The pseudo code of algorithm 2 explains the sorting process. This is involved when the objectives have been calculated for each. The goal is to set each individual to its corresponding front and assign rank value based on the front which it belongs to. It is a fast sorting process compared with NSGA (Srinivas & Deb, 1994) This is because the information about the set that an individual dominates and the number of individuals that dominates the individual is properly utilised.

(5) Crowded-comparator-operation: In the process of creating new population P_{t+1}, a special operation termed crowded-comparator ($\prec_n$) is used. As described in algorithm 3 line 10, when individuals come from the same rank, then the one with larger crowd-distance will be selected. The method of calculating crowd-distance is described in the work of (Deb et al., 2002) measures how close an individual is to its neighbours. A large crowd distance leads to a better diversity in the population.

(6) Merge operation: A major differences between NSGA-II and other evolutionary algorithms is its operations of recombination and selection. The generation of P_{t+1} is generated by combining P_t and its offspring Q_t using Algorithm 3. The new generation is filled by each front from the combined population until the population size is reached. If the size exceeds the limit by adding all the individuals from the current front, then only partial individuals will be selected based on their crowding distance. In doing so, the elitism and diversity of the population are ensured.

The pseudo of an NSGA-II algorithm is given in Algorithm 4. Initially, a population P_1 with size of n_p individuals are randomly generated. After this, the SA algorithm will be employed to find the route of each barge; (2) the objectives of each individual are calculated; (3) the fast-non-dominate-sorting Algorithm 2 will be applied to sort P_1 into several fronts of non-dominated solutions. An offspring Q_1 of P_1 is generated through a crossover and mutation operation; (4) After obtaining the objectives of each individual in Q_1, it will be sorted by algorithm 2; (5) Finally, the individuals in both P_1 and Q_1 will be merged together to generate the second generation P_2 by using algorithm 3. The given process will be executed iteratively until the predefined threshold is reached.

Algorithm 2 Fast-non-dominant-sorting algorithm

Input: the population P_t;
for each individual p in P_t **do**
  initialise $S_p = \emptyset$; $n_p = 0$ // S_p is the set which contains all individuals that are being dominated by p; n_p is the number of individuals that dominate p)
  for each individual q in P_t **do**
    if p dominates q **then**
      $S_p = S_p \cup q$;// add q to S_p
    end if
    if q dominates p **then**
      $n_p = n_p + 1$;
    end if
  end for
  if $n_p = 0$ **then**
    $F_1 = F_1 \cup p$; //add p to the first Pareto front
  end if
end for
initialise $i = 1$;
while $F_i! = NULL$ **do**
  initialise $Q = \emptyset$; // the set for keeping the individuals in the i_{th} front
  for each individual p in F_i **do**
    for each individual q in S_p **do**
      $n_q = n_q - 1$;
      if $n_q = 0$ **then**
        $Q = Q \cup q$; $q_{rank} = i + 1$;
      end if
    end for
    $i = i + 1$;
    $F_i = Q$;
  end for
end while

Algorithm 3 Merge algorithm

Input: population P_t and offspring Q_t;
initialise $P_{t+1} = \emptyset; i = 1$;
$R_t = P_t \cup Q_t$;
F = fast-non-dominate-sorting(R_t) ;
while $|P_{t+1}| + |F| \leq N$ **do**
  compute-crowding-distance(F_i);
  $P_{t+1} = P_{t+1} \cup F_i$;
  $i = i + 1$;
end while
Sort($F_i, \prec_n$);
$P_{t+1} = P_{t+1} \cup F_i[1 : (N - |P_{t+1}|)]$; //$F_i[1 : (N - |P_{t+1}|)$ choose the first N elements in the front F_i.
return P_{t+1};

Algorithm 4 NSGA-II Algorithm

1: initialise generation: $t \leftarrow 0$, $t \in [0, k]$// k the number of generations
2: generated initial population P_t;
3: compute objective values for each individual $i \in P_t$;
4: fast-non-dominate-sorting(P_t);//non-dominant sort of the generation P_t where $t \in [0, k]$
5: $Q_t = Evolve(P_t)$;//generate Q_t by crossover and mutate P_t.
6: fast-non-dominate-sorting(Q_t);//non-dominant sort of offspring Q_t
7: merge(P_t, Q_t);//generate generation P_{t+1} by merging P_t and Q_t
8: **if** $t + 1 \leq k$ **then**
9:   $t = t + 1$;
10:   go to step 4;
11: **else**
12:   Pareto front set has been found;
13: **end if**

5.3 Planning model

In this section, the planning models incorporated in the HBT planning process will be discussed. In section 5.3.1, the key performance indicators considered in the HBT planning process are introduced. Some general terms will be defined in section 5.3.2. A barge and terminal planning model will be presented in sections 5.3.3 and 5.3.4. Finally, a discussion of the HBT planning models will be presented in section 5.3.5.

5.3.1 Key performance indicators of the HBT planning

To improve the performance of the HBT planning process, it is important to understand the main objective of every participant. This case study considers the two most important actors: the terminal operator and the barge operator.

For a barge operator, according to Groningen (2006), the key objective is to minimise the total turn around time in the port region.

For a terminal operator, the key objective is to maximise the resource utilisation rate. The terminal utilisation rate can be measured in different ways. It can be measured by the average number of crane moves, the speed of a crane move, the number of crew employed, and so on (A. M. Douma, 2008). In the context of HBT planning, how barges are planned will have an impact on terminal resource utilisation. For instance, if barge calls are planned in close succession, then the terminal resource utilisation rate can be maximised for a certain period. In this case, however, a barge may suffer long waiting times. Conversely, barge calls may be randomly distributed across a terminals idle time, and this will reduce the terminal capacity rate. Consequently, how to efficiently utilise a terminals idle time to plan barge calls would be significant for the terminal resource utilisation rate.

Based on this, the two key performance indicators (KPI) considered in the HBT planning process are listed as follows:

- KPI1- Barge total turnaround time: This is the difference between the time a barge enters the port region and the time a barge finishes its last terminal call. A detailed calculation will be given in section 5.3.3.

- KPI2- Terminal resource utilisation: Terminal resource utilisation rate will affect how barges are planned in a terminal plan board. A compactness value will be used to measure how the terminals idle time is used in serving barge calls. A detailed calculation will be discussed in section 5.3.4.

5.3.2 Introduction of general terms in the HBT planing model

In this section, three general terms are defined as follows:

- Barge call profile (bcp): A barge call profile contains three elements: the arrival time into the port region (AT_i), the departure time from the port region (DT_i), and a set of terminal call profile ($TC_{i,j}|j \in N^i$). This is constructed as a tuple. The bcp_i for barge i is represented by $bcp_i = \{\{TC_{i,j}|j \in (1,2,...,N)\}, AT_i, DT_i\}$. Parameter N^i is the total number of terminals to be visited by the barge i and $N^i \in J$. The terminal call profile $TC_{i,j}$ is also represented by a tuple structure as $TC_{i,j} = \{L_{i,j}, UL_{i,j}, AR_{i,j}\}$. Parameter $L_{i,j}$ and $UL_{i,j}$ represent the number of containers to be loaded and unloaded by barge i at terminal j . Parameter $AR_{i,j}$ represents the arrival time of the barge i at the terminal j. An example of a barge call profile is depicted in Table 5.2. Barge 1 has planned a rotation which consisting of two terminal visits (terminal 1 and 2). The arrival time, expected departure time, and terminal call profiles at terminal 1 and 2 are also defined in the terminal call profile.

Table 5.2: An example of terminal call profile of a barge i

Element	*ConFigureuration*	
Arrival time (AR_i)	5	
Departure time (DT_i)	20	
Terminal Call profiles	$TC_{1,2}$	(2,2, 10)
	$TC_{1,4}$	(1,5,30)

- Terminal service profile (tsp): A terminal service profile is defined as a tuple. The format of a tsp is as : $tsp_p = \{\alpha_p, OR_p\}$. Each terminal p has its own tsp_p. The α_p is the current berth occupancy level and the OR_p is the operation time needed for handling the loading and unloading of a container.

- Event profile: An event profile represents the event that has been planned in a terminal plan board. Such an event can be a sea vessel call, a scheduled maintenance, or a terminal close. For each event k in terminal p where $k \in K$, an event profile ($E_{k,p}$) is defined as a tuple structure: $E_{k,p} = OS_k^p, OF_k^p$ where parameter OS_k^p is the start time of event k at terminal p and parameter OF_k^p is the finish time of the event k at terminal p.

In addition to the terms described above, two decision variables are defined as follows:

- Barge visiting sequence: a terminal visit sequence followed by a barge.

- Terminal service time window: a time window which provided by a terminal operator, indicating the time to serve a vessel call.

In light of the parameters and decision variables, the considered HBT planning process is summarised in Table 5.3. Given a barge call profile and a terminal service profile, decisions include a barge rotation sequence and service time window at each terminal should be determined. The decisions should be made such that constraints, such as terminal resource availability, are satisfied. In addition, objectives from both barge and terminal operators should be fulfilled. In other words, it should minimise barge turn around time and maximise of terminal resource utilisation.

Table 5.3: HBT planning problem specification

Planning model
Barge rotation planning process
Given
Planning periods
Barge call profiles
Terminal service profiles
Hard Constrains
Barge rotation constraints
Terminal resource availability
Determine
Barge rotation sequence
Terminal service time window
Objective
minimise barge turn around time
maximise terminal resource utilisation

5.3.3 Barge planning model

In this section, a barge planning model is discussed. Table 5.4 summarises the parameters used in the barge planning model. For a barge i, parameters include: the set of terminals (N^i) to be visited, the number of container to be loaded (L^i_p) and unloaded (UL^i_p) at each terminal p, and the sailing time ($S^i_{q,p}$) between two different terminals p and q. In addition, the arrival time of the barge i in the port region (AT^i) and the latest time they should leave the port region (DT^i), are also given. Parameters a^i_p and b^i_p define the time window provided by terminal p for serving barge i. In the barge planning model, the time window is presented as input. In this case study, it is assumed that each terminal q will be visited by the barge i only once. The total rotation of a barge i is considered to begin when the barge arrives at the at the first terminal, and considered end when the barge leaves the last terminal.

Table 5.5 defines the decision variables that are used in the barge planning model. Variable $X^i_{p,q}$ decides whether the barge i visits terminal q after terminal p. It determines the sequence of a barge rotation in a port region. Variable σ^i_p determines whether the

Table 5.4: Parameters variables for the barge planning model

Variables	*Definitions*
I	a set of barges that are considered in a HBT planning model
N^i	a set of terminals that the barge i has be visited
AT^i	arrival time of i
DT^i	latest departure time of i
$S^i_{q,p}$	sailing time of barge i from terminal p to q
L^i_p	container load number of i at p
UL^i_p	container unload number of i at p
M	a very large number
a^i_p	operation start time of barge i at terminal q
b^i_p	operation end time of barge i at terminal q

terminal p is the first terminal to visit for the barge i. Variable ς^i_p determines whether the terminal p is the last terminal to visit for the barge i. In addition, the arrival time (AR^i_p), operation start time (OS^i_p), and departure time (D^i_p) of the barge i at terminal p needs to be determined.

Table 5.5: Decision variables used in the barge planning model

Variables	*Definitions*
∂_i	turn around time of the barge i
AR^i_p	arrival time of barge i at terminal p
D^i_p	departure time of barge i at terminal p
OS^i_p	operation start time of barge i at terminal p
σ^i_p	binary variable determines if terminal p is the first terminal to visit by barge i
ς^i_p	binary variable determine if terminal p is the last terminal to visit by barge i
D_i	departure time of barge i at last terminal.
$X^i_{p,q}$	binary variable that determine if barge i travel from terminal p to terminal q

Given the parameters and variables, the complete barge planning model is defined as follows:

$$minimise \quad \sum_{i=1}^{|I|} \partial_i; \tag{5.1}$$

subject to.

$$\partial_i = D_i - AT_i, \forall i \in I \tag{5.2}$$

$$AR^i_p + M(1 - X^i_{q,p}) \geq D^i_q + S^i_{q,p}; \forall p,q \in N^i; \forall i \in I \tag{5.3}$$

$$AR_p^i \leq M(1 - X_{q,p}^i) + D_q^i + S_{q,p}^i, \forall p, q \in N^i; \forall i \in I \tag{5.4}$$

$$AR_p^i \geq AT_i - M(1 - \sigma_p^i), \forall p \in N^i; \forall i \in I \tag{5.5}$$

$$AR_p^i \leq M(1 - \sigma_p^i) + AT_i, \forall p \in N^i; \forall i \in I \tag{5.6}$$

$$OS_p^i \geq AR_p^i, \forall p \in N^i; \forall i \in I \tag{5.7}$$

$$OS_p^i \geq a_p^i, \forall p \in N^i; \forall i \in I \tag{5.8}$$

$$D_p^i = OS_p^i + OR_i(L_p^i + UL_p^i); \forall p \in N^i, \forall i \in I \tag{5.9}$$

$$D_p^i \leq b_p^i, \forall p \in N^i; \forall i \in I \tag{5.10}$$

$$D^i = \sum_{p \in N^i} \varsigma_p^i D_p^i; \forall p \in N^i, \forall i \in I \tag{5.11}$$

$$D^i \leq DT^i; \forall i \in I \tag{5.12}$$

$$X_{p,q}^i = \begin{cases} 1 & if\, barge\, i\, visit\, terminal\, q\, after\, terminal\, p \\ 0 & otherwise \end{cases} \tag{5.13}$$

$$X_{p,q}^i + X_{q,p}^i \leq 1, \forall p, q \in N^i; \forall i \in I \tag{5.14}$$

$$1 - \sum_{p \in N^i} X_{p,q}^i = \sigma_q^i; \forall q \in N^i, \forall i \in I \tag{5.15}$$

$$1 - \sum_{q \in N^i} X_{p,q}^i = \varsigma_p^i; \forall p \in N^i, \forall i \in I \tag{5.16}$$

Constraints 5.3 and 5.4 define the arrival time of barge i at terminal q. If barge i sails from terminal q to p, then the arrival time of barge i at terminal p equals the departure time (D_q^i) from terminal q plus the sailing time ($S_{q,p}^i$) between terminal p and q. Constraints 5.5 and 5.6 ensure that if the terminal p is the first terminal to visit by i, then the arrival time equals to the time i arrives at the port region. Constraints 5.7 and 5.8 ensures that the operation start time of barge i at terminal p is large or equal to the provided start time a_p^i from p. Constraint 5.9 defines the departure time of barge i from terminal p which takes into account the operation start time and operation duration time. In addition, constraint 5.10 ensures that the departure time of barge i from p is no later than the time slot b_p^i provided by terminal p. Constraint 5.11 defines the departure time of barge i at the final terminal call, and constrain 5.12 ensures the departure time is no later than the pre-defined deadline (DT^i).

Constraint 5.13 defines the decision variable $X_{p,q}^i$. If barge i travel from terminal p to q, than $X_{p,q}^i = 1$ else $X_{p,q}^i = 0$. Constraint 5.14 ensures that each terminal will be visited exactly once. Constraints 5.15 and 5.16 define the first and last terminals the barge i will visit. Two auxiliary variables are introduced. Variable σ_q^i determines if terminal q is the first terminal to visit. If $\sum_{p \in N^i} X_{p,q}^i = 0$, it implies that there is no terminal which is connected with terminal q, and barge i cannot travel from any other terminals to q. As a result, terminal q will be the first terminal to visit. Similarly, variable ς_p^i determines if terminal p is the last terminal to be visited by i. If $\sum_{q \in N^i} X_{p,q}^i = 0$ which means there is no terminal that is connected with terminal p indicating that p is the last terminal to visited.

To generate a barge visit sequence, the time window provided by a terminal operator has to be determined. In this sense, the barge planning model depends on the terminal planning model. In the following section, the terminal planning model will be presented.

5.3.4 Terminal planning model

Table 5.6 summarises the parameters used in a terminal planning model. Parameter P represents a set of terminals in a port region. Index p represents individual terminal of which $p \in P$. Parameter OR_p represents the operation rate of the terminal p. Parameter E^p represents a set of events that have been planned in a terminal plan board. The events include a regular terminal close, scheduled maintenance, a sea-vessel call, and a barge call. Parameters OS_k^p and OF_k^p represent the start and finish time of the event k in the terminal p.

Table 5.6: Parameter variables in a terminal planning model

Variables	*Definitions*
P	a set of terminals in port region
K	number of planned events in single terminal
OR_p	operation rate, time taken to load or unload a container
E^p	a set of events which have been planned in terminal p
OS_k^p	operation start time of event k in terminal p
OF_k^p	operation finish time of event k in terminal p

Table 5.7 summarises the decision variables used in a terminal planning model. Parameters a_p^i and b_p^i represent the time window provided by terminal p for the barge call i. Variables A_k^p and B_k^p represent the available time slot of terminal i between event k. Variable $l_{k,k+1}^p$ represents the idle time between event k and $k+1$. The definitions of variables α_p, β_p, and C_p are given as follows:

- terminal utilisation level α_p: This is a floating number ranging between 0 and 1. If $\alpha = 0$, it indicates that the barge will be served immediately upon its arrival. If $\alpha = 1$, it indicates the barge operation will be delayed to a maximum amount of time, while operations can still be done within $[A_j^k, B_j^k]$.
- buffer time β_p: This is a fixed amount of time that is added between planned events.
- compactness C_p: This is an indicator that measures the terminal resource utilisation rate. A similar indicator is used in a class-teacher timetabling problem with

Table 5.7: Decision variables in a terminal planning model

Variables	*Definitions*
a^i_p	operation start time of barge i provided by terminal p
b^i_p	operation end time of barge i provided by terminal p
$l^p_{k,k+1}$	idle time between event k and $k+1$ in terminal p
A^p_k	k_{th} available start time in terminal p
B^p_k	k_{th} available end time in terminal p
α_p	terminal utilisation level of p
β_p	buffer time of terminal p
C_p	compactness of terminal p
ϕ^i_p	waiting time of barge i at terminal p

compactness requirements (CTTPCR) (Dorneles, de Araújo, & Buriol, 2014). For CTTPCR, a teacher is giving lectures in several schools. It is important to compact the lessons in each school to allow the teacher presenting courses in different schools. Furthermore, the solution also requires the reduction or elimination of idle periods between lessons in a teacher's schedule in compliance with pedagogical demands or personal preferences. A terminal operator is similar to a teacher that is in charge of serving multiple vessels. For a daily terminal plan, events like scheduled maintenance, closing, and sea vessel service are planned days or even months before the actual date of execution. Consequently, only a limited number of time slots are available for handling barge calls. It is desirable to avoid planning the incoming barge calls where many isolated active periods occurs in the idle period. In other words, a compact plan is desired. To measure how the incoming barge calls are planned in a terminal plan board, a compactness value is introduced. The calculation will be given below.

The primary task for a terminal operator is to determine time windows to serve incoming barge calls. Figure 5.4 presents a simplified schematic of a time window selection process. The first step is to find all the idle periods ($[A^k_j, B^k_j]$) from the current plan board. It is obtained by extracting the time gap between consecutive events (E_k, E_{k+1}) which have already been planned. The event E_k could be a vessel call, maintenance task or terminal closed. They usually have priority over barges, so they are non-changeable. When the available time period ($[A^k_j, B^k_j]$) has been found, the next step is to issue an available time window $[a_{i,j}, b_{i,j}]$ and inform the barge operator. It should be noted that a cluster of idle time periods may exist ($k = 1...N$) and that $[a_{i,j}, b_{i,j}]$ is a subset of ($[A^k_j, B^k_j]$). The remaining step is to formulate the relationship between the idle period and the selected time window. Two scenarios are discussed as below:

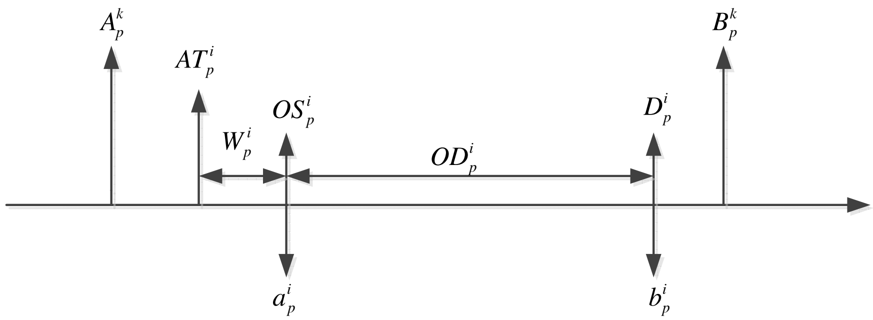

Figure 5.4: An explanation of a terminal operator time window selection

Scenario one

In this scenario, the barge i arrived during operation of event k. The earliest available time is after the operation of event k if the time interval between event k and $k+1$ is sufficient to serve the barge i, as demonstrated in Equation 5.17.

$$\begin{aligned} &A_p^k > AR_p^i \\ &B_p^k \geq A_p^k + OR_p * (L_p^i + UL_p^i) \end{aligned} \tag{5.17}$$

An example of scenario one is depicted in Figure 5.5. A terminal plan board is presented which has a fixed event planned in time slot [0,3]. Two barge calls are available at different times. A call of barge 1 is available at t_k which is expected to arrive on time slot 2. Its operation duration is 1. A call of barge 2 is available at T_{k+1} which is expected to arrive on time 1. Its operation duration is 4. The barges 1 and 2 arrive during the operation of event k. Thus, the terminal operator would push both calls into a waiting queue. The expected waiting time is 1 and 5 for barge 1 and 2 receptively. From a terminal operator perspective, the resource utilisation rate is reaching an optimal level because the resources are fully occupied in the period [0,7].

Based on the above discussion, the time window $[a_p^i, b_p^i]$for serving the barge i at terminal p is calculated by Equation 5.18,

$$\begin{aligned} &a_p^i = OF_k^p \\ &b_p^i = a_p^i + OR_p * (L_p^i + UL_p^i) \end{aligned} \tag{5.18}$$

Scenario two

In this scenario, a barge is expected to arrive at a terminal in the middle of one idle time period and the idle time window $[A_p^k, B_p^k]$ is sufficient to serve the barge i. This is

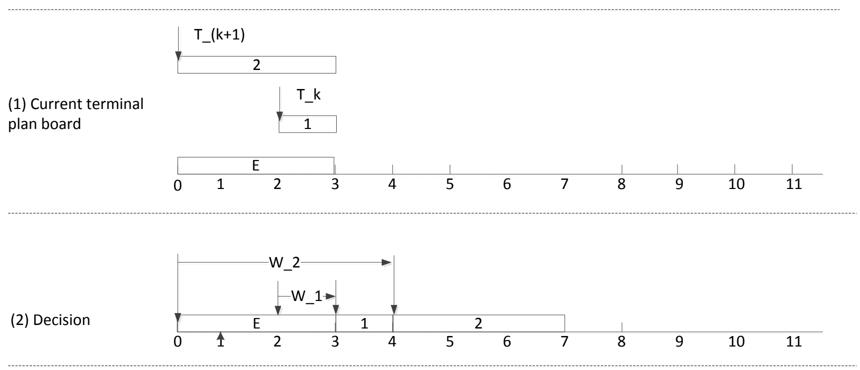

Figure 5.5: Time window selection scenario one

represented in Equation 5.19.

$$\begin{aligned} &A_p^k \leq AR_p^i; \\ &B_p^k > A_p^k + OR_p * (L_p^i + UL_p^i) \end{aligned} \tag{5.19}$$

In this scenario, decision-making becomes complex because multiple choices are available for the terminal operator. One option is to serve the barge upon its arrival which will lead to a minimum waiting time. Alternatively, the terminal operator could also postpone the service to a certain extent in order to improve the resource utilisation rate. Both choices are explained in Figure 5.6.

As shown in Figure 5.6, a fixed event has been planned in time slot [5,8] on the terminal plan board. Two barge calls are available at different times. A call of barge 1 is available at t_k which is expected to arrive at time 2 and has an operation time of 1. A call of Barge 2 is available at T_{k+1} which is expected to arrive at time 1 and has an operation duration of 4. Two possible ways of issuing the time windows are discussed below.

- Decision-1: With current terminal plan board, the barge operator 1 will have the chance to immediately plan its call immediately after its arrival and leave the terminal at time 3 (the operation will occupy the terminal in the period of $[2,3]$). By evaluating the current plan board, the operation of barge 2 can be planned to follow the event E, and it will occupy the terminal in the period $[8,11]$. As such, the start time of barge operator 2 is delayed by 8 time units. The barge operator 2 has a higher chance of cancelling the current call and seeking another visit sequence that yields a shorter waiting time.
- Decision-2: The terminal operator prefers to keep its resource (e.g., equipment and working crew) working during a period to increase its resource utilisation.

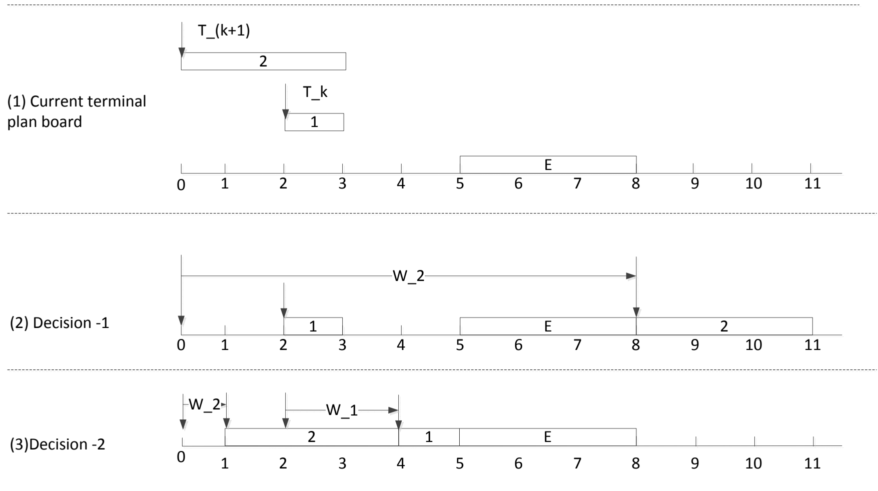

Figure 5.6: Time window selection scenario two

> A compact plan in which events are tightly planned is preferable. As presented in Figure 5.6 (3), if the terminal operator postpones the operation of barge 1 for 3 time units and starts at 5, and postpones the operation of barge 2 for 1 time unit which starts at time 2, then the call of barge 1 and 2 will be finished at time 5. Decision-2 yields a better result compared to decision-1 in which the call of barge 1 and 2 finish at time 11. In addition, the resource utilisation of the terminal in decision-2 is higher compared to decision-1. In the period of [1,8], the decision-1 yields a resource utilisation rate of 50% whereas the decision-2 yields a resource utilisation of 87.5%. Moreover, as a result of the compact plan, the remaining capacity of the terminal can be used to serve more vessel calls. It can be seen from the Figure 5.6 that the period $[8,11]$ is free compared to decision -1.

Therefore, the time windows generated by decision-2 (i.e., [4,5] for barge 1 and [1,4] for barge 2) is favourable to a terminal operator. On the one hand, it yields a better resource utilisation rate in the period of [1,8]. On the other hand, it enables the terminal operator to accept more vessels service requests. This means the period of [8,11] is free compared with decision -1 which is occupied by a barge call. This is important for a busy terminal with a number of service-requests during a specified period.

To generate an uniform decision, the control variable named berth occupancy level (α) is used. Figure 5.7 gives a demonstration of how variable α controls the time window selection process. Given a barge i arrives at a terminal j at $AR_{i,j}$, an idle time window ($l^p_{k,k+1}$) between the barge and nearest event E_{k+1} is calculated by Equation 5.20 . After this, the $\phi^\star$ is updated to ϕ by multiplying the berth occupancy level α (see Equation 5.21). The ϕ^i_p is the waiting time of barge i at terminal p under current berth occupancy level α_p. Following this, the time window is selected by applying Equation 5.22.

$$\phi_p^{i\,\star} = B_k^p - AR_p^i - OR_p * (L_p^i + UL_p^i), \tag{5.20}$$

$$\phi_p^i = \phi_p^{i\,\star} * \alpha_p \tag{5.21}$$

$$\begin{aligned} a_p^i &= AR_p^i + \phi_p^i \\ b_p^i &= a_p^i + OR_p * (L_p^i + UL_p^i) \end{aligned} \tag{5.22}$$

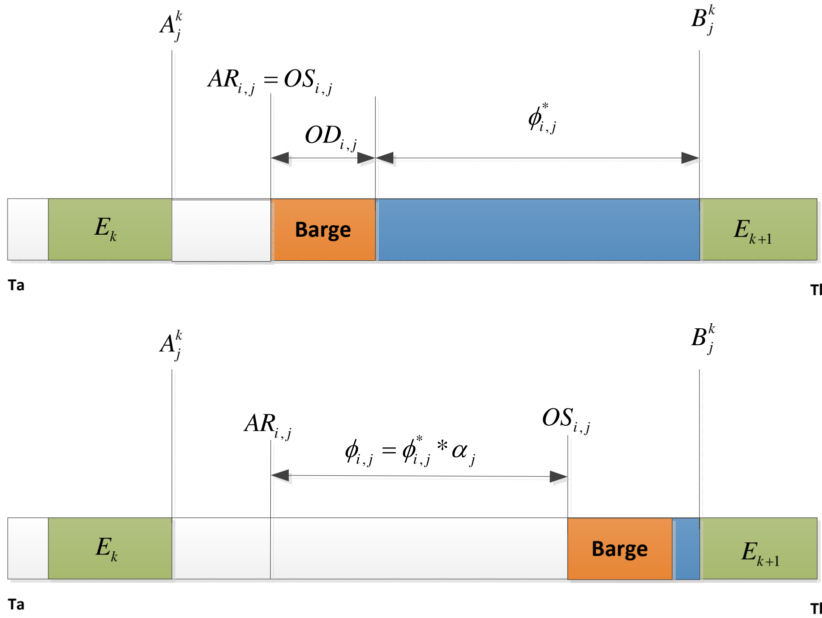

Figure 5.7: The barge service postponement mechanism

Terminal objective

The remaining task is to quantify the terminal objective in these two different scenarios. First, the concept of gap time $GT_{k,k+1}$ is explained. This calculation is shown in Equation 5.23. This is used to measure the time difference between two consecutive events, E_k and E_{k+1}.

$$GT_{k,k+1} = OF_{k+1}^p - OS_k^p; \forall k \in E^p \tag{5.23}$$

To quantify the terminal utilisation rate, an indicator compactness value C_p is introduced . The calculation can be found in Equation 5.24. This is obtained by taking the reciprocal of gap time value and controlling it using variable β_p. If the value is under β_p, it is treated as a buffer time added between two events. A relatively large value could result in a waste of resources because no more events can be inserted in the gap. As such, the smaller the C_p is, the better the terminal utilisation is.

$$C_{k,k+1} = \begin{cases} 1/GT_{k,k+1} & if\, GT_{k,k+1} > \beta_p \\ 0 & otherwise \end{cases} \tag{5.24}$$

Finally, the objective of the terminal operator is to maximise the terminal resource utilisation by keeping all planned activities as compact as possible. This can be translated to minimising the compactness value for all terminals in a port region. This is calculated by Equation 5.25

$$minimise \quad \sum_{p=1}^{P} C_p \tag{5.25}$$

where the compactness value is calculated in Equation 5.26,

$$C_p = \sum_{k \in E^p} C_{k,k+1} \tag{5.26}$$

5.3.5 Discussion

The barge and terminal planning models are presented in the sections 5.3.3 and 5.3.4. In this section, a discussion on the model's complexity is given. Following this, the necessity of collaboration will be discussed.

Model complexity

The barge planning model, which is presented in section 5.3.3, is formulated as a mixed integer linear programming (MILP) model. The objective is to determine a barge visit sequence in order to minimise the total turnaround time. The turnaround time of a barge is influenced by two decision variables: a visiting sequence and the time windows provided by terminal operators. If the time windows ($[a^i_p, b^i_p]$) are fixed, the MILP model can be treated as a vehicle routing problem with time windows (VRPTW). In a VRPTW, a set of customers need to be visited by a set of vehicles to deliver packages. A time window is used to limit the time frame in which a vehicle arrives at each customer location. A vehicle may arrive before the time window opens. However, the customer cannot be serviced until the time window opens. By adding the time windows constraints, a VRPTW is given. The similarity between the VRPTW and the HBT is given that for any given number of barges (vehicles), and terminals (customers), routes (the sequence of terminals to be visited) needs to be determined that result in the total turnaround time of barges, and the total capacity utilisation of terminals are optimised. The VRPTW is categorised as NP-hard combinational optimisation problems (Lenstra & Kan, 1981; Solomon, 1987). The HBT is more complex than the VRPTW because the time window provided by terminals varies. In light of the planning model discussed in section 5.3.4, it can be found that a time window provided by a terminal operator depends on two factors: a barge call profile and its resource utilisation level. Consequently, different barge call profile and preference of resource utilisation will impact the time window. In other words, decisions made by a barge operator and a terminal operator depends on each other.

The complexities of the HBT planning model make it difficult to apply an exact method to solve the model and obtain an optimal solution (see section 3.3). For such a complex problem, the use of meta-heuristics is more suitable (see discussion in section). In section 5.2.2, two meta-heuristics are introduced. In section 5.4, the details of applying the methods will be elaborated on.

The necessity of collaboration

Both barge and terminal operators have their objectives and aim to have their interests satisfied. However, these two objectives are difficult to improve simultaneously. This is because the improvement of one objective could be obtained by sacrificing the other.

As discussed in section 5.3, a barge operator tends to aim to find a rotation plan in which the turnaround time is minimal. This can be achieved by considering the minimum waiting time at each terminal. As presented in section 5.3.3, the objective of a barge operator is influenced by the time window provided by terminal operators. If the barge call at each terminal can be served immediately without any waiting time (e.g. arrival at a terminal idle period), then the turnaround time can be optimised. However, since the terminal plan board is not shared with the barge operator, the barge operator has no control over the time window selections.

Moreover, as discussed in section 5.3.4, the goal of a terminal operator is to maximise its resource utilisation rate. To achieve this goal, a compact plan is desirable. A compact plan can be achieved by selecting a time window in which resources are kept busy in a certain period. The consequence of this is that the barge call will be placed in a queue, leading to long waiting times.

In order to improve the performance of the HBT planning process, collaboration mechanisms should be designed to enable a more efficient planning process. In the following sections, two collaboration mechanisms are designed. The interaction between agents, the coordination procedures, and the decision-making algorithms will be specified.

5.4 Collaboration mechanisms

In the previous section, the necessity of collaboration between a terminal and a barge operator was discussed. To achieve this, collaboration mechanisms are required. These mechanisms should consider two aspects: information exchange and coordination. In this section, two collaboration mechanisms are proposed. For each mechanism, a collaboration framework is proposed. Following this, the decision-making process will be explained.

5.4.1 Collaboration mechanism 1

In collaboration mechanism 1, the interaction between a terminal and a barge operator is modelled on the leader-follower Stackelberg game. The Stackelberg game is a two-person game proposed by economist Von Stackelberg (Von Stackelberg, 1952) wherein one person (the leader) is able to impose his decision on the other person (the follower(s)). It is a tool designed to analyse the interaction of strategic decision makers. It has been extensively applied in economics, politics, computer science, and biology. Recent research outputs reflect the potential use of the games theoretic approach to solving complex engineering problems, especially for decision-making and multi-objective optimisation (Ohazulike & Brands, 2013; Y. Liu et al., 2013; Chu et al., 2015). In general, several well-established two-player game models are available. These include the Stackelberg, Cournot, and Bertrand models. The major difference between the Stackelberg model and the other models is the timing properties (Bode & Ferreira, 2014).

In a Stackelberg model, players make moves in a sequential manner. Different types of competition are engaged in the Stackelberg game if one player has an advantage which enables it to move first and force followers to take actions sequentially. In the HBT planning process, the conflict of interests between terminal and barge operators is by nature a leader-follower competition in which the terminal operator (leader) possess an advantage by issuing time windows to barge operators (followers). Following this, barge operators will obtain the time window and make decisions by sequencing the best visit in order to obtain their objective. The two-stage decision-making process best matches the Stackelberg model. Moreover, the sequential movement enables an ideal information sharing structure for both players (Dai & Gao, 2014). This is an ideal context for the negotiation between barge operators and terminal operators because both players need the others' decision variables to make a move. In contrast, decision-making in the Cournot and Bertrand models are independent, and information sharing depicts an incomplete structure. Based on this, it is reasonable to select the Stackelberg model for this research.

Figure 5.8 presents a leader-follower interaction structure for the HBT planning process. In the HBT planning process, a terminal operator is a leader, and a barge op-

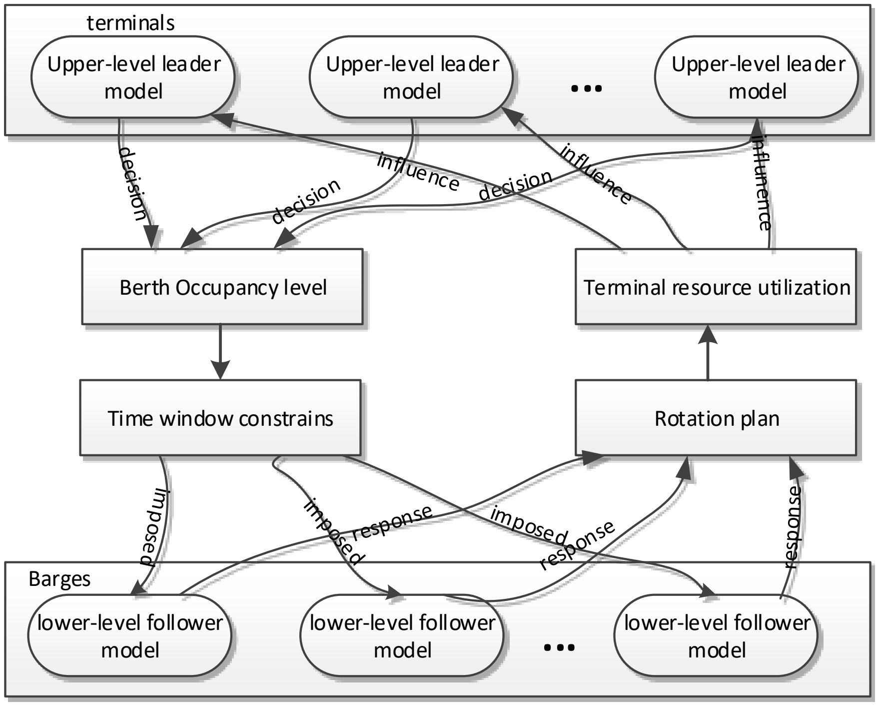

Figure 5.8: The framework of collaboration mechanism 1

erator is a follower. In the model, multiple terminal operators make decisions simultaneously. This leads to a multi-leader structure in the upper level. In the process of decision-making, the terminal operator will first move by choosing decision variables in an attempt to guarantee the resource utilisation level. After they are provided with time window constraints, the follower will respond by seeking a route to obtain the best turnaround time. The generated route plan will be inserted on the terminal plan board, thus imposing their influence on the terminal utilisation level concerning compactness value (equation5.24). The smaller the compactness value, the better the terminal berthing capacity utilisation. As a consequence, the leader can update its berth occupancy rate, forcing the lower-level barges to respond again.

The fundamental of enabling an efficient leader-follower coordination is that the utility value of each game player is transferable. To be exact, the one with a high utility value can compensate the other one which has a low utility. In our case, the leader makes a decision that results in an improved compactness condition which could lead to a long turn rotation time for the follower. Thus, in the following negotiation round, the leader can reduce its utility to compensate the follower's interest. It should be noted that this structure leads to a more complicated iterative negotiation. This is because both leaders and followers have no guarantee of being able to find its best strategy to ensure global optimality in one negotiation phase.

To enable collaboration between the terminal and barge operator, the mediator-based agent system (discussed in section 5.2) will be applied. In section 5.2, the responsibility of each agent is briefly explained. In the following sections, the agent intelligence and decision-making process will be further elaborated on.

Barge Agent (BA)

BA is a type of agent type, and it stands for a barge operator. Figure 5.9 depicts an activity diagram of a BA agent. It includes two building blocks:(1) negotiation; (2) decision-making.

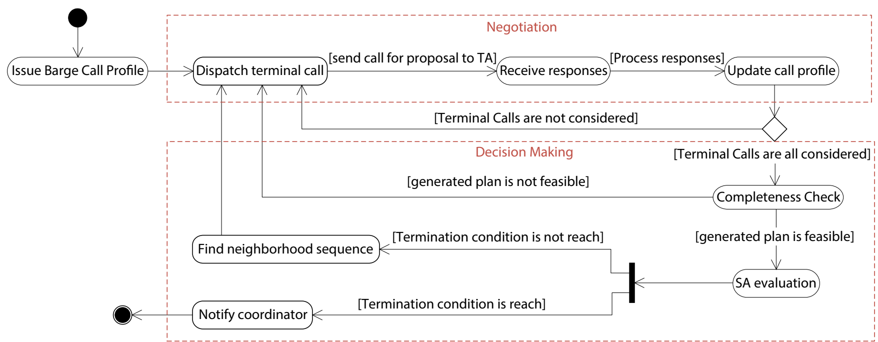

Figure 5.9: A barge agent activity diagram

Negotiation: As shown in Figure 5.9, a BA sends a terminal call profile to targeting TAs one by one. A terminal call profile includes the expected arrival time at terminal p, and the number of containers to be loaded and unloaded. The response from the TA will include the available time window for the incoming barge call. This will be sent back to the BA and aggregated and evaluated by BA. Negotiations will continue until all terminal calls $TC_{i,j}$ are considered.

Decision-making: As shown in Figure5.9, the decision-making include two parts namely feasibility check and solution evaluation.

- Feasibility check (FC): A rotation plan is checked to ensure: (1) the "double-booked" scenario is avoided; (2) the last terminal call finishes before the defined deadline. The FC can be implemented by applying Algorithm 5.
- Solution evaluation (SE): The SE is used to assess the quality of the generated rotation plan. The input is the current terminal visit sequence. By calculating the current turn around time, the SE can decide whether a new negotiation is required. If necessary, a new visit sequence is generated by alternating the previous one. In this research, the simulated annealing (SA) algorithm is used to assist the SE process. The SA was introduced in section 5.2.2. Its design in

Algorithm 5 Solution Evaluation and decision-making for bcp_i (Simulated Annealing Process)

Input: ;
for ALL p in P **do**
    if $\varsigma_p^i = 1$ and $D_p^i > D_i$ **then**
        Return false;
    end if
    if $X_{p,q} = 1$ and $AR_q^i > D_p^i$ **then**
        Continue;
    else
        Return false;
    end if
end for

SE is discussed in Algorithm 6. As discussed in section 5.2.2, the term *energy E* represents the the turn around time of the current rotation plan. The total rotation time of the newly generated plan (which is represented by *eNew*) is compared with the best rotation time obtained so far and accepted as the best solution with a probability p. An improved solution is obtained by considering the neighbourhood of the current plan sequence s, namely, reordering the visit sequence $neighbourhood(s)$. The stop criterion is given when (1) the pre-defined control variables (which is represented as temperature T) reaches a pre-defined threshold;(2) the best rotation time cannot be improved during a pre-defined iteration. Upon the termination of evaluation, a decision regarding whether a re-plan is needed will be made. If necessary, a new round of negotiation will be initiated. The negotiation will continue until a suitable solution is found.

Algorithm 6 Solution Evaluation and decision-making for bcp_i (Simulated Annealing)

Input: terminal visit plan for bcp_i, denote as $s1$;
Output: Decision for rescheduling($reSchedulelDecision$) and new visit sequence bcp_i^*;
initialisation: $s \leftarrow s0; e \leftarrow E(s)$;
$T_0 \leftarrow 100; n \leftarrow 0; coolingRate \leftarrow 0.01$ //initial state and energy(turn around time for bcp_i)
if $n < NumMax$ and $e > emin$ and $T < T_{threshold}$ **then**
    // while counter not reach maximum and energy is not reaching minimum;
    $eNew \leftarrow E(s1)$;//compute new sequence energy
    if $P(e, eNew, T) > random()$ **then**
        //calculate the probability of accept new solution
        $s \leftarrow s1; e \leftarrow eNew$;
        accept and change state
        $n \leftarrow n+1$; //update iteration counter
    end if
    $RePlanDecision \leftarrow true$;// reschedule is required
    $T \leftarrow temperatureUpdate(coolingRate)$;//update temperature by cooling rate
    $newSequence \leftarrow neighbour(s)$;//pick neighbourhood from input solution
else
    $RePlanDecision \leftarrow false$; //solution is validated and optimised, negotiation ends
end if
Return $RescPlanDecision$ and bcp_i^* with $newSequence$;

Terminal Agent (Upper layer)

For the terminal agent (TA), its corresponding obligation is to assign resources and periods for severing pick-up and delivery activities at minimum cost and maximum utilisation of resources. As shown in the activity diagram 5.10, the chosen actions are related to the incoming message type. Typical activities include process BA calls, evaluation of current berth capacity utilisation, and updates regarding its strategies. This section will explain how the TA makes decisions to handle BA calls and how berth capacity is evaluated.

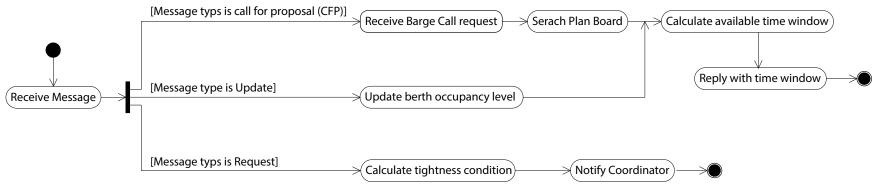

Figure 5.10: A terminal agent activity diagram

The decision-making process of a TA includes four phases:

- *Find available time window:* When the TA receives the barge call, the decision-making process is activated. The first step is to calculate the corresponding idle period from the current plan board.
- *Find available time window for barge handling:* A list of available time windows created in phase 1 will be evaluated. It aims to target the appropriate windows for the incoming request.
- *Applying berth occupancy level:* The given time window selected in phase 2 will be further processed by taking into consideration the berth occupancy level α_j. In doing so, the best-inserted position is found in favour of the terminal's objective. During the negotiation, the α_j will be updated accordingly. This update will be instructed by the mediator agent. Details of this will be covered in a later section. The update procedure is shown as follows:

$$\alpha_j^{\star} = \alpha_j * (1 - \theta) \tag{5.27}$$

  where θ is constant value specify the step size of the update process.
- *Issuing time window:* The final step is to package the created information and send it back to the corresponding BA.

Mediator Agent (MA)

The major task of an MA is to provide instructions to coordinate the generated plan and achieve a globally optimal solution. In collaboration mechanism 1, the search for the feasible space is controlled by the utilities of BA and TA. As shown in Figure 5.11 below, the red arrow represents the inner loop where the TA and BA interact to create

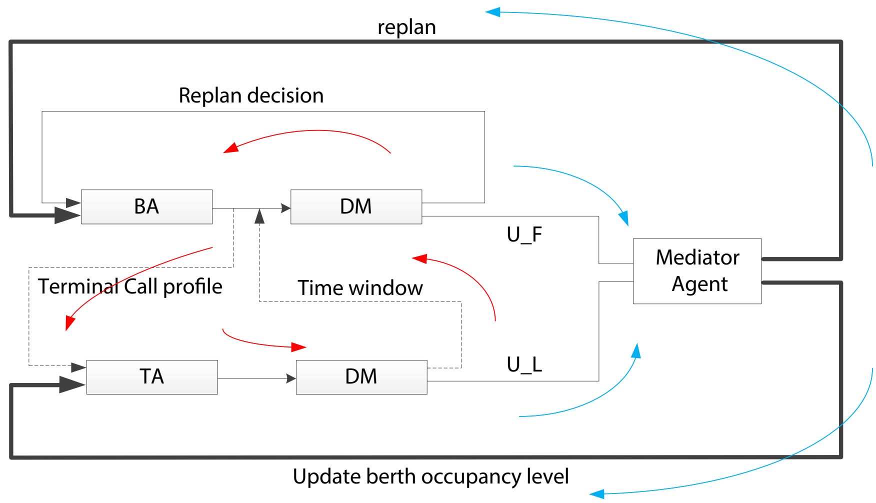

Figure 5.11: The leader-Follower decision-making with a mediator coordination

a plan. The blue arrow represents the outer loop where mediation is introduced to assess multiple objectives and provide guidance if the objective is not optimised. The decision-making details are presented as follows:

x_1 denotes as the strategy of the upper-level leader. This is referred to as the berth occupancy level.
x_2 denotes as the strategy of the lower-level follower and is referred to as the rotation sequence of barges.
$x_1\star$ denotes as the best strategy of the upper-level leader
$x_2\star$ denotes as the best strategy of the lower-level follower
$U_L(x_1,x_2)$ denotes as the utility function of the leader by taking x_1 and x_2 into account. This is calculated as follows:

$$U_L(x_1,x_2) = (\sum_{p \in P} C_p)/|P| \tag{5.28}$$

where $|P|$ is the total number of terminals, C_p is obtained in Equation 5.24.

$U_F(x_1,x_2)$ denotes as the utility function of the follower by taken x_1 and x_2 into account, it is calculated as:

$$U_F(x_1,x_2) = (\sum_{i \in I} \partial_i)/|I| \tag{5.29}$$

where $|I|$ is the total number of barge calls, ∂_i is calculated in Eq 5.2.

$D_F(x1,x2)$ denotes the decision strategy chosen by the follower. This takes the decision of the leader into consideration, where $x_2 \subseteq D_F(x1,x2)$.

The optimal strategy of follower is obtained when:

$$D_F(x_1,x_2) = \{x_2^\star | U_F(x_1,x_2^\star) \geq U_F(x_1,x_2) \\ ,x_1 \subseteq D_L(x_1,x_2)\} \tag{5.30}$$

$D_L(x1,x2)$ denotes the decision strategy chosen by leader which taken the decision of follower into consideration, where $x_1 \subseteq D_L(x1,x2)$.
The Stacklberg equilibrium is obtained as $D^\star = (D_L^\star, D_F^\star)$ when:

$$D_L^\star(x_1^\star,x_2^\star) = \{x_1^\star | U_L(x_1^\star,x_2^\star) - U_L(x_1,x_2^\star) \leq \varepsilon_L, \\ x_2 \subseteq D_F(x_1,x_2)\} \tag{5.31}$$

$$D_F^\star(x_1^\star,x_2^\star) = \{x_2^\star | U_F(x_1^\star,x_2^\star) - U_F(x_1^\star,x_2) \leq \varepsilon_F, \\ x_1 \subseteq D_L(x_1,x_2)\} \tag{5.32}$$

where ε_F and ε_L is a small number to identify the convergence.

The strategies of the leader and follower will be converged to create an equilibrium whereby neither the leaders nor the followers utility can be further improved by sacrificing others. In other words, no strategy can dominate the decision-making process, and the final solution is beneficial for both.

Integrated decision-making

In this section, the integrated decision-making process is outlined. During the process of decision-making, a follower problem is linked to the leader via the berth occupancy level (this will transfer to available time windows) and operation rate. Since the operation rate is assumed to be a constant number, the interface will lead to a straightforward collaboration mechanism between the leader and the follower. The leader has full knowledge of the response of the follower. Therefore, it can adjust its strategy accordingly. As a result, by updating the berth occupancy level, the leader can guide the followers to optimise their objectives. The decision-making process is summarised in Algorithm 7. The algorithm includes four phases:

- Phase 1: After acquiring the information about barge call profiles and terminal service profiles, an initial feasible plan can be generated for each bcp_i. The generated plan will include the decision variables of visit sequences and start and departure times.
- Phase 2: In every iteration loop(lines 5-8), a feasible plan is generated. The evaluation process is then activated. If the decision variable *RePlanDecision* is true. then a newly created sequence will be attached to the decision, and a new round of *phase 1* will be initiated.
- Phase 3: When all followers in set *BCP* have found their best strategies (denoted as the optimal strategy of followers under the current strategies of the leaders). The global objectives are calculated in this phase.

Algorithm 7 Solution algorithm

1: Input: *BCP* : a set of barge call profiles. bcp_i: single barge call profile contains a set of terminal call TC_j, where $j \subseteq [1,2,...,N]$ and N is the total number of terminal calls of bcp_i. ε_L and ε_L optimisation threshold; θ the step size of α.
2: Output: An ordered terminal visit sequence for each bcp_i.
3: initialisation: α : berth occupancy level for terminals, m a game play stage counter, where start from 1;
4: **for** All bcp_i in *BCP* **do**
5:     **for** All TC_j in bcp_i **do**
6:         *//Phase 1: Find feasible visit sequence for each* bcp_i.
7:         Find available time window from terminal j;
8:     **end for**
9:     *//Phase 2: Evaluate the generated plan for each* bsp_i *using Algorithm 6.*
10:     Evaluate rotation plan performance, return decisions;
11:     **if** $RePlanDecision \neq false$ **then**
12:         acquire update call sequence bcp_i to $bcp_i^{\star}$ and go to step 5;
13:     **else**
14:         plan is validated for bcp_i, continue to step 4 with bcp_{i+1};
15:     **end if**
16: **end for**
17: *//Phase 3: Assess the global objective of the plan.*
18: calculate average turn around time TR for *BCP*;
19: calculate average compactness value GT of terminals ;
20: *//Phase 4: If necessary, update* α *and initiate a new game stage* $m = m+1$*, otherwise, plan complete for BCP*
21: **if** *conditions* 5.31 *and* 5.32 *holds* **then**
22:     Plan complete;
23: **else**
24:     $\alpha_j^{\star} = \alpha_j * (1-\theta), m = m+1$, go to step 4 //Update berth occupancy level and game iterations
25: **end if**

- Phase 4: Every iteration m (lines 5-19, Algorithm 7) is one game stage wherein all barges are seeking their best visit sequence with minimum turnaround time. Their decisions (rotation plan) will influence the terminal resource utilisation in Phase three (lines 17-20) and calculate the objectives of both barge and terminal operators. In phase four, these two objectives are evaluated based on Equations 5.31 and 5.32. The variable ε_L and ε_F are used to control the number of game stages and the degree of optimisation. If the condition at line 21 is met, it indicates that performance improvement cannot be gained by further gained. As a result, an equilibrium is reached. Alternatively, a new game stage is initiated, and the corresponding strategy of the leader is modified.

5.4.2 Collaboration mechanism 2

To further explore the possible collaborations,another level of collaboration between terminal and terminal is also added to the model. This is termed collaboration mechanism 2. As discussed in collaboration mechanism 1, the terminals which reside at the upper level, turn by turn, make decisions to achieve better resource utilisation. This is obtained by selecting a suitable berth occupancy level. The overall compactness value not only relies on the berth occupancy level of a single terminal, but it is also influenced by other terminals choices. Therefore, selecting a suitable combination of berth occupancy level is a critical issue. Collaboration mechanism 2 (shown in Figure5.12, annotated by the red line) aims to identify the berth occupancy level for each terminal so that the overall compactness value is minimised. A small compactness value will yield a better terminal resource utilisation rate and allow terminals to serve more vessels within a certain period. In essence, each individual makes its own decisions by alternating its berth occupancy level independently. Upon a successful negotiation with the barges, the decision variable will be adjusted based on the evaluation.

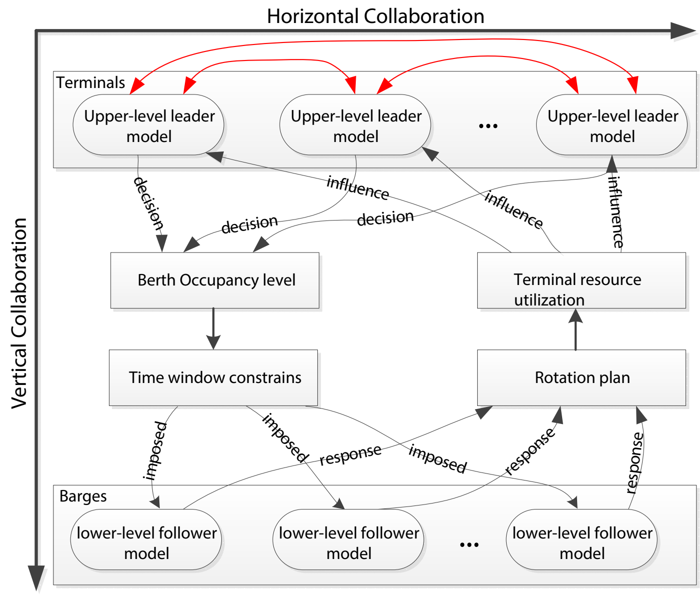

Figure 5.12: The framework of collaboration mechanism 2

Barge agents and terminal agents

In the collaboration mechanism 2, the design of the BA and TA are identical with the those discussed in collaboration mechanism 1 (see the section 5.4.1).

Mediator agent (MA)

The major difference between collaboration mechanisms 1 and 2 is the coordination strategy incorporated by the coordination agent. As shown in Figure 5.13, the introduction of an MA leads to a double-negotiation loop structure.

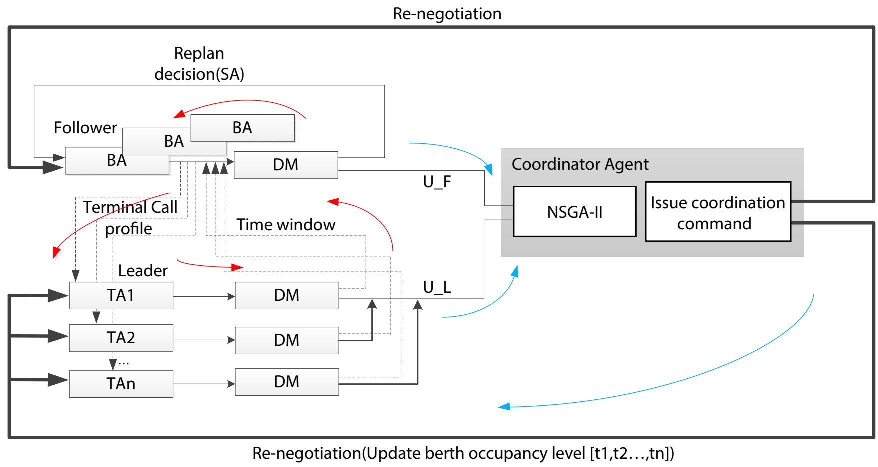

Figure 5.13: A coordinator agent control loop framework

The outer loop is controlled by an MA by executing the NSGA-II algorithm 4. At each iteration, the MA first generates a population of individuals where each individual contains berth occupancy level for each terminal. This will be distributed to each TA and be converted to available time windows by the TA. Following this, negotiations take place among BAs and TAs in order to obtain a feasible plan. This is done according to the leader-follower structure (see Figure 5.12) described in the previous section. The output of the inner loop is a feasible plan for each barge.

The MA will receive the barge turn-around time and terminal compactness value (U_f, U_l). The generations will be evolved under the principle of the NSGA-II framework. Such iterations will continue until the exit condition is met.The MA plays the role of enabling the cooperation process. By participating in the implementation of NSGA-II, it can establish the evolutionary environment and the authority to control the BA-TA negotiations towards a win-win result.

The pseudo code is presented in Algorithm 4. The NSGA-II operations include:

- Initial population: In order to manipulate the genetic operations, the decision variables vector has to be encoded as a chromosome. In our case, a gene represents the berth occupancy level of one terminal (see Figure 5.14).

- Fitness calculation: As a population of chromosomes is created, it will be interpreted as the available time window. The information will be assessed by the followers. Each follower will set their sailing plan accordingly by incorporating a simulated annealing algorithm. As the sailing plan is created, the solution vector is available. This is represented in Figure 5.15. Given the decision vector, the fitness value is calculated.

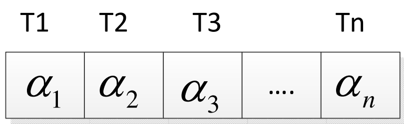

Figure 5.14: The individual chromosome representation in the genetic algorithm: α is a float number ranging from 0 to 1

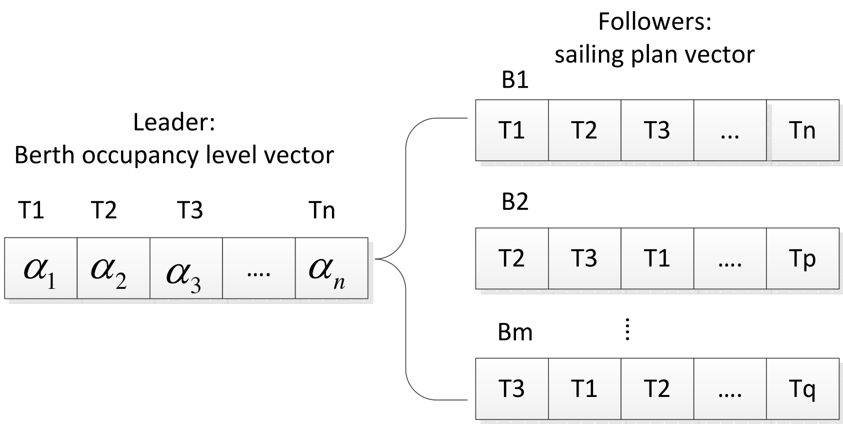

Figure 5.15: A decision vector representation in the genetic algorithm

- Ranking and selection: To tackle the bi-level leader-follower negotiation and find non-dominant solutions, the Pareto dominance criterion is used. When all decision vectors are available, they will be ranked. The ranking procedure is iterative where non-dominant solutions of the current generations ranked, and the highest rank has been removed from the population. Then the non-dominant solutions from the remaining part will be ranked as the second highest rank. This process continues until every solution within the population has been considered.

- Evolving: The evolving process includes two operations: crossover and mutation. They are used to reproduce new generations in order to obtain a better fitness value. The tournament selection method is used to determine which individuals are selected to participate in the reproduction phases. The crossover and mutation operations allow the exchange of existing genes so that new genetic material can be generated. During the reproduction process, elitism is also applied to guarantee a better propagation of the chromosome of the non-dominant solutions.

Integrated decision-making

The overall structure of the hybrid algorithm is shown in Figure 5.16. In the upper level, a genetic algorithm, specifically the NSGA-II, is applied to supervise the horizontal collaboration process. It aims to search for a suitable berth occupancy level that enhances its compactness value. At the same time, the simulated annealing (SA) algorithm is adopted at the lower level to find the best rotation route for followers. This facilitates vertical collaboration because it enables followers to find the best route by considering the leaders' decision.

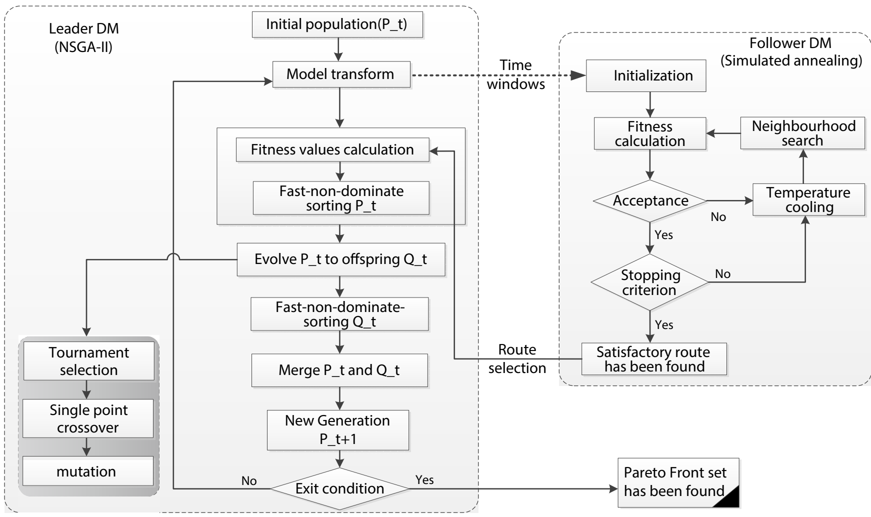

Figure 5.16: NSGAII-SA decision-making structure

5.5 Evaluation

In this section, an evaluation of the proposed collaboration mechanisms will be given. It aims to demonstrate the performance of the algorithm against the current practice. First, a scenario design is presented in section 5.5.1. Second, two mechanisms will be evaluated in sections 5.5.2 and 5.5.3 respectively.

5.5.1 Scenario design

In order to evaluate the performance of the system, test scenarios need to be defined. within the literature, works related to hinterland barge planning problems all include self-defined test scenarios (A. M. Douma, 2008; S. Li, 2016). Currently, there is no available test instance to provide a benchmark for this study. Moreover, for privacy reasons, the involved parties are reluctant to provide information that can be used as a test scenario. In light of this, the test scenario used in this thesis is defined below.

Port settings

Port settings include the parameters which are related to the port layout and the terminal service profiles.

- Port layout: A port layout is defined as the physical position of terminals in a port region. In this case study, the port layout is given as a linear structure which mimics the shape of the Port of Rotterdam. A layout with eight terminals is considered, and its geographical position is specified.
- Terminal service profile: The terminal service profile defines the parameters of terminal operations. Each terminal has one quay for berthing vessels. The operation rate for each terminal is fixed. This indicates that the time for handling one container is fixed.

Barge settings

Barge settings are the parameters of barge rotation planning. These include arrival time into the port region, the latest departure time from the port region, the set of terminals to be visited, and the number of containers to be loaded and unloaded at each terminal. Besides, the sailing time of each barge is assumed to be identical for all considered barges because the distance between two terminals means the sailing time is known.

Test scenario

Given the port and barge settings, the following scenarios are considered:

- Scenario 1 (S1): [2,4] barges, each visits [3,5] terminals;
- Scenario 2 (S2): [5,7] barges, each visits [4, 8] terminals;
- Scenario 2 (S3): [8,12] barges, each visits [6,10] terminals;

For each scenario, the number of containers to be loaded and unloaded at each terminal varies between 5 and 20.

For the terminal settings, events are randomly inserted into the terminal plan board. Such events can represent sea-vessel calls, terminal close, scheduled maintenance, and so on.

To evaluate the performance of each collaboration mechanism, four different types of solutions are compared. These are listed as follows:

- Mechanism-1(M1): In M1, the primary goal is to find a solution which results in a minimum turnaround time. It is assumed that the terminal operator will be flexible to help satisfy barge calls without considering its resource utilisation rate. The solution generated by M1 will be preferred by a barge operator.
- Mechanism-2 (M2): In M2, the terminal operator has more power in determining the time slot. The solution generated by M2 is similar with current practice (see discussion in section 2.3.2). As such, the M2 solution is preferred by the terminal operator.
- Collaborative mechanism-1 (CM1): a CM1 is generated by applying collaboration mechanism 1.
- Collaborative mechanism-2 (CM2): a CM2 is generated by applying collaboration mechanism 2.

Simulation environment

Figure 5.17 depicts a framework of the simulation tool. It includes three main parts: an agent system, a system logger, and a seeds generator. All building blocks are coded using Java. The agent system is built on a JADE framework (see the section 3.2.3 for an introduction). The system logger and the data seeds generator are also implemented with Java. The seeds generator is used to generate test cases, and the system logger is used to store the generated result.

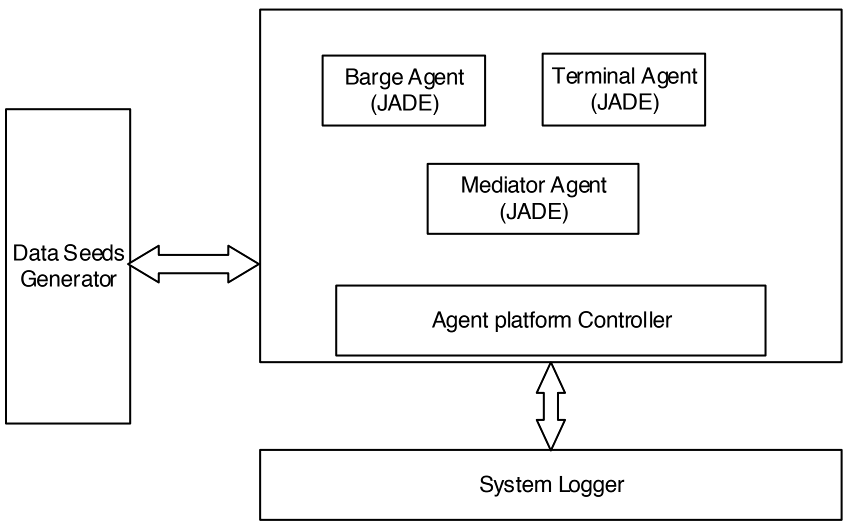

Figure 5.17: The overall simulator framework

5.5.2 Evaluation- collaboration mechanism 1

In this section, the performance of collaboration mechanism 1 is evaluated by the following three steps. First, a case study is presented. Second, a parameter tuning process is performed. Third, an extensive performance comparison between two different hybrid algorithms is given.

A generated rotation plan

By applying CM1, the visiting sequence, arrival time, and departure time of each barge i at each visited terminal p is determined. Figure 5.18 presents the rotation plan generated by applying collaboration mechanism 1 in a scenario where six barges visit eight terminals. Figure 5.19 presents the objective value of both terminal and barge operators in each iteration. At the initial stage, the turnaround time of barges is high, and the terminal compactness value is low. This indicates that the terminal operator is placing the incoming barges into a queue to achieve a maximum resource utilisation rate. Thus, the barge experiences a long waiting time. As the iteration continues, the collaboration mechanism takes action to develop a compromise between these two objectives. Consequently, the turnaround time is reduced and the compactness value is increased. This means the terminal operator offers more flexibility to barge operators.

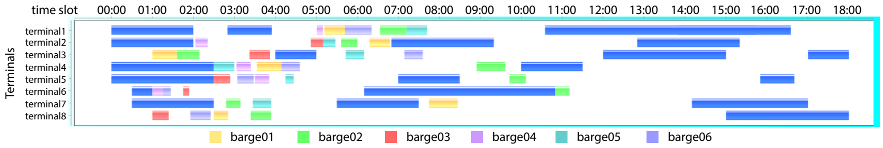

Figure 5.18: An example of the generated plan with a Gantt chart representation. The blue bar represents the planned events and the other colored bar represents the plan for each barge call.

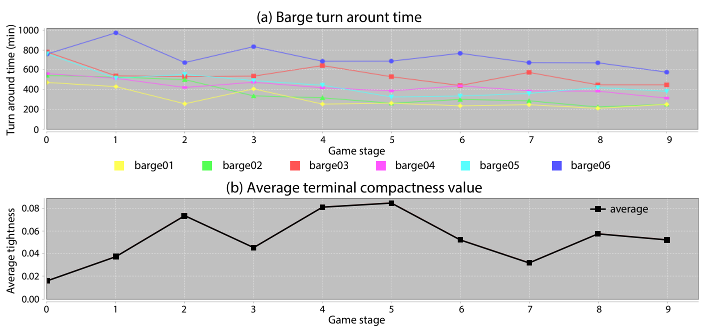

Figure 5.19: Global objective:the upper figure is the summary of barge turn around time, and the lower Figure is the average terminal compactness value

Parameters tuning

Parameter tuning is used to tune the algorithm parameters so that the accuracy of algorithm performance can be maximised. As discussed in section 5.2.2, two parameters are considered in an SA. These include the initial temperature and the cooling rate. To find the best values of the initial temperature and the cooling rate, a fractional factorial experiment (FFE) method is used. An FFE is regarded as a tool that can be applied to determine the main effect of factors and some of their interactions with a limited number of experiments (Naderi et al., 2009). Among various FFE approaches, the Taguchi method is proven to be effective in determining the appropriate parameters for meta-heuristics (Sels & Vanhoucke, 2012; Sarrafha et al., 2014).

In the Taguchi method, two tools are used which are an orthogonal array and the signal-to-noise-ratio (Sels & Vanhoucke, 2012). The orthogonal array is a fractional factorial matrix where each row represents the level of the chosen parameters, and each column represents a specific factor that can be tuned. According to the orthogonal array, different experiments are performed, and the obtained results are termed response value and transformed into the signal-to-noise ration (S/N). The signal refers to the desired value. The noise denotes the undesirable value. The objective of the Taguchi method is to maximise the signal-to-noise ratio. In general, in order to minimise the problem, the corresponding S/N ratio is represented in Equation 5.33 (Naderi et al., 2009), where O is the objective value obtained in each run.

$$S/N = -10 * \log O^2 \tag{5.33}$$

The parameters to be tuned in SA is summarised in Table 5.8. The levels considered are summarised in Table 5.8. The response value is calculated by considering two factors: the computation time and the objective value. The computation time is the time elapsed to obtain a solution. The objective value is the barge turn around time. First, both values are normalised. The calculation of the response value can be found in Equation 5.34 and the corresponding signal to noise ration is calculated in Equation 5.35.

$$r = \sum (N_t + N_{obj}) \tag{5.34}$$

$$S/N = -10 * \log r^2 \tag{5.35}$$

Table 5.8: SA parameters level

Parameters range	Range	Low	Medium	High
Initial temperature	500-1500	500	1000	1500
Cooling rate	0.01-0.05	0.01	0.03	0.05

The numerical results are summarised in Table 5.9. Figure 5.20 depicts the signal to noise ratio for each combination of parameters. As shown in the figure, it can be concluded that the best combination of parameters is given at the medium level. They are 1000 for initial temperature and 0.03 for the cooling rate.

Table 5.9: A summary of experiment results for parameters tuning for CM1

Run	T	C	N_t	N_obj	R	S/N
1	1	1	0.66	1	0.7178	1.439965
2	1	2	0.7	0.976	0.721288	1.418913
3	1	3	0.71	0.931	0.685431	1.640366
4	2	1	0.705	0.887	0.641897	1.925347
5	2	2	0.71	0.851	0.614151	2.117252
6	2	3	0.73	0.84	0.61925	2.08134
7	3	1	0.951	0.832	0.798313	0.978271
8	3	2	0.987	0.811	0.815945	0.883391
9	3	3	1	0.903	0.907705	0.420555

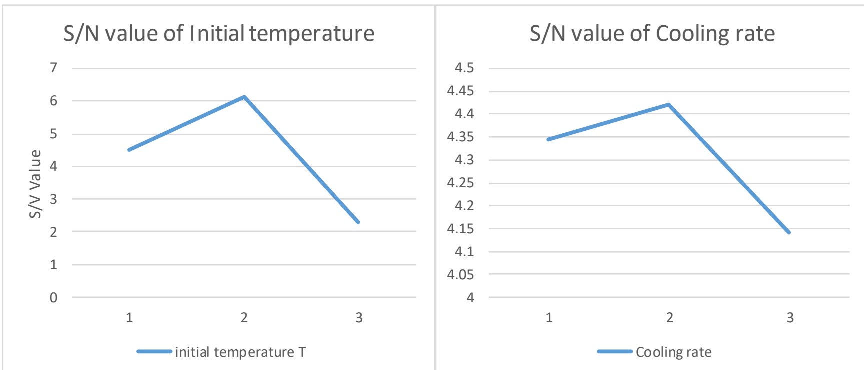

Figure 5.20: The parameters tuning result of SA

An extensive comparative study

To further evaluate the performance of CM1, an extensive comparative study is performed in this section. The three strategies will be applied to three different scenarios with a different number of barges and terminals. Each scenario is running five times, and an average value is calculated as the final result. The result is normalised by dividing the objective value by the largest number among three strategies, followed by Equation 5.36.

$$N_{O_{M1}} = \frac{O_{M1}}{\max\{O_{m1}, O_{m2}, O_{CM1}\}} \tag{5.36}$$

A comparison of normalised results for each scenario are summarised in Figures 5.21, 5.22, and 5.23 respectively. For KPI-1, the solution generated by M1 yields the best quality. However, the terminal resource utilisation rate is low (a higher compactness value indicates lower terminal utilisation level). This is because, in M1, the barge operator has full control over its rotation planning process. The barge operator attempts to find an available time in a terminal plan board which leaves a large amount of idle time between consecutive events. For KPI-2, the M2 outperforms the others. This is because the terminal operator has full control of the time window selection. Thus, its objective can be fulfilled. The goal of applying a collaboration mechanism is to harmonise the interests of both parties and reach a mutually beneficial solution. CM1 is generated by applying collaboration mechanism 1. It attempts to achieve a mutually beneficial solution for both terminal and barge operators. Comparing CM1 with M2, the turnaround time was improved by 15 %, 30%, and 35% in each scenario. In addition, the compactness value obtained by CM1 outperforms those generated by M1 with a 22%, 35%, and 38% improvement in each scenario. As a demonstration, the generated rotation plan for each strategy in scenario 2 (six barges visiting eight terminals) is presented in Figure 5.24.

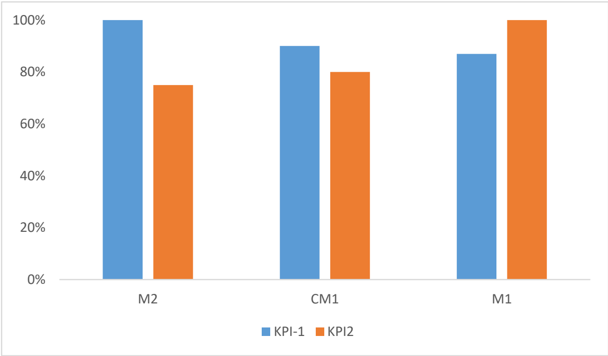

Figure 5.21: S1: Comparison result for CM1

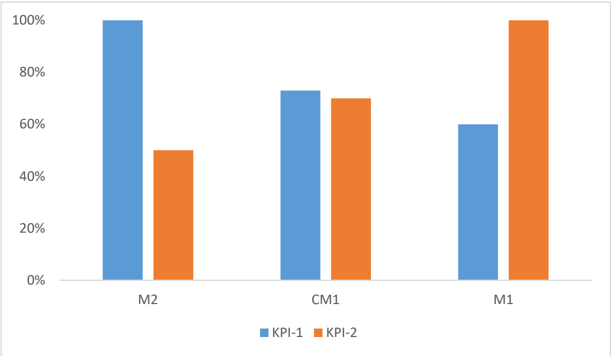

Figure 5.22: S2: Comparison result for CM1

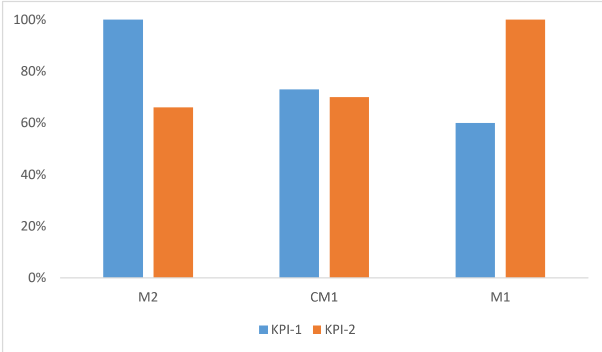

Figure 5.23: S3: Comparison result for CM1

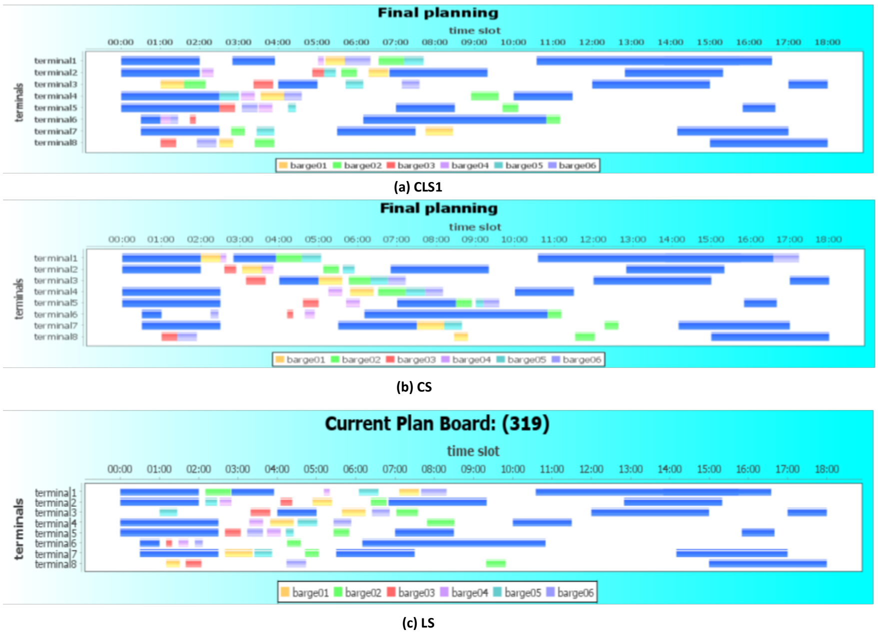

Figure 5.24: Result comparison for CM1

5.5.3 Evaluation- collaboration mechanism 2

In this section, the performance of the proposed algorithm is evaluated by following three steps. First, the system is applied to the benchmark system. Second, a parameter tuning process is performed. Third, a comparative study is conducted to compare the results with those obtained using different strategies.

A generated rotation plan

Figure 5.25 presents the rotation plan generated by applying CM2 in a scenario where seven barges visit eight terminals. As discussed in section 5.4.2, an NSGA-II algorithm is applied to find a Pareto front. In each iteration, a number of solutions are generated with different objective values. Figure 5.26 depicts some chosen Pareto frontiers (front in the generation 10, 20, and 30). As a result, it can be easily recognised that the obtained Pareto front has improved at each generation and the individual in the final frontier is evenly spread which covers the extreme point for single objectives.

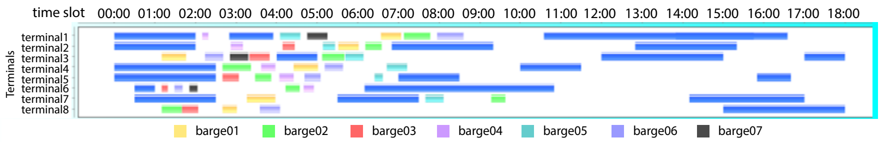

Figure 5.25: An example of generated plan by applying collaboration mechanism 2

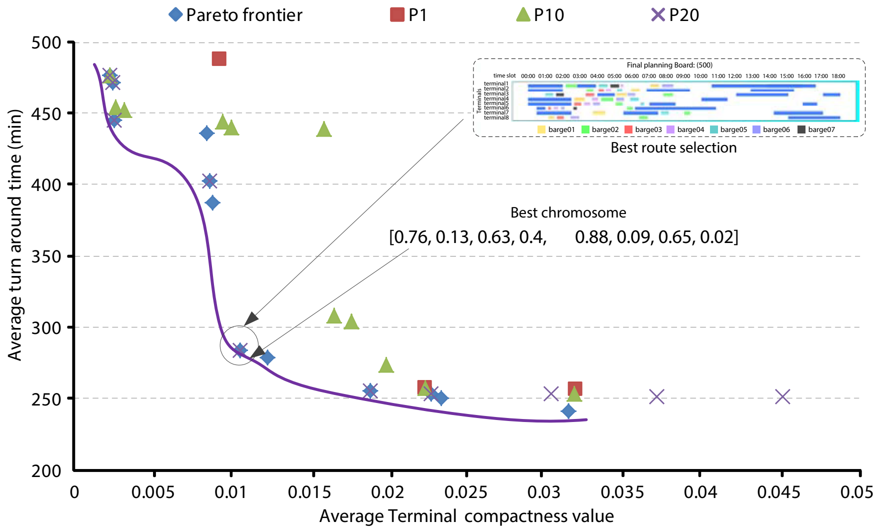

Figure 5.26: The NSGA(II)-SA Pareto front (for 8 terminals and 7 barges)

The use of NSGA-II will generate a set of Pareto front solutions. Table 5.10 below presents the objectives value of each individual in the Pareto frontier. Following this, the task of the decision makers is to determine the best solution (Zio & Bazzo, 2012). In our HBT problem, choosing the best solution from the Pareto front is a non-trivial task because there is no party that could be in charge of both terminals and barges. Put simply, both terminal and barge parties run their own business. The port authority or the government could play a role, but this would be a long term policy (Feng, Pang, &

Table 5.10: A summary of experiment results for NSGAII-SA

$O_{terminals}$	O_{barges}	$NO_{terminals}$	NO_{barges}	W_{value}
0.0023196	476.25	0	1	0.5
0.0024355	471.75	0.0039841	0.9808408	0.4924
0.0024606	471.625	0.0048449	0.9803086	0.4925
0.0025578	445.5	0.0081884	0.8690792	0.4386
0.0084945	436.375	0.2122586	0.8302288	0.5212
0.0086221	402.125	0.2166438	0.6844065	0.4505
0.0088391	387.375	0.2241055	0.6216072	0.4228
0.0105602	283.875	0.2832675	0.1809473	0.2321
0.0123068	279	0.3433063	0.1601915	0.2517
0.0188437	255.625	0.5680101	0.0606705	0.3143
0.0227303	254	0.7016101	0.0537519	0.3776
0.0232986	250.625	0.7211438	0.0393826	0.3802
0.0314109	241.375	1	0	0.5

Lodewijks, 2015b). Consequently, this study applied a weight-based approach (Harris et al., 2014) to determine the best solution where the objectives of both barges and terminals have equal importance. First, the solutions that appear in the final Pareto front are normalised and the weighted value is obtained by adding the two objectives and multiplying them by a factor of 0.5. The calculation is given in Equation 5.37:

$$\begin{aligned} N_{f_i}^m &= \frac{{f_i}^m - f_{\min}}{f_{\max} - f_{\min}} \\ W_m &= \sum_{j=1}^{2} 0.5 * N_{f_j} \end{aligned} \tag{5.37}$$

Where f_i is the i_{th} objective value of individual m, f_{min} is the lowest objective among the Pareto front whereas f_{max} is the largest. N_{f_i} is the normalised value for m and W_m is the weighted value for m.

The result is given in Table 5.10 and the best solution is highlighted by red. Finally, the option with the lowest weighted value is chosen as the best solution.

Parameters tuning

In this section, the parameters that are used in the NSGA-II algorithm will be tuned followed by the Taguchi method (introduced in section 5.5.2). The parameters associated with NSGA-II will be tuned which are the population number, the generation number, the probability of crossover and the probability of mutation (see Table 5.11). Their best combination is obtained by applying FFE to different combinations. In order to obtain the S/N ratio, the response value for each test is calculated by considering three factors, namely the number of individuals in the Pareto frontier and the two objectives from the best solution obtained in each run. As shown in Table 5.12, before obtaining the response value, the results of each run are normalised first. Then the response

Table 5.11: The parameters levels of NSGA-II algorithm

Algorithm Parameters	Parameters Range	Low (1)	Medium (2)	High (3)
N_p	30-50	30	40	50
P_c	0.5-0.7	0.5	0.6	0.7
P_m	0.01-0.03	0.01	0.02	0.03
N_g	30-50	30	40	50

value is calculated by Equation 5.38 which considers the convergence and diversity of the solution simultaneously. Finally, the S/N is given in Equation 5.39,

$$r = (\sum_{i=1}^{k} w_i * N_{o_i}) + N_{pf} \tag{5.38}$$

$$S/N = -10 * \log r^2 \tag{5.39}$$

where w_i is the weight for each objective i, N_{o_i} is the normalised value of o_i and N_{pf} is the normalised value of the number of individuals in Pareto front. r is the response value of each run.

The results are summarised as in Table 5.12. For each combination, this study performed 5 runs, and the mean value of the objective values are taken as the response value. The number of the Pareto front is the largest number obtained during the five runs. Figure 5.27 presents the mean S/N for each , and the highest S/N indicates the optimal choice. The best combination is given as: N_p is 50, N_g is 50, P_c is 0.6 and P_m is 0.01.

Table 5.12: A summary of experiments results for parameters tuning for CM2

Run order	NSGA-II Parameters				Results			normalised value			Response value	S/N
	N_p	N_g	P_c	P_m	pf	Obj_1	Obj_2	N_{pf}	N_{obj_1}	N_{obj_2}	R	S/N
1	1	1	1	1	13	0.0105	283.9	0.542	0.7	0.976	1.38	2.789
2	1	2	2	2	12	0.0149	281.75	0.5	0.993	0.969	1.481	3.411
3	1	3	3	3	15	0.0147	250.875	0.625	0.98	0.863	1.564	3.784
4	2	1	2	3	14	0.0145	290.75	0.583	0.967	1.0	1.567	3.901
5	2	2	3	1	20	0.0104	273.015	0.833	0.693	0.939	1.649	4.344
6	2	3	1	2	18	0.0124	262.86	0.75	0.827	0.904	1.615	4.163
7	3	1	3	2	16	0.0133	265.13	0.667	0.887	0.912	1.567	3.901
8	3	2	1	3	15	0.0153	271.733	0.625	1.0	0.934	1.592	4.039
9	3	3	2	1	24	0.0142	260.87	1.0	0.933	0.897	1.915	5.643

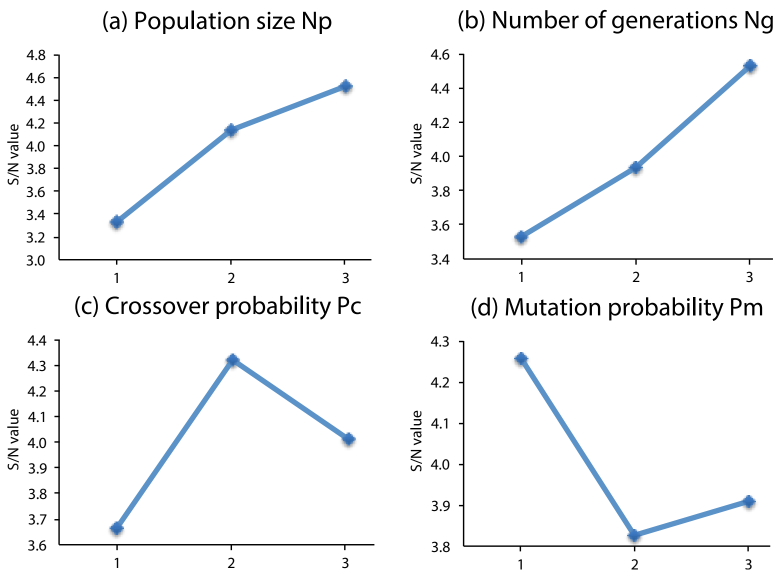

Figure 5.27: The mean S/N plot for different levels of NSGA-II parameters

Comparative study

In this section, the solution generated by CM2 is compared with CM1 and M2 for three designed scenarios. The results of the comparative study can be found in Figures 5.28, 5.29, and 5.30.

The findings drawn are as follows. For KPI-1, both CM1 and CM2 outperform M2. This indicates that the average turnaround time of barges was reduced by applying the collaboration mechanism. Furthermore, CM2 yields a better result compared with CM1, meaning the use of NSGA-II outperforms CM1.

For KPI-2, the M2 provides the best result because the terminal operator has full control. This is because M2 only considers the terminals objective while CM1 and CM2 consider both terminal and barge interests simultaneously. This study considered the results of M2 as a starting point for the use of CM1 and CM2. By applying a collaboration mechanism, the interests of both terminal and barge operators can be compromised on. As a result, a shorter barge turnaround time is gained by a loss of the compactness value. However, the compromise is relatively small and resulted in 4%, 5%, and 4% loss when comparing M2 and CM2.

According to the simulation result, CM2 outperforms CM1. For KPI 1, the improvements were 5%, 8%, and 5% for each scenario. For KPI 2, the improvements were 4%, 3%, and 3% for each scenario.

To demonstrate, the generated rotation plan for each strategy in scenario 2 (six barges and eight terminals) is presented in Figures 5.31, 5.32 and 5.33.

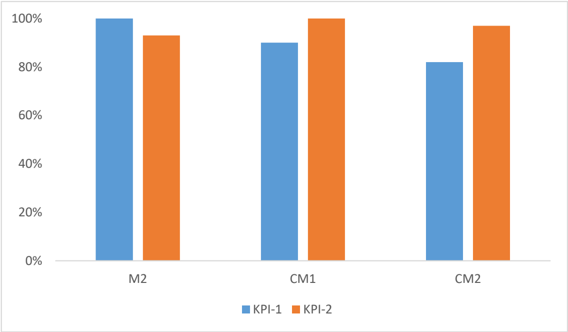

Figure 5.28: S1: Comparison result for CM2

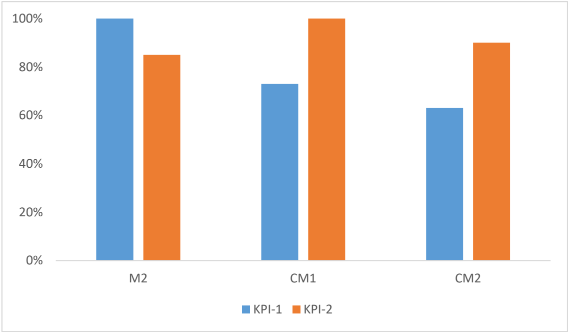

Figure 5.29: S2: Comparison result for CM2

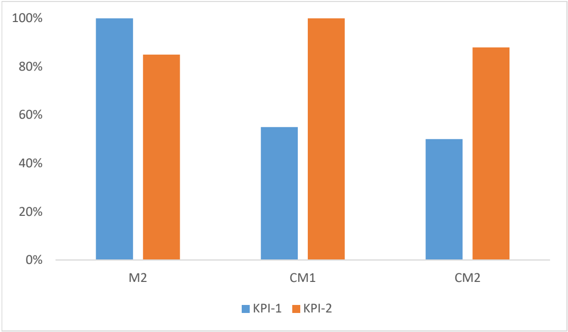

Figure 5.30: S3: Comparison result for CM2

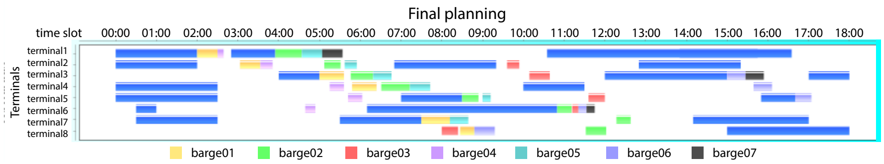

Figure 5.31: Solution generated by CP

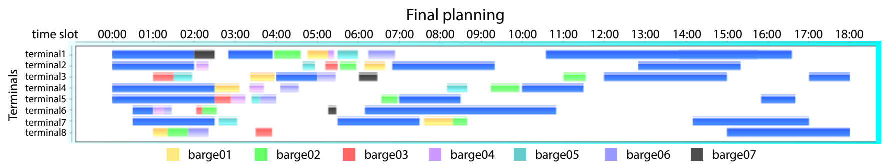

Figure 5.32: Solution generated by CM1

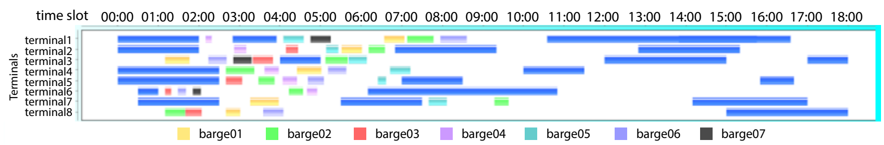

Figure 5.33: Solution generated by CM2

5.6 Towards implementation

In section 4.1, it was mentioned that the use of an integrated DM system in a PL system should support three major functions: handling of users and external requirements, control of information flow, and an information sharing service. In previous sections, efforts were taken to design DM algorithms that support the control of information flow. It was also found that there is a gap in the implementation of the designed algorithms in practical IT systems. In this section, issues associated with implementation will be discussed. It will explain how agents communicate and collaborate and how user requests are handled by the DM system.

5.6.1 Agent communication

Agent communication concerns how agents exchange messages. To handle message exchanges, two aspects should be considered:

- The sender agent needs to convert its internal content to a corresponding message that can be shared with other agents. The receiver agent needs to perform the opposite operation to decipher useful information from the message.

- Once the message has been received by an agent, the agent should be able to check the message origin, unpack the message, and respond to the message by activating specific actions.

As discussed in section 3.1.2, the JADE framework is used as the key technology for implementing the proposed agent system. In JADE, communication between agents is supported by an ontology. A general framework of the JADE agent message exchange is shown in Figure 5.34. JADE provides mechanisms to perform the convention and check operations automatically. The user needs to define the general rules for coding and decoding the messages. In JADE, such rules are defined using ontology (F. L. Bellifemine et al., 2007).

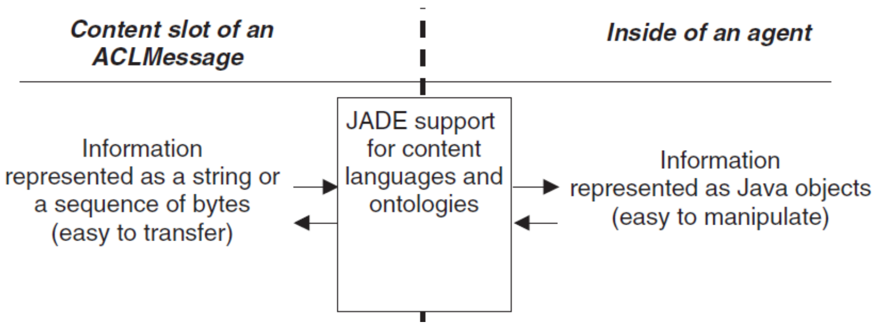

Figure 5.34: JADE agent message handling framework, adapted from F. L. Bellifemine et al. (2007)

Two categories of ontology are developed to organise the agent negotiation, namely, personal ontology and mutual ontology. Personal ontology abstracts the logic and strategic knowledge of agent. It is developed to organise the agents local computing logic and regulate agent intelligence. Mutual ontology ensures understanding between multiple agents. It is designed for information sharing and re-use. The establishment of mutual ontology guarantees no misunderstandings between different agents and therefore enhances the interoperability of the negotiation process. Besides, the use of shared mutual ontology is vital for effective agent communication.

Table 5.13 lists several major negotiation processes. Senders and receivers are represented by assigning agents. Performative indicates the intention the sender wants to achieve by sending the message. Item is the content included in the message. The ontology is structured as a schema. The definition of each schema and typical use is presented as follows:

- Concept schema: an entity with a complex structure that describes the content elements and items. As shown in Table 5.13, the item NewCallSchedule is a concept schema proposed by the TA to activate a new round of BA-TA negotiation.

- Predicate schema: this reflects the status of certain events or scenarios. As shown in Table 5.13, the item CheckSchedule is a predicate schema that is sent from the BA to the MA. It requires the MA to evaluate the value of the current schedule with a status of OK or Not OK.
- Agent action schema: this concept defines specific actions that can be performed by agents. As shown in Table 5.13, the itemUpdateTerminalSchedule is an agent action sent from the MA to the TA. It requires the TA to act to update the terminal schedule.

Table 5.13: The interoperability of mutual ontology (A:action; C:concept; P:predicates)

Negotiation Agents		***Ontology***			
Sender	**Receiver**	**Per formative**	**Schema**	**Items**	**Purpose**
BA	TA	CFP	A	NegotiationAction	Search time slot in terminal schedule list
TA	BA	Proposal	P	Time-Infor	Return available time slot
BA	MA	Inform	P	Check-Result	Return the schedule evaluation Result
BA	MA	Request	A	Re-Schedule	Request MA to create reschedule plan
MA	BA	Propose	C	New-Call-Schedule	Return to BA the new terminal negotiation sequence
MA	TA	Inform	A	Update-Terminal-Schedule	Negotiation end and inform TA to update terminal schedule list

5.6.2 Decision-making process

The whole DM process involves both horizontal and vertical communication. Horizontal communication is conducted between agents in the agent model layer. Vertical communication is conducted between agents and the mediator agent in the agent control layer. To better illustrate the DM process, a sequence diagram is shown in Figure 5.35. The DM process can be separated into three phases:

- Call dispatching: Normally, a barge call contains several elements. This includes terminal visit sequences, the number of containers loaded and unloaded at each terminal, and estimated arrival time for entering the port. Suppose we have a call with three terminals and the sequence is (TA1, TA2, TA3). As shown in Figure 5.35, the barge agent first decodes the call profile and forwards the corresponding messages to dedicated terminal agents. The forward sequence obeys the sequence provided by the barge agent.
- Terminal agent processing: The terminal agent keeps listening to the message sent by the barge agent. Once they have received the request, this will activate

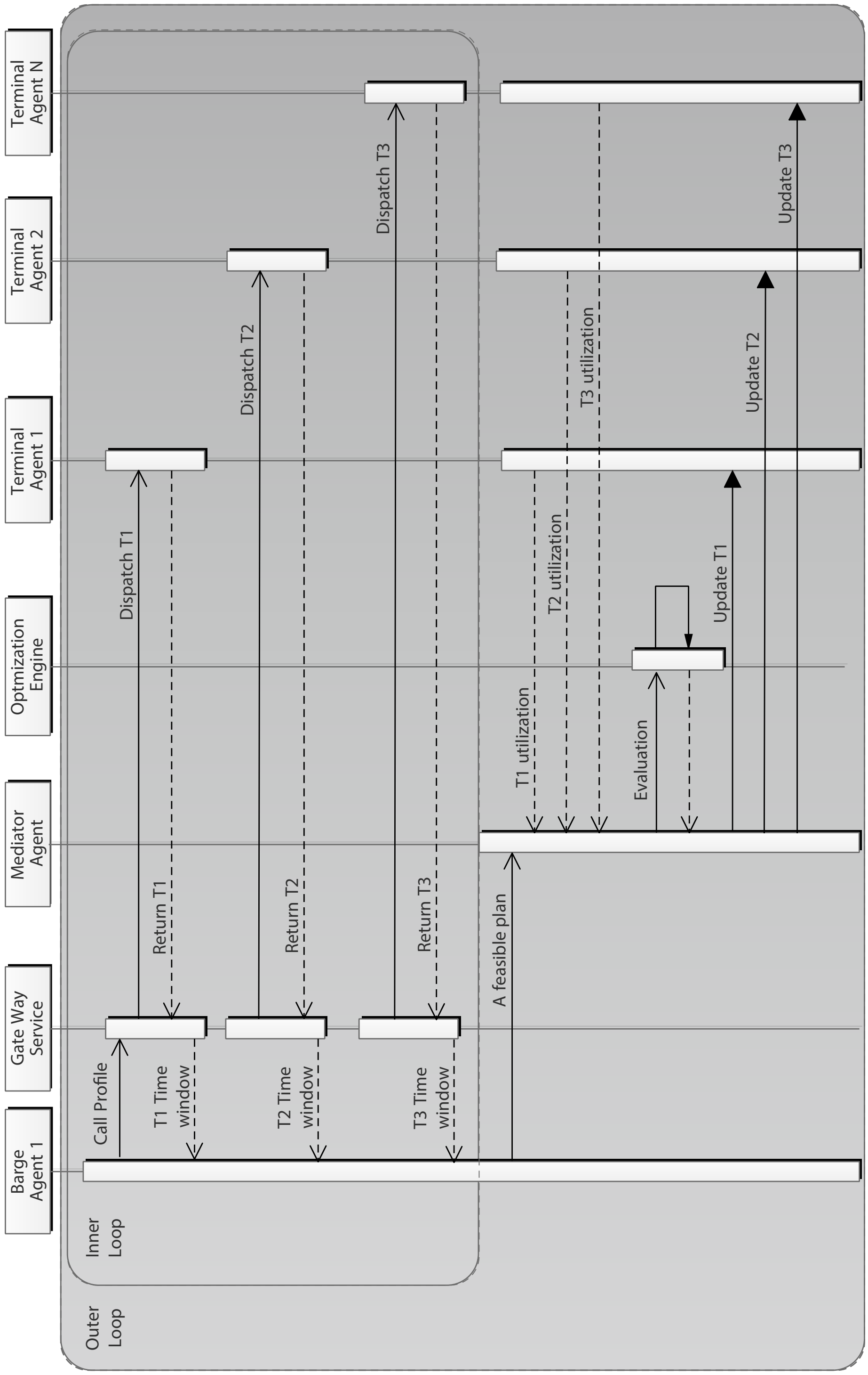

Figure 5.35: decision-making process: a sequence diagram representation

the internal function and connect with the local schedule database. It will search the terminal plan board and return a feasible time slot for the barge calls.

- Evaluation: A feasible plan is generated through negotiation between terminal and barge agents. This will be buffered inside the mediator agent. To verify the performance of the current plan, the mediator consults with the optimisation engine. The engine will evaluate the quality of the plan concerning rotation time and terminal resource utilisation. Decisions such as cancelling terminal calls or re-planning terminal calls will be issued.

5.6.3 User-system interaction

To improve the applicability and usability of the integrated DM system, the integrated DM system is further combined with service-oriented architecture. First, though the use of agents is attractive in the context of intensive negotiation among different parties, it appears that its use has not gone further than university laboratories. Therefore its practical application is limited (Van Rijswijck & Davydenko, 2007). On the other hand, SOA is considered a promising architecture for many practical applications which are characterised as interoperable, heterogeneous and dynamic. The concept of service is highly modularly and can be implemented using different programming methods. Consequently, exposing agent functionalities and interactions as a service under the umbrella of SOA could have potential value in making agent system applicable in industry (Tapia et al., 2009; Dong & Srinivasan, 2013). Second, mutual benefits can be gained by combining MAS with SOA. For instance, the SOA-based system does not necessarily concern the properties of autonomy, social ability, reactivity and proactiveness. These are essential characteristics of the multi-agent system. Combining these two could bring added values to industry (Marik & McFarlane, 2005).

To combine the SOA with the integrated DM system, a gateway agent and a blackboard object are introduced. A gateway agent is used to interact with web services on one side (service consumer) and for web service requests to be served by an agent (service provider) on the other side. A blackboard data object holds all the status-related information. It is directly connected with the gateway agent. Status-related information is initiated and updated by the MAS planning system. The web services initialise the blackboard object which synchronises the data between MAS planner and the blackboard. A demonstration of the implemented system can be found in Appendix B.

A typical sequence is explained as follows:

- The barge operator requires the system to issue a plan. The terminal call profile is given via a browser, which generated as a POST message.
- The servlet (a piece of software program) handles the message by invoking the sent message actions.

- Following this, a new Blackboard object is created. The proper content is extracted from the message and filled into the blackboard object.

```
BlackBoard blackboard = new BlackBoard();
blackboard.setContent();
blackBoard.setReceover();
try{
jadeGateway.execute(blackboard);
}catch(Exception e ){e.printStackTrace();}
```

- The gateway agent receives the blackboard object and extracts all the information from it. The following actions involves data persistence and message forwarding.

```
public void processCommand(java.lang.Object obj){
if(obj instanceofBlackBoard){
blackboard = (BlackBoard) obj;
ACLMessage msg = new ACLMessage(ACLMessgae.CFP);
msg.addReceiver();
msg.setContent();
send(msg);}}
```

- Whenever a plan is issued by the MAS, the results will be aggregated and delivered to the gateway agent. The gateway agent initialises the blackboard object. Then, the plan result is finally sent back to the servlet.

```
AddBehaviour(new CyclicBehaviour(this)){
public void action(){
ACLMessage = receive();
if(msg!=null && blackboard!=null){
blackboard.setMessage(msg.getcontent());
releaseCommand(blackboard);
}else{
block();}}}
```

5.7 Discussion

This chapter aimed to present the design of the proposed integrated DM system, with a goal to improve the coordination and planning of hinterland barge transport. The contributions of this case study are summarised as follows:

- Improve collaboration between terminal and barge operator: Compared with the previous studies (A. M. Douma et al., 2011; S. Li et al., 2014) which chosen the barge rotation time as the primary goal to optimise, this case study considered the interests of both terminal and barge operators together. The interaction between the terminal and barge operators are first modelled as a leader-follower bi-level model. Based on the interaction model, two collaboration mechanisms are proposed aim at harmonizing the conflict of interests between barge operators and terminal operators. The mechanisms incorporated two meta-heuristics algorithms, namely non-dominant genetic algorithms and simulated annealing algorithms. From the simulation result, it can be concluded that the algorithm optimize both barges turn around time and terminal resource utilisation at the same time.

- Improvement of communication and information sharing: Compared with the previous study, which proposed a distributed multi-agent system consists of two types of agents (the barge and terminal agents) (A. M. Douma et al., 2011), a mediator-based agent system was proposed in this research. By introducing a mediator-based MAS structure, the interactions between the barge and terminal agents are controlled and coordinated by a mediator agent. The interaction of the mediator-based agent system includes two phases. In the first phase, terminal agents allocate their available time slots. A barge agent determines its rotation route by collecting available time windows from terminal agents. In the second phase, a mediator is involved to assess the performance of the rotation plan regarding the average waiting time and the terminal utilization level. During barge and terminal agent interaction, the mediator agent can judge the performance by considering both the interests of barges and terminals. In case there is a room for further improvement, a new interaction is initiated. For example, a new round of negotiation can be initiated by the mediator if the terminal utilization can be further increased by adjusting the route of barges. The multi-agent system proposed in this research allows, in contrast with the distributed agent system, for balancing the interests of the barge and terminal operators through a high level of coordination. On the one hand, it allows barge and terminal operators to optimise their operations with limited information. On the other hand, the mediator agent takes the role of coordinator to adjust the interaction between the barge and terminal agents, thus guarantee better global performance. A mediator agent system was designed and implemented to assist information exchange and sharing. The model of the agent,the intelligence of the agent, and the communication protocol of the agent were defined.

- The issue was solved from an ICT architectural point of view: In this research, an integrated DM framework was designed before directly addressing the DM problems. Guided by this framework, each component was then specified and implemented. The proposed framework works effectively to solve HBT planning problems. Moreover, the designed framework can be extended and adapted to other similar challenges in PL.

5.8 Conclusion

The content of this chapter answers sub-question 2.1 as listed in Chapter 1. It investigated how the integrated DM framework proposed in Chapter 4 can be used to improve the performance of PL further, specifically concerning tactical decision-making problems in a PL system. A case study of hinterland barge planning was performed. More specifically, the proposed integrated DM framework was applied to achieve two goals: (1) to enhance the communication and coordination between barge operators and terminal operators; (2) to improve planning performance for barge rotation time and terminal resource utilisation. Each self-interested party was able to maintain its own objectives. At the same time, a centralised coordination platform assisted the negotiation process in order to promote a global optimum solution. The mechanisms considered the inputs and outputs from both terminal and barge parties in order to incorporate both contexts into the global interests optimisation process.

To enhance collaboration, two different strategies were developed in section 5.4. In collaboration mechanism 1 (CM1), the interactions between terminal and barge operators were modelled as leaders and followers according to the Stackelberg game. To assist decision-making, a hierarchical meta-heuristics was developed to assist the equilibrium searching process. According to the simulation results, the results obtained from CM1 were better than M1 (the mechanism in which the barge operators interests are dominant) and M2 (the mechanism in which the terminal operators interests are dominant). Comparing CM1 with M2, the turnaround time (KPI-1) improved by 15 %, 30%, and 35% for each scenario respectively. In addition, the terminal resource utilisation rate (KPI-2) obtained by CM1 outperformed those generated by M1 with a 22%, 35%, and 38% improvement for each scenario respectively.

In collaboration mechanism 2 (CM2), decision-making was further improved by identifying the berth occupancy level for each terminal in order to improve the overall terminal resource utilization rate by minimising the average compactness value. Compared with collaboration mechanism 1, it extended the space search by allowing terminal operators to alternate its own berth occupancy level. A hierarchical algorithm with NSGA-II and simulated annealing was designed to assist the Pareto front searching process. The solution generated by CM2 was compared with CM1 for three designed scenarios. According to the simulation result, CM2 out performed CM1. For KPI-1, the improvements were 5%, 8%, and 5% for each scenario. For KPI-2, the improvements were 4%, 3%, and 3% for each scenario.

On top of the designed collaboration mechanisms, a mediator-based multi-agent system is designed to assist the information exchange and sharing process where agents are developed, deployed, and linked to the decision-making process.

Chapter 6

A case study of reliability assessments of belt conveyor systems *

This chapter investigates how the ICT technology and framework proposed in Chapters 3 and 4 can be applied to improve the reliability assessments of equipment in PL systems. A case study of reliability assessments of belt conveyor systems is conducted in this chapter. An introduction to reliability assessments of large-scale belt conveyor systems is presented in section 6.1. An integrated DM framework and the applied ICT technology is explained in section 6.2 and 6.3. To demonstrate how the framework works, a case study concerning belt tear condition supervision will be given in section 6.4. Section 6.5 reflects the contribution of this research. Finally, a conclusion will be drawn in section 6.6.

6.1 Introduction

In section 2.5, the challenge of current reliability assessments practice of belt conveyor systems was introduced. The current reliability assessments have several challenges. First, there is no ICT system to support automated reliability assessment, decisions were still made manually which can lead to inconsistent and error-prone results . Second, the available reliability assessment approaches are applied only on component levels. There is currently no integrated system that can combine and integrate different measurements and system conditions. To enable an in-depth assessments of a complex system like the BCS, an integrated DM system is essential. Third, BCS is a complex system which generates large amounts of data. Different categories of data need to be associated with each other in order to support integrated decision-making. No such support is currently available.

To further improve the reliability assessment of a large-scale BCS, Section 2.7 proposed two requirements. First, an information system is required to process and inter-

*The content discussed in this chapter has been published in (Feng et al., 2016, 2018) pret gathered information and deliver it to the right person at the right time. Second, an integration approach is needed to mine and integrate of different data sources generated from the BCS.

Based on the requirements, the ICT technologies were selected in Chapter 3. In section 3.1, the MAS technology was selected to support information sharing between different components. Section 3.3 selected the ontology as the DM approach to achieve context-awareness via information integration and data mining. Further, a hierarchical framework was proposed in section 4.4.2 aims to integrate the selected technology.

Figure 6.1 depicts the integrated DM framework that generalised from the conceptual design in Figure 4.5. The integrated DM system is implemented to support context-aware supervision of BCS. Specifically, the supervision tasks are executed by the multi-agent system. The context-modelling is performed by the ontology. The functionalities of each layer are briefly discussed below.

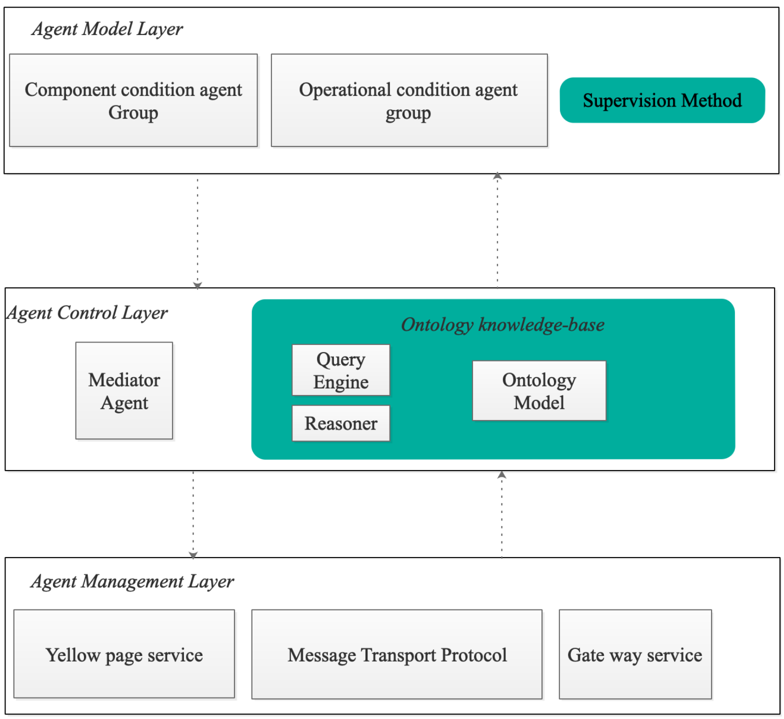

Figure 6.1: The integrated DM framework for the BCS reliability assessments, a generalization of the proposed framework depicted in Figure 4.5.

- In the agent model layer, agents are grouped into different sections based on how they are used and what their capabilities are. Different supervision methods, such as the model-based approaches and knowledge-based approaches are designed into for the purposes of decision-making. A complex reliability assessment task is partitioned into sub-tasks. These can be solved among different agents, and the partial results can be consolidated into a final decision.
- In the agent control layer, a mediator agents is designed. The mediator agent is used to coordinate the actions of the agents in the model layer. The goal is to collect and provide appropriate decisions for system monitoring and supervision. In addition, an ontology model named **ontoSupervision** has been designed and incorporated. It models the relevant information of the system supervision process. The inference engine allows **ontoSupervision** to derive hidden knowledge such as classes and relationships. The query engine provides an interface which enables external systems and functions to interact with the ontology model to perform information manipulations.
- The agent management layer provides the facilities to support agent communication. In this case study, the open source software JADE is used. The introduction of JADE has been given in 3.1.2.

In the following sections, each element will be further discussed. First, the design of the MAS and the ontology are elaborated in section 6.2. Second, the integration of the MAS and the ontology are discussed in section 6.3.2. A protocol named knowledge filtering template is developed to support a new way of integration. To demonstrate the applicability of the developed system, a case study of belt tear condition supervision is performed in section 6.4. Several implementation details are explained, and a prototype system is implemented.

6.2 Implementations of the integrated DM framework

6.2.1 Multi-agent system

The topology of the MAS is shown in Figure 6.2. Three major groups of agents can be identified: the information facilitator agent (IFA), the system supervision agent (SSA), and the user agent (UA). The IFA serves as an information portal for all distributed agents to interact with. It is responsible for delivering relevant information to agents. It is also in charge of coordinating collaborations among agents. The SSA contains two groups, the healthiness condition related agent (HCRA) group and the operational condition related agent (OCRA) group. A HCRA agent is designed for fault diagnosis at different levels of system granularity. An OCRA agent is designed for operational abnormalities. Either a model-based or knowledge-based supervision method could be

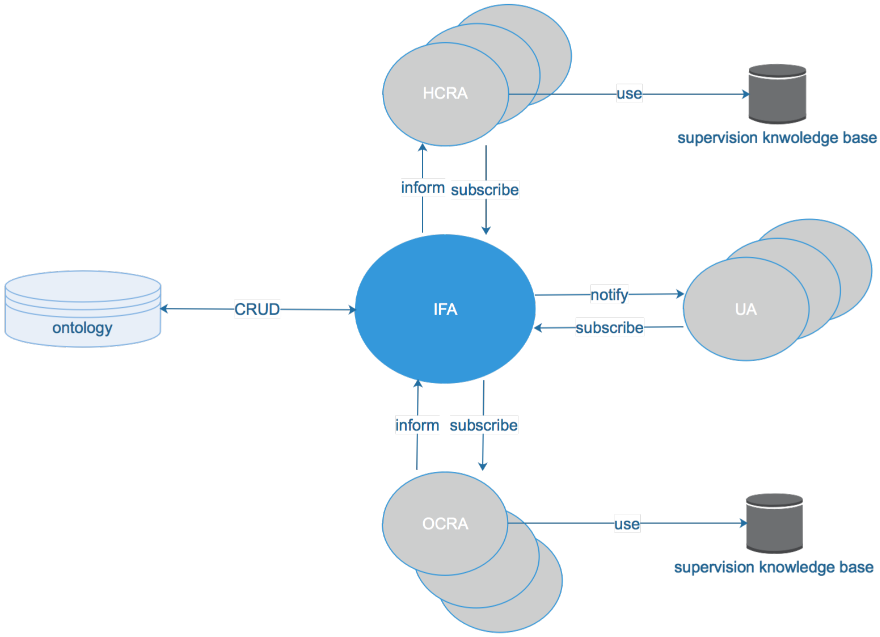

Figure 6.2: The interaction protocol of the MAS

embedded into the agent. The UA contains an agent that corresponds to different end users with different interests. It maintains connections with a mediator agent. This means the diagnosis and supervision results can be acquired in real time.

The agent communication follows a subscription interaction protocol. The initiator (e.g., supervision agent and user agent) send subscription requests to the participator (in formation mediator agent) indicating its desired information. If this action is successful, a permanent communication channel is established between the initiator and participator. The advantage of adapting the subscription protocol is that only the necessary information is delivered to the target agent.

Agent communication and coordination

Figure 6.3 presents an example of how agents communicate with each other and how different components are linked together. A belt tear condition supervision process was selected for illustrative purposes (a detailed case study will be given in section 6.4). To supervise the belt tear condition, three agents are considered. First, an IFA is considered. This is the central coordinator. Second, a belt tear supervision agent is in charge of the main assessment task. Third, a maintenance agent describes the user of the supervision result. In addition to the agents, there are two important components involved, a belt tear supervision method and an ontology. A belt tear supervision

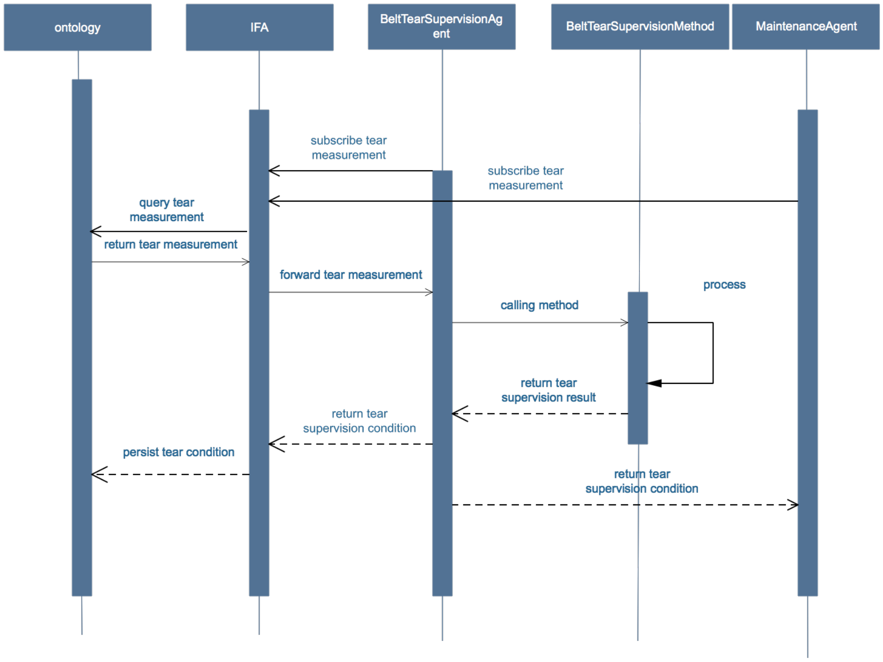

Figure 6.3: An sequence diagram of agent communication

method includes intelligence to assess the tear condition. An ontology stores and organises all relevant context information.

A belt tear supervision agent sends the subscription to an IFA to subscribe the tear measurements. The IFA will continuously communicate with the ontology to query the tear measurement. As soon as the measurement is available, it will forward the information to the belt tear supervision agent. Subsequently, the belt tear agent will call the tear supervision method for assessment. The assessed result will then be returned to the IFA. Following this, the IFA will forward the tear condition to the ontology for further processing. At this time, the maintenance agent will also receive information about the tear condition.

The following section will elaborate on the ontology design and the integration of agent and the ontology.

6.2.2 Ontology

This section introduces details about the implementation of ontology. This includes the design of ontology class, the design of ontology object property, and the design of ontology data property.

Ontology design method

As introduced in section 3.4.3, an ontology is a tool used to capture and specify domain knowledge into a machine understandable format. In an ontology, the domain knowledge is modelled by concept, relationships, and individuals. In principle, any concepts, relationships, and individuals can be defined in an ontology. It is up to an ontology engineer to determine the most critical and important concepts. As suggest by Evchina et al. (2015), two factors should be considered in line with the construction of a domain ontology: (1) The concepts. These are defined in an ontology and should have enough expressiveness to the application; (2) It should be simple and available for further extension.

To design an ontology for a domain, a common approach suggested within the literature is to adapt the existing ontology and possibly extend or refine it to suit specific needs. However, with respect to the reliability assessments of large scale equipment, particularly in a PL system, no suitable ontology exists to fit the scope. Therefore, it was necessary to define the ontology from scratch. In section 3.4.3, three approaches are introduced. First, the top-down approach requires a systematic view of the domain. Second, the bottom-up approach enables fast development without considering the hierarchical structure of the domain. This can capture dynamic features more quickly but lacks generality. Third, according to Uschold and Gruninger (1996), the use of the middle-out approach is a suitable method. This is because it captures the concept in the middle which is more descriptive and application-specific. In the following section, the middle-out approach is adapted to develop the **ontoSupervision**.

Scope of ontology

The definition of context-aware supervision is given in the section 3.4.1. Based on this definition, the scope of interests are as follows:

- System related concept: this defines the object that needs to be monitored and supervised. Concepts under this scope include components, processes, and devices.
- Condition supervision related concept: this captures the main concept of supervision. Concepts under this scope include system conditions, system state, supervision agent and method.
- User related concept: this incorporates user interests into consideration. Concepts under this scope include the type of users, and the interests of users.

In light of this, the classes hierarchy of **ontoSupervision** is presented in following section.

ontoSupervision class hierarchy

In this section, the main classes in the **ontoSupervision** are designed. First, the most important concepts will be identified and further expanded.

(I) **System**: context-aware supervision is to improve the reliability assessment of the large-scale material handling system in a PL system. Consequently, the equipment is the core object. To capture the knowledge of a piece of equipment, the concept of system is introduced. This is understood as the core class and contains individuals to be supervised. As discussed in Pang (2010)), a large-scale belt conveyor system consists of different components, processes, and devices. Consequently, several concepts are also categorised under the scope of system. These include system components, system processes, and system devices. Put simply, it structures a description regarding different aspects of a system. The hierarchy of concept **System** is outlined in Figure 6.4.

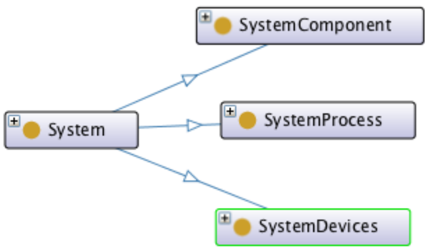

Figure 6.4: **ontoSupervision**: A hierarchy structure of concept **System**

Figure 6.5 depicts a partial tree-like structure of the concepts under the scope of **System**. It should be noted that the link between each class (concept) represent a "is-a" relationship. For instance, a **BrakingProcess** is a **SystemProcess** and a **Belt** is a **SystemComponent**. By categorising more concepts, a more concrete description of system can be given.

(II) **Condition**: The goal of a reliability assessment is to determine the system conditions. As such, another core concept is defined which is **Condition**. This incorporates a hierarchy of different system conditions. It basically represents the supervision results of the systems. Sub classes include component conditions, operation conditions, and condition partitions. Figure 6.6 depicts the level of **Condition** class.

The component condition presents the conditions at component level. As shown in Figure 6.6, a **BeltTearCondition** is a **ComponentCondition**. Since a large-scale system can be represented by any piece of equipment or a group of equipment, thus the component condition is applied to any level of system hierarchy. Not only does this consider the condition of system itself but also the condition of any peripheral devices.

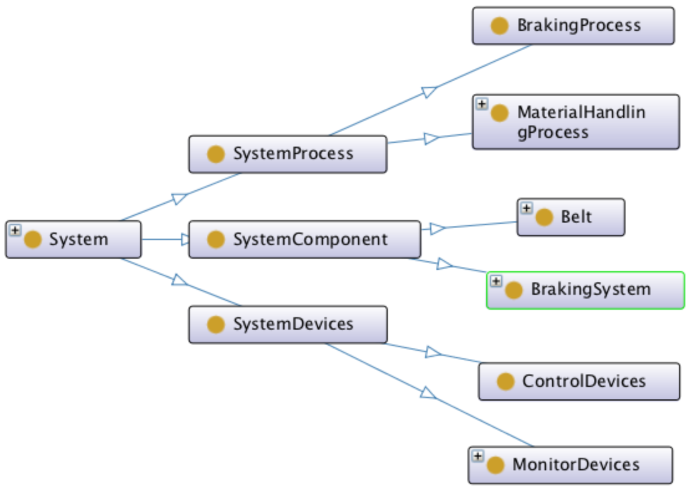

Figure 6.5: **ontoSupervision**: A sub-level structure of concept **System**

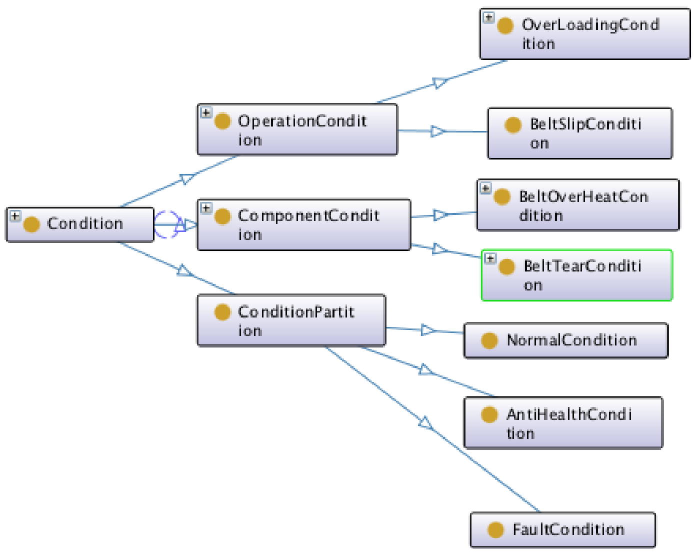

Figure 6.6: **ontoSupervision**: A hierarchy structure of concept **Condition**

The operational condition concerns the abnormality introduced during system operations. It is not necessarily induced by any fault in the components but may be caused by misuse or incorrect configuration of the system. Eventually, this triggers catastrophic conditions if not handled correctly. As shown in Figure 6.6, an **OverLoadingCondition** is an **OperationCondition**.

Finally, a **ConditionPartition** is presented. This adopts the design pattern in ontology engineering to refine descriptions for certain classes. In our case, the **ConditionPartition** consists of three subclasses namely **NormalCondition**, (expected behavior in given state), **AntiHealthCondition** (deviations from normal behaviour. This requires attention before failure happens), and **FaultCondition**(instant actions are required and it normally leads to system shut down). This is used to describe the damage levels for both the component conditions and the operation conditions.

(III) **State**: The concept of state is important for a system supervision task. It defines the possible state that a system could be in at certain point in time. In essence, each supervision method has a scope of applicability, indicating that different result can be generated under different states. Figure 6.7 depicts the class hierarchy of class **State**. It includes four different subClasses: **SteadyState**, **MaintenanceState**, **OperatingState**, and **TransientState**.

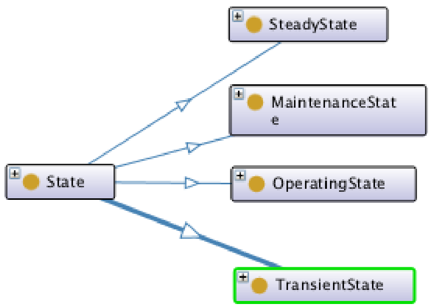

Figure 6.7: **ontoSupervision**: A hierarchy structure of concept **State**

(IV) **Resources**: In order to determine the system condition, the first step is to collect relevant information. The concept of resources class represents system information that the supervision methods need to access. As an upper concept, it does not intend to specify any specific information but rather it concerns the type of information. In the ontoSupervision, two types of resources are specified, namely, static resources and dynamic resources. Figure 6.8 depicts a hierarchy of class **Resources**. The difference between them refers to how it retrieves information and the temporal properties it possesses. **StaticResoures** represent the resources that do not change over time. For example, this includes system specific data and historical log information. Conversely, **DynamicResources** are acquired in real-time. For example, this includes the sensor data retrieved in run-time. Both resources will be valuable in determining the system conditions. Therefore, they are crucial for a reliability assessment.

(V) **SystemSupervisionAgent and SystemSupervisionMethod**: The concept of **systemSupervisionAgent** and its method serves as a bridge to connect the agent system and ontology model. The agent system is responsible for executing supervision tasks. By instantiating the concept, each deployed agent together with its attached information are firmly registered with the ontology. This allows the ontology model to monitor

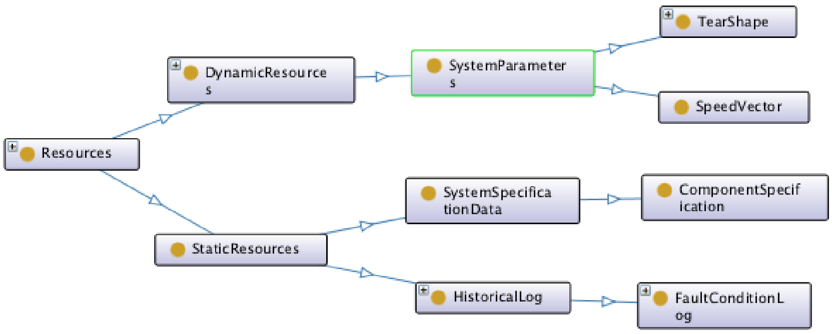

Figure 6.8: **ontoSupervision**: A hierarchy structure of concept **Resource**

and manage the agent system. The concept of **SystemSupervisionMethod** represents methods that actually perform condition identification, monitoring, diagnosis, and risk assessment tasks.

Figure 6.9 depicts a class hierarchy of **SystemSupervisionAgent**. It can be seen that each **Condition** defined previously has a corresponding **SystemSupervisionAgent**. A class of **OperationSupervisionAgent** is a **SystemSupervisionAgent** which is dedicated to **OperationCondidition**. For example, an **OverLoadingSupervisionAgent** is responsible for supervising an **OverLoadingCondition**. The same principle applies to **ComponentSupervisionAgent**. The class **UserAgent** represents end-user interests which can be treated as a supervision result consumer.

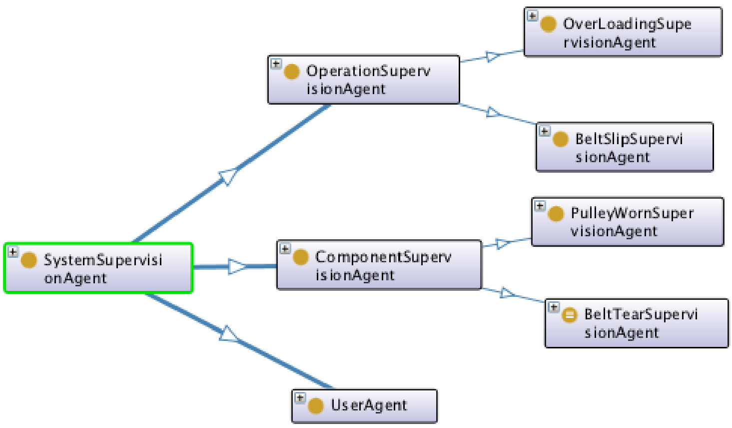

Figure 6.9: **ontoSupervision**: A hierarchy structure of concept **SupervisionAgent**

Figure 6.10 depicts a class hierarchy of **SystemSupervisionMethod**. A general principle is that for each **SystemSupervisionAgent**, a **SystemSupervisionMethod** is defined and associated. For example, class **PulleyWornSupervisionMethod** is a method that used to determine the pulley conditions. This will be associated together with the **PulleyWornSupervisionAgent**. The semantic meaning is that each supervision agent will use the supervision method to perform the supervision task.

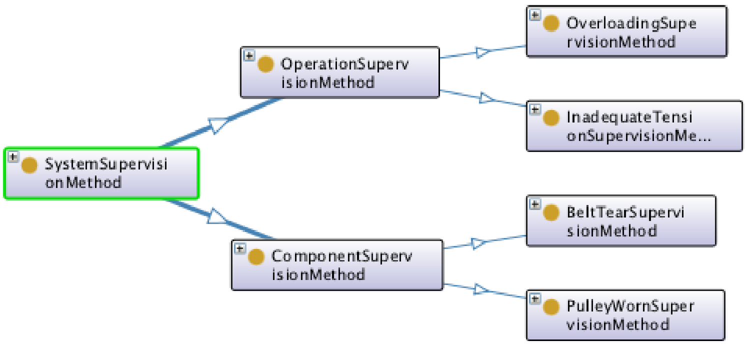

Figure 6.10: **ontoSupervision**: A hierarchy structure of concept **SupervisionMethod**

(VI) **Alarm**: This presents a category of different levels of alarm triggered by ongoing conditions in the system. It should be designed according to the domain of applications and the activation of alarm could be reasoned through ontology with the support of commercial reasoners.

(VII) **Users**: This concept represents the users that are actually using the system as end users. Individuals are further specified by data properties such as roles, areas of interest, areas of responsibility, and status. The final information delivered then relies on individual responsibility. As shown in Figure 6.11,three users have been identified: **MainteanceOperator**, **SystemEngineer**, and **ContractManufactor**. The **MainteanceOperator** represents the user who uses the supervision result to perform maintenance work. The **SystemEngineer** represents the user who determines which condition needs to be supervised based on which method. The **ContractManufactor** represents the user who provides system information.

ontoSupervision object property

In the previous section, the primary classes of the **ontoSupervision** were explained. In total, seven categories of concept were identified. However, the class and its subclasses defined in **ontoSupervision** alone cannot provide sufficient information to fully understand the domain. Consequently, the properties of the classes must be defined in the **ontoSupervision** ontology. There are two kinds of properties include object property and data property. The object property is used to define the relationship between

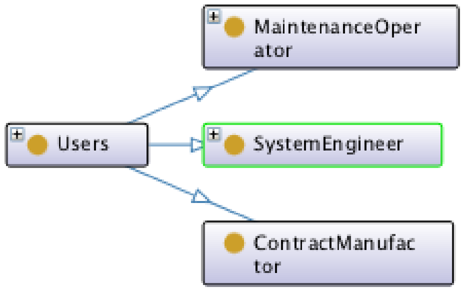

Figure 6.11: **ontoSupervision**: A hierarchy structure of concept **User**

classes. The data property determines the data values each class should have. In this section, the object property will be discussed. The data property will be addressed in next section.

Figure 6.12 depicts the ontology which consists of fundamental classes and relationships that model the contexts and necessary entities. This consists eight main classes and each class also contains sub-Classes. Classes are interrelated via the object property. The primary object properties will be discussed as follows.

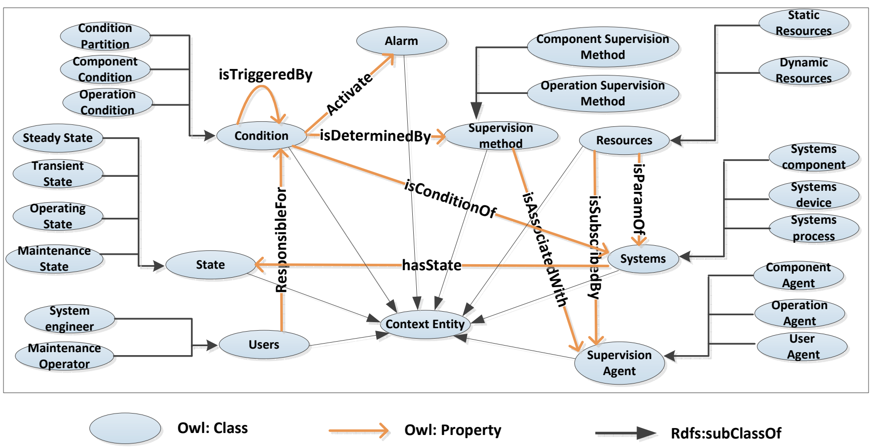

Figure 6.12: **ontoSupervision**: the definition of classes and relationships

(1) **Resource** *isParameterOf* **Systems**

The class **Resource** contains data that is used to describe system conditions. This can be a measured data, a system specification, or a historical data. Thus any individuals that belongs to **Resource** has a relationship of *isParametersOf* with class **System**.

(2) **State** *isStateOf* **Systems**

The class **State** describes different system states. This includes static state or transient state. Thus, it is intuitive to define the relationship *isStateOf* between classes **State** and **Systems**.

(3) **Systems** *hasConditionOf* **Conditions** The goal of designing **ontoSupervision** is to support reliability assessment. In another words, it determines the condition of a system. The class **Systems** is the core class in which to model the supervised equipment and the class **Condition** is the core class in which to represent the supervised condition. Thus, it is important to connect the class **Systems** and **Condition** with a relationship of *hasConditionOf*.

(4) **Condition** *isDeterminedBy* **SupervisionMethod**

Each condition is determined by applying specific method. Such methods can be a model-based approach or a knowledge-based approach. Depending on the nature of the condition, the method is selected. The object property *isDeterminedBy* is used to link the class **Condition** and **SupervisionMethod**.

(5) **SupervisionMethod** *isAssociatedWith* **SupervisionAgent**

As discussed in previous section, each **SupervisionMethod** is associated with a **SupervisionAgent**. The **SupervisionAgent** provides a running environment in which the supervision method can be executed. As such, a relationship *isAssocatedWith* is established.

(6) **SupervisionAgent***Subscribes* **Resources**

To execute supervision method, data needs to be provided. Different supervision methods rely on different data source. For example, a belt tear supervision method needs the tear shape measurement in order to quantify the level of damage. A belt energy supervision method needs the belt speed as the input. Consequently, a relationship is required to specify the data source and supervision condition. Since the **SupervisionAgent** provides the environment for method execution, each agent should *subscribes* data from **Resources**. The agent will then provide the data to its associated method for supervision.

(7) **Condition** *isTriggeredBy* **Condition**

The condition is not independent. This means that a condition may be induced or triggered by another condition. For example, a belt brake pad abrasion condition or a low brake hydraulic pressure condition may lead to a longer braking time. Therefore, if a long braking time condition is supervised, it can be triggered by either a low brake hydraulic pressure condition or a belt brake pad abrasion condition. To reveal the intrinsic relationship of different conditions, the object property *isTriggeredBy* is used.

(8) **Condition** *Activates* **Alarm**

When a **Condition** is supervised, it should reflect the severity level to the end user. In **ontoSuerpvision**, the class **Alarm** is used to quantify the level. The class **Alarm** presents a hierarchy of on-going events from the system, and should be linked with the supervised condition. Thus, the object property **Activates** is used to link **Condition** and **Alarm**.

(9) **User** *ResponsibleFor* **Condition**

The class **User** models different stakeholders that have interests in the supervised conditions. For example, maintenance workers need to understand the condition and take the proper action. Thus, a relationship *ResponsibleFor* is used to link the **User** and **Condition**.

To better illustrates the use of object properties, an example is given as shown in Figure 6.13. The semantic meaning of the example is given as: **Belt** is a subclass of **System**, it has a condition of **BeltTearCondition**. The **BeltTearCondition** can be determined by a method **BeltTearSupervisionMethod**. The **BeltTearSupervisionMethod** is associated with a **BeltTearSUpervisionAgent** which subscribes the parameter **TearShape**. The parameter **TearShape** is the input data for tear condition supervision. Upon successful supervision, the **BeltTearCondition** will activate an **Alarm** and the corresponding **MaintenanceOperator** will be notified.

Despite the nature of different conditions, the above semantic meaning is generic and can be applied to any conditions that exists in the system. It provides a mechanism to link different concepts together which makes information context-aware. For example, if a measurement is available (e.g., tear shape on the belt), the **ontoSupervision** is capable of finding all the relevant information linked with the measurement. For example, the **ontoSupervision** knows which parameter is linked to which component, which method to apply to determine which condition, which condition activate which alarm, and so on.

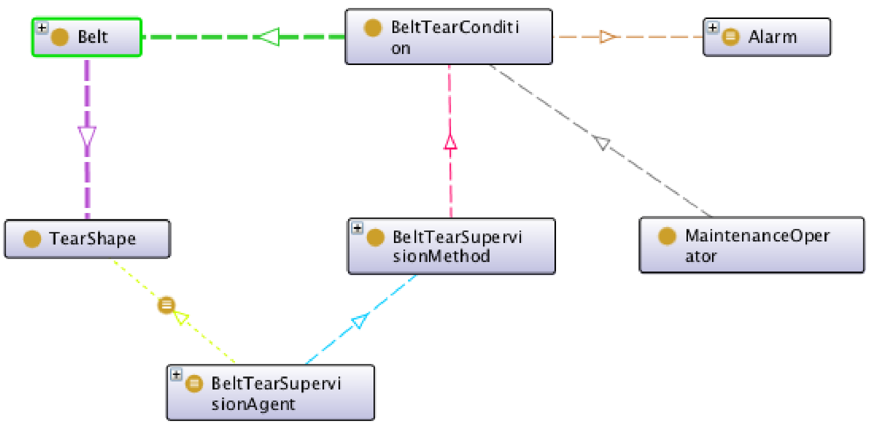

Figure 6.13: **ontoSupervision**: An example of object properties

ontoSupervision data property

In the previous section, the object property of **ontoSupervision** was discussed. The object property expresses the relationships between classes. In this section, another property termed data property is discussed. The data property describes the relationship between individuals and data values. The data property does not describe the relationship between classes, rather it provides properties within single classes. Via data property, each individual from a class can be described using numerical values.

As discussed, the data property is used to describe the single class. In other words, it is class-specific. Different classes have different data properties. To give a demonstration of how data property is used, an example is given below.

The data property of individuals from the class **Condition**: When a condition is supervised, a description is required to describe the condition property. The condition property should contain the supervised result, the action needed, and other useful information that could help end users to take proper actions. An example of a supervision result of a **BeltTearCondition** is given in Figure 6.14, a belt tear condition can be fully described by the following properties: the belt width, the broken cable percentage, the diameter, and the extension factor. By filling the value of each data property, a complete description of the belt tear condition can be determined.

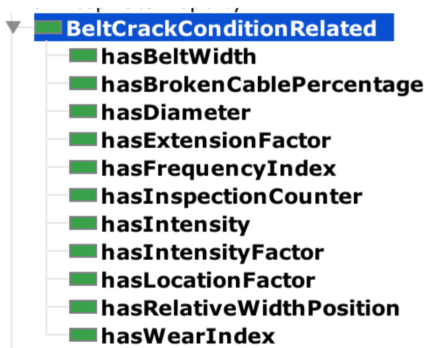

Figure 6.14: **ontoSupervision**: an example of data property

Applicability of ontoSupervision

The most important aspects of designing the **ontoSupervision** have been discussed. These include class, object property, and data property. In general, the **ontoSupervision** can be used in two ways. First, it can be used passively as a database. In this case, users can use ontology to query data and domain knowledge. Second, the ontology can be used actively. In this case, ontology is integrated as part of the decision-making engine to integrate information and infer new knowledge. It can also control and coordinate other modules in the system. In this section, we will discuss three ways of using **ontoSupervision** to support the reliability assessment of a belt conveyor system.

- Knowledge sharing: A straightforward way of using ontology is to enable knowledge sharing of a domain. This is because ontology contains the domain knowledge in a well-structured hierarchy. Users can view the structure of a class, find the associated class via the object property, and check the attributes of a class via data property. For example, a user can use **ontoSupervision** to explore information about the conditions that are currently being supervised. As shown in Figure 6.15, the user can view the conditions related to components by zooming into the category of **ComponentCondition**. Furthermore, by zooming into each condition, such as **BeltTearCondition**, the associated classes can be found and accessed via the object properties. For example, via object property *isDeterminedBy*, the user can easily find the methods that are currently used to supervise the condition.

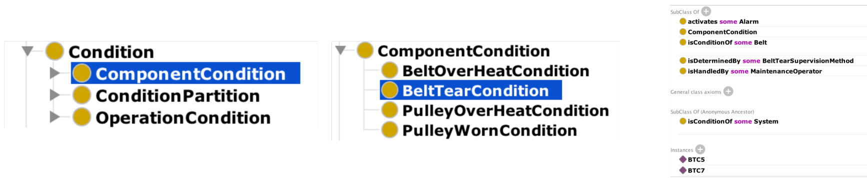

Figure 6.15: **ontoSupervision**: A zoom-in view of class **Condition**, its subClasses and annotations

- Knowledge query: Knowledge queries enable a user to acquire information from ontology by using meaningful query language. As discussed previously, domain knowledge is modeled as concepts and related via property. The user can use the defined properties to acquire information via a certain reasoners (e.g., Fact++ in protege). An example is given as shown in Figure 6.16. The semantic meaning of the query is *which condition can be determined by means of belt tear supervision method?*. The query result suggests that the **BeltTearCondition** is one that currently supervised by **BeltTearSupervisionMethod**.

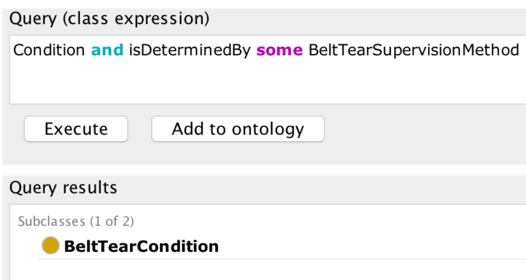

Figure 6.16: ontoSupervision: an example of ontology query

- Knowledge inferring: the two features provided above are passive. This means the user should take action to decipher information from the ontology. As discussed, the ontology can also be used actively. One aspect is to infer new knowledge in real-time. This property is supported by built-in rules in ontology.
- Agent system control: In section 6.3.2, a discussion on integrating ontology and agent system is discussed. In light of this discussion, ontology will be assessed not only as a stand-alone tool for knowledge management, but also for focusing on how to integrate ontology with different tools to enable more advanced decision-making capabilities.

6.3 Ontology-agent Integration

In section 6.2.1, the MAS system was chosen as the tool to support system supervision. In section 6.2.2, the ontology was chosen as the tool to support context aware modelling. In this section, how to integrate the multi-agent system and the ontology will be discussed.

6.3.1 Potential way of integration for context-aware supervision

Hadzic et al. (2009) claimed that an ontology and an agent-based system can complement one other, and that their integration could bring benefits at different levels. The benefits which include: problem decomposition, information retrieval and locating, agent communication, information manipulation and information presentation. Natarajan et al. (2012) developed an ontology (**ontoSafe**) that provides operational abnormality management of a large chemical plants using a multi-agent system (ENCORE) to facilitate the information retrieval and decision-making tasks. Dibley et al. (2012) used three different ontologies (sensor ontology, building ontology, and supporting ontology) to capture the semantics of a building environment where the monitoring and decision-making is supported by MAS. Mahdavi et al. (2013) developed a monitoring and control system for the fault identification in the cement production system using the agent system. Agent communication is coordinated by means of ontology.

In sum, current research regards the agent-ontology integration mainly focuses on two aspects: (1) using ontology to guarantee coherent communication among agents; (2) using ontology as a hybrid database for information and knowledge retrieval and location. An ontology behaves passively in an ontology-agent platform. Potential aspects such as information analysis and problem decomposition is rarely addressed. To enable context-aware supervision of large-scale equipment in a PL system, a potential method is to enable ontology to behave pro-actively. Two potential aspects of this are discussed as follows

- Agent coordination: The ontology will be treated as a controller of the agent system. Firstly, the ontology will continuously integrate the data and transform it into context information. Secondly, based on the context information, the ontology will determine the responsible agent. The responsible agent will be assigned to process certain context information. Moreover, the ontology could activate multiple agents to collaborate together for complex condition supervision.

- Information mining: The ontology will serve as a knowledge hub. The supervision results generated by the agent will be sent back to ontology for further processing. The processing include s storing of supervision results , reasoning new knowledge , and activating a new supervision thread if needed.

To achieve these two aspects, a specific integration scheme between agent and ontology will be defined and explained in following section.

6.3.2 Ontology-agent integration in ontoSupervision

As discussed in the previous section, the goal of an agent and ontology integration is to enable agent coordination and information mining. To achieve this goal, a specific interaction schema needs to be defined. The scheme can be viewed as a communication protocol between the ontology and agent system. In this section, a scheme called knowledge filtering template (KFT), as shown in Figure 6.17, is introduced. It defines patterns that the agent can use to acquire knowledge from the ontology. The use of the term knowledge instead of information is used because not only can ontology provide information by running certain queries, but is also able to yield inferred or reasoned facts dynamically, and such statements are helpful in coordinating agent behaviour. In detail, the design of KFT uses high level abstractions referring to the elements in the knowledge base. When certain conditions are fulfilled, or executable events are ready, the high-level abstraction terms are dynamically substituted with parameters defined in the ontology.

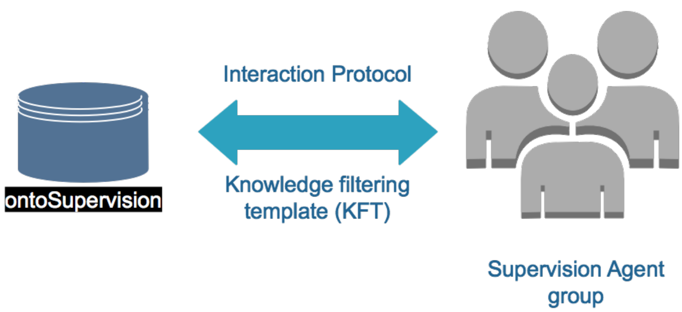

Figure 6.17: The general process of the Agent-Ontology integration

SPARQL language

The KFT is implemented via the SPARQL language [1]. The SPARQL language is a semantic query language which is able to retrieve and manipulate data stored in an ontology. The reasons for implementing KFT by SPARQL are two-fold: (1) first, the SPARQL language can be used to interact with ontology to retrieve data and knowledge. (2) second, the SPARQL query can be implemented via different programming languages, which gives it more power to adapt SPARQL in different software environments. It provides opportunities to embed SPARQL in a software agent and thus achieve the goal of agent-ontology integration. The basic format of a SPARQL query is given as follows:

[prefix declarations] SELECT <variable list> WHERE <graph pattern> .

The variable list in a SPARQL outlines the variables to be retrieved from the ontology. The graph pattern defines several predicates or constraints on the variables. These predicates or constrains should be satisfied.

Agent-ontology interaction via KFT

In this section, five KFTs are introduced. Each KFT is implemented via one SPARQL query. A sequence diagram is shown in Figure 6.18. As shown in the figure, the information facilitator (IF) agent serves the hub between the agent system and ontology. The KFT is initiated and executed in IF. For each KFT, the purpose and detailed use will be given as follows.

The use of KFT1 is considered as a configuration process that prepares all relevant information to set up the agent execution environment. The KFT1 executes inside a ticker behaviour which is activated periodically. The KFT1 is shown in Figure 6.19. The semantic meaning is given as: if data (e.g. a new sensor measurement) becomes available, the ontology should collect relevant knowledge in order to proceed with supervision. This includes the system state, the supervision agent that subscribes the data, the method associated with the agent, and the device that is used to acquire the data. For example, if more than one method is associated with the agent, the result of KFT1 will specify the most effective method and guide the agent to link with it. While it is initialised, the onto:AgentName will be replaced by the registered agent name.

[1] SPARQL: https://www.w3.org/2007/03/VLDB/

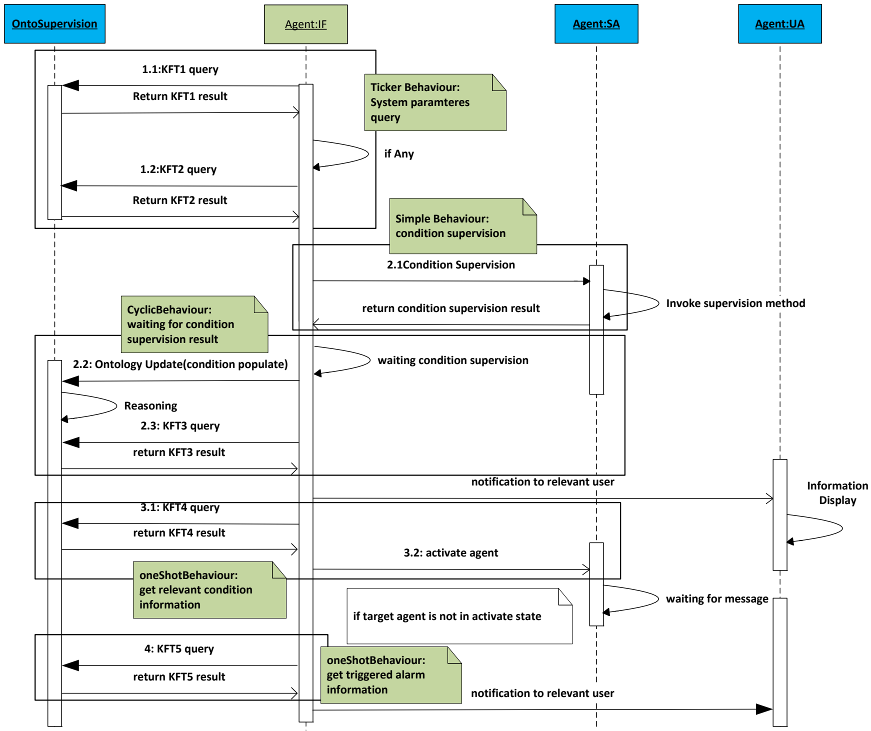

Figure 6.18: Sequence diagram: agent-agent & agent-ontology interaction

```
SELECT ?par ?agent ?system ?method ?timeStamp
?state ?measurement
WHERE {
            ?par onto:isSubscribedBy ?agent.
            ?par onto:isParametersOf ?system.
            ?agent rdf:type onto:AgentName.
            ?method onto:isAssociatedWith ?agent.
            ?system onto:hasState ?state.
            ?par onto:hasTimeStamp ?timeStamp.
            ?par onto:isMeasuredBy ?measurement.}
```

Figure 6.19: An example of knowledge filtering template 1

If the returned result set of the KFT1 is not empty, the KFT2 (see Figure 6.20) is activated. This aims to retrieve detailed data properties of the available measurement. For demonstration purposes, the query of a belt speed is given below. It queries the data properties of a belt speed that has been used for slip condition supervision of a belt conveyor system. As shown in the KFT2, the part marked in red can be replaced with any domain specific data properties.

```
SELECT ?par ?agent ?bs ?ps ?ts ?agentName
WHERE {
        ?par onto:isSubscribedBy ?agent.
        ?agent rdf:type ?agentName.
        ?par onto:hasBeltSpeed ?bs.
        ?par onto:hasTimeStamp ?ts.
        ?par onto:hasPulleySpeed ?ps.
        FILTER(?par=onto:currentMeasurement)}
```

Figure 6.20: An example of knowledge filtering template 2

Upon completion of KFT1 and 2, the supervision process is ready to begin. As shown in Figure 6.18, the IF agent will deliver information to relevant SA where a dedicated method will be invoked for condition supervision. Such a process is executed via an agents simple behaviour. The identified condition and specifications will be sent to IF and further populated into **ontoSupervision** by running a SPARQL update clause (see figure Figure 6.21). The template given below presents a partial template for updating a belt tear condition to ontology, the detailed illustration of the context will be given in section 6.4.

```
INSERT DATA
{
        onto:conditionName rdf:type onto:BeltTearCondition
        onto:conditionName onto:hasWearIndex onto:wearIndexValue
        onto:conditionName onto:hasDescription onto:description
        .....}
```

Figure 6.21: An example of update template

Following this, the IF can apply KFT3 (see Figure 6.22). The use of KFT3 is to reason the relationship between different conditions. In case a relationship **isCausedBy** was defined between two conditions. If one condition is supervised, the running of KFT3 will return the associated condition that might cause this supervised condition. In addition, the KFT3 will also return information include the activated alarm and relevant component associated with the condition.

```
SELECT ?cond ?cause ?alarm ?comp
WHERE{
        ?cond onto:isConditionOf ?comp.
        ?cond onto:causedBy ?cause.
        ?cond onto:activates ?alarm.
        FILTER(?cond=onto:condition)}
```

Figure 6.22: An example knowledge filtering template 3

The use of KFT4 (see Figure 6.23) is to discover intrinsic relationships between relevant conditions. Namely, when a condition is confirmed, the ontology should be able to identify the root cause and consequences associated with such condition if any exist. If the returned result set is not null, it indicates that the agents equivalent to the inferred condition should be activated and the supervision process should be activated. For example, a belt slip condition is identified when there is a difference between belt speed and pulley speed. However, the root cause could vary from improper power supply synchronisation between master and slave drive to inadequate tensions. The consequences could be a temperature increase in the belt. In this scenario, the ontology is required to infer such knowledge and activate the relevant supervision process to enable an in-depth evaluation alongside agent supervision and collaboration.

```
SELECT   ?Condition ?property ?confirmedCon  ?agent ?method ?agentInd
WHERE {
        ?condition rdfs:subClassOf ?restriction.
        ?condition rdfs:subClassOf ?restriction1.
        ?restriction owl:onProperty ?property.
        ?restriction owl:someValuesFrom  ?confirmedCon  .
        ?restriction1 owl:onProperty onto:isDeterminedBy.
        ?restriction1 owl:someValuesFrom  ?method.
        ?agent rdfs:subClassOf ?restriction2.
        ?restriction2 owl:onProperty onto:hasAssociateMethod.
        ?restriction2 owl:someValuesFrom ?method.
        ?agentInd rdf:type ?agent
        FILTER(?restrictionValue=onto:conditionType)
        FILTER(?property=onto:triggers||?property=onto:isTriggeredBy)}
```

Figure 6.23: An example of knowledge filtering template 4

Finally, through KFT5 (see Figure 6.24), the condition specification together with its root cause and consequences will be composed and become available to the person in charge.

In sum, the use of KFT leads to a semantic-rich processing engine, it makes the agent aware what reactions and/or behaviour it should/can invoke and furthermore which

```
SELECT ?cond ?condtype ?alarm ?desc ?rootCause ?user ?agent
WHERE{
        ?cond rdf:type ?condtype.
        ?condtype rdfs:subClassOf onto:ConditionPartition.
        ?cond onto:activates ?alarm.
        ?cond onto:hasDescription ?desc.
        ?cond onto:isTriggeredBy ?rootCause.
        ?cond onto:isHandledBy ?user.
        ?cond onto:isRepresentedBy ?agent.
        FILTER(?cond=onto:conditionName)}
```

Figure 6.24: An example of knowledge filtering template 5

new behaviour it should trigger. As such, the defined knowledge and reasoning facts from ontology are applied to control and manage the operation of the agent system.

6.3.3 Context rules

It should noted that the use of KFT intends to extract information from the ontology model by a well-defined template through query language (SPARQL in our case). In terms of context-awareness, it also requires a system capable of inferring new knowledge and facts based on current context. Such requirements can be fulfilled by enriching the ontology model with context rules. With respect to system supervision, fundamental tasks include triggering different levels of alarms and generating decisions based on incoming conditions. For instance, if the temperature of a critical component rises above a certain threshold, a fire alarm should be issued. If a component wear condition has been expanded significantly, then an immediate replacement should be activated. Such context rules are domain and application dependent, their pattern design and configuration should be done along with the design of application ontology. Some typical uses of context rules are discussed in section 6.4.

6.4 Cases study: A belt tear condition supervision

In this section, a case study of belt tear condition supervision is given. In section 6.4.1, an introduction of the case presented. In section 6.4.1, a fuzzy-logic based approach will be designed as the supervision method. In section 6.4.3 the context-aware supervision framework is applied to assist the belt tear condition supervision. The agent intelligence, context modelling, and decision-making process will be elaborated on.

6.4.1 Introduction of belt tear condition

In a PL system, a BCS is normally deployed in an open and hash environment. Consequently, critical components such as belts, idlers, and pulles are expected to deteriorate as the system ages. Among the various component fault conditions, four damages are reported including breadth tears, punctures, tongue-tears, and longitudinal tears. These account for 85 % of all damages (Lodewijks & Ottjes, 2005a). They have been categorised as the belt tear condition which could rapidly develop into a major problem if not handled properly.

In order to perform an accurate and timely assessment of a belt tear condition, three aspects need to be considered:

- Condition measurement: An inspection method is required to enable timely and accurate measurement of the tear conditions. The measurement includes the properties of a tear shape. It includes the length and width of the shape, and the position of a tear on the belt.
- Data mining: For a proper identification of the tear condition, the historical data (the measurement in the past) and the knowledge are required to understand the physical process that influences the damage growth. To link all the relevant information and knowledge together, a proper data mining process is required.
- Decision-making: When all required information and knowledge is available, a decision is made available. With respect to belt tear condition, a decision can be made on whether the tear condition on the belt should be fixed (e.g., replace the fault belt instantly) or the next inspection interval should be determined.

In current belt tear condition supervision practice, the mentioned aspects outlined above are not executed properly.

- Condition measurement: Currently, belt inspection largely relies on human visual inspection. This is unreliable and error-prone.
- Data mining: There is no proper data mining process. Systematic gathering and use of measurement, historical information, and knowledge for reliability assessment purposes is not well developed.
- Decision-making: The identification of a tear condition is executed by human experts. There is no intelligent decision-making system available that can perform decision-making automatically. Moreover, the current assessment heavily relies on human experts, meaning knowledge sharing and reuse can be difficult.

To improve the current practice of belt tear condition supervision, the above three aspects can be improved. In this case study, data mining and decision-making will be focused on.

6.4.2 Supervision method: a fuzzylogic-based approach

A supervision method is needed to support reliability assessment. Such method can be a model-based approach or a knowledge-based approach depending on the property of the supervised condition. To support the belt tear condition supervision, a fuzzy-logic based method is adopted as the supervision method. This was introduced in the work of Lodewijks and Ottjes (2005a).

The goal of using fuzzy-logic is to create straightforward advice on the maintenance actions of a belt tear condition. A simple membership function could be used to demonstrate a membership function between the tear measurement on the belt and to a scalar value which is used to indicate the severity of the crack. However, it is not applicable in practice. On the one hand, to quantify the degree of damage there are several factors that need to be considered. These include, for example, the shape, location, and size of the crack. Moreover, in order to outline proper maintenance action, the historical information is required to assess the propagation of the given crack.

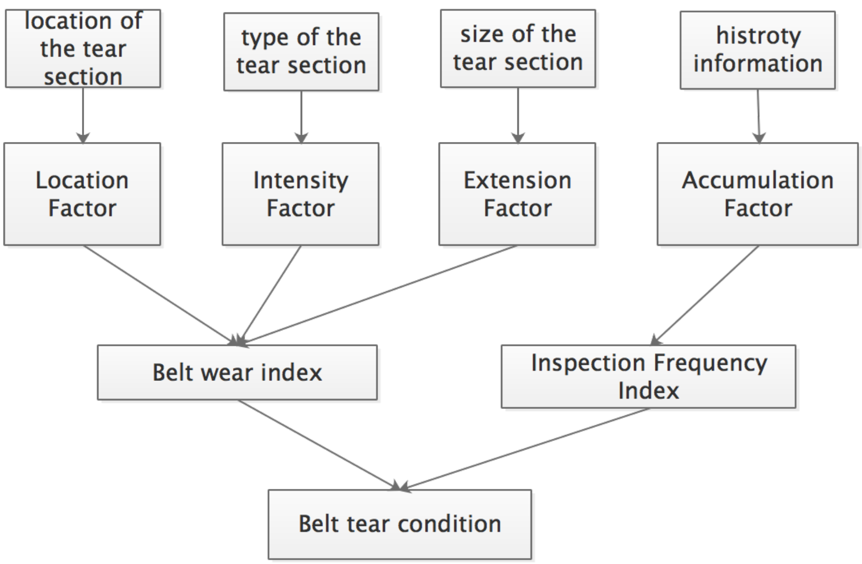

Figure 6.25: The factors that associated with the belt tear condition,derived from Lodewijks and Ottjes (2005a)

Figure 6.25 depicts the possible factors that are relevant to a belt tear condition. To quantitatively identify a tear shape, two indexes are defined, namely, a belt wear index (WI) and an inspection frequency index (FI). The WI is used to quantify the level of damage of a tear condition and the FI is used to determine the inspection interval of the tear condition. The WI and FI is determined by different factors and each factor is a mapped from an input measurement. A brief introduction of the developed fuzzy-logic method and the explanation of the belt wear index and belt frequency inspection index can be found in Appendix A. A detailed explanation of the WI and FI can be found in Lodewijks and Ottjes (2005a). In the following sections, how these two factors are calculated and how the maintenance action is made will be outlined.

6.4.3 Context-aware supervision

This section applies the proposed framework to supervise a belt tear condition. The focus of this section is to specify the work flow of the whole system. Firstly, a belt tear condition is simulated. Secondly, the whole supervision process will be divided in to three parts: context identification, context supervision and response action. For each part, the integration of ontology and agent system will be elaborated on.

Two inspections of a tear condition at distinctive time stamp 5 and 7 are considered. It assumes that a new tear shape was found on time stamp 5 and the next inspection is scheduled for time stamp 7. The supervision process should be able to identify the tear shape propagation pace and its potential damage during this period (time stamp 5 to 7) and eventually create a proposition for future activity (immediate repair or reduce the inspection intervals). The process can be divided into three steps as shown in Figure 6.26, namely the context identification, context supervision and response actions. The partial ontology presented in Figure 6.26 also shows how context information is organised and presented in **ontoSupervision**. The following sections will further elaborate on each process.

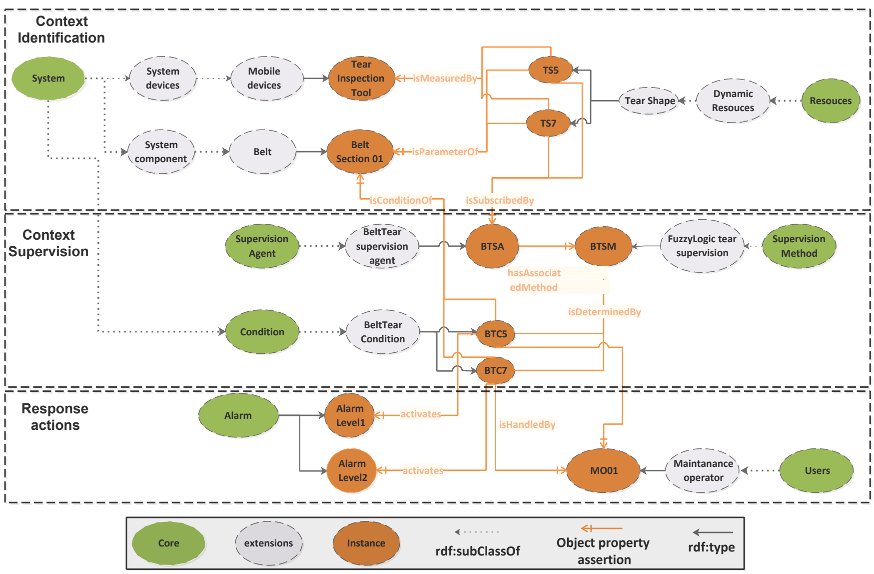

Figure 6.26: Ontology-MAS context aware supervision: a case study of belt tear condition supervision

Context Identification

The purpose of context identification is to extract context information from measurements. When a belt tear measurement is available, it will populate to the **ontoSupervision**. Inside the ontology, a SWRL rule will be activated as shown below:

```
TearShape(?par), BeltTearSupervisionAgent(?agent)
->isSubscribedBy(?par, ?agent)
```

By running the above rule, the tear measurement is associated with the belt tear supervision agent. Afterwards, all the context associated with the measurement can be located. Such contexts include which agent is responsible for this measurement, which method should be used, what was the system state when the measurement was taken, and so on. To achieve this, KFT 1 will be activated inside the supervision agent. As discussed in section 6.3.2, KFT 1 is used to collect all relevant information for context supervision. By initialising the KFT 1 template, the KFT 1 for a belt tear condition is show in Figure 6.27.

```
SELECT ?par ?agent ?system ?method ?timeStamp
?state ?measurement
WHERE {
    ?par onto:isSubscribedBy ?agent.
    ?par onto:isParametersOf ?system.
    ?agent rdf:type onto:BeltTearSupervisionAgent.
    ?method onto:isAssociatedWith ?agent.
    ?system onto:hasState ?state.
    ?par onto:hasTimeStamp ?timeStamp.
    ?par onto:isMeasuredBy ?measurement.}
```

Figure 6.27: KFT1 for belt tear condition

The belt tear supervision agent will run the above query to get the belt tear measurement conditions. The semantic meaning of the query is return any measurement that is subscribed by a belt tear agent. As discussed above, the SWRL rule ensures that the tear measurement is associated with the belt tear agent. As such, by running KFT 1, the belt tear agent is able to acquire the recent tear measurement from **ontoSupervision**.

The belt tear supervision agent will run the KFT1 to query the belt tear measurement conditions. The semantic meaning of the query is ”return any measurement that is subscribed by a belt tear agent”. As discussed above, the SWRL rule ensures that the tear measurement is associated with the belt tear agent. As such, by running KFT 1, the belt tear agent is able to acquire the recent tear measurement from **ontoSupervision**.

Followed by KFT1, a KFT 2 is executed by the belt tear agent. The purpose of running KFT 2 is to extract all relevant data properties related to a tear measurements. The extracted data properties will be used fro the supervision method. The KFT 2 for a belt tear condition is shown in Figure 6.28:

```
SELECT ?par ?agent  ?IF ?rwp ?bw ?d ?uri ?ts ?bcp
WHERE {
        ?par onto:isSubscribedBy ?agent.
        ?agent rdf:type onto:BeltTearSupervisionAgent.
        ?par onto:hasIntensity ?IF.
        ?par onto:hasRelativeWidthPosition ?rwp.
        ?par onto:hasDiameter ?d.
        ?par onto:hasTimeStamp ?ts.
        ?par onto:hasBeltWidth ?bw.
        ?par onto:hasBrokenCablePercentage ?bcp.
        ?par onto:hasURI ?uri.
        FILTER(?par=onto:currentMeasurement)}
```

Figure 6.28: KFT2 for belt tear condition

The context identification process is supported by executing KFT 1 and KFT2. The results obtained from KFT 1 and KFT 2 are summarised in Tables 6.1 and 6.2.

Table 6.1: KFT1 result

Parameter	Agent	System	Method	Time
tearShape5	BTSA	BeltSection01	BTSM	5
tearShape7	BTSA	BeltSection01	BTSM	7

Table 6.2: KFT 2 result

IF	d	rwp	bcp	bw	ts	uri
”irregular”	50mm	250mm	5%	1000mm	5	1
”irregular”	75mm	220mm	7%	1000mm	7	1

Context Supervision

Upon completion of identification, context information is ready for further processing. The data properties of a tear shape will be sent to the belt tear supervision agent (BSTA). Supported by the fuzzy-logic based method (see in Appendix A), a tear condition can be determined inside BSTA. The supervision result of a belt tear will generates a belt wear index. By running the KFT 3 which is shown in Figure 6.29, the ontology is able to identify the condition type and determine the alarm level. The result is summarized in Table 6.3.

Table 6.3: supervision result obtained by KFT3

Condition	System	Cause	Wear Index	Type
BTC5	BeltSection01	tearShape5	0.55	AntiHealth
BTC7	BeltSection02	tearShape7	0.74	Fault

```
SELECT ?cond ?cause ?alarm ?triggers ?comp
WHERE{
  ?cond onto:isConditionOf ?comp.
  ?cond onto:causedBy ?cause.
  ?cond onto:activates ?alarm.
   FILTER(?cond=onto:BeltTearCondition)}
```

Figure 6.29: KFT3 for belt tear condition

The final step in this phase is to check if the given condition has relationship *isTriggeredBy* or *Triggers* with any other conditions. In the case of a tear condition, as discussed, this is generated as the system ages. It is also common for most belt conveyor systems to continue functioning with tears on belt for a considerable amount of time. Thus, no server conditions could be triggered by a tear condition instantly. As a result, the execution of KFT4 would return an empty result.

Response Actions

When a condition is supervised, the final step is to create response actions.

- Fault identification: As explained in Lodewijks and Ottjes (2005a), a defuzzyfication process is required to translate the result (e.g., belt wear index, inspection frequency index) to a proper maintenance actions. In this case, we use the ontology rule inference to implement the defuzzyfication process. If the belt wear index is below 0.55, then the repair index is 0 (no repair needed) where if the value large than 0.55 then the repair index is 1 (repaired needed). The rule is shown in the block below:

```
BeltTearCondition(?condition), greaterThan(?level, 0),
hasWearIndex(?condition, ?level), lessThanOrEqual
(?level, 0.55)->AntiHealthCondition(?condition)

BeltTearCondition(?condition), greaterThan(?level, 0),
hasWearIndex(?condition, ?level), greaterThanOrEqual
(?level, 0.55)->Fault(?condition)
```

- Alarm generation:Based on the inferred condition type, a different alarm level is generated. The rules are shown in the block below: if the belt wear index is below 0.55, then an anti-health condition is identified and an alarm level 2 is activated. Otherwise, an alarm level 3 is activated.

- Identify responsible worker: Given the supervised conditions, a work force needs to be assigned to perform the corresponding maintenance work. In the **ontoSupervision**, all available work forces (e.g., maintenance workers) are registered.

```
AntiHealthCondition(?condition)-> activates
(?condition, AlarmLevel2)

FaultCondition(?condition)->activates
(?condition, AlarmLevel3)
```

The rule (shown in the block below) is used to determine which maintenance worker should be assigned. The semantic meaning is ”Given a belt tear condition which activates alarm level 3, if a maintenance worker is available and is responsible for the belt tear condition, then the maintenance worker is responsible for the supervised condition”.

```
ComponentCondition(?condition), isConditionOf(?condition, ?comp),
Belt(?comp), activates(?condition, ?alarmLevel), Alarm(?alarmLevel),
MaintenanceOperator(?MO), hasAreaOfResponsibility(?MO, ?aoi),
equal(?aoi, "BeltFault")
->responsibleFor(?MO, ?condition)
```

A complete supervision result for a single belt tear condition is can be acquired by running KFT5 as shown in Fig 6.30 and the results are summarised in Table 6.4. The supervision at time stamp 5 throws an anti-health condition and suggests a normal inspection. The second inspection at time stamp 7 discovers that the tear condition has been significantly expanded and an immediate repair has been suggested.

```
SELECT ?cond ?condtype ?alarm ?desc ?rootCause ?user ?agent
WHERE{
        ?cond rdf:type ?condtype.
        ?condtype rdfs:subClassOf onto:ConditionPartition.
        ?cond onto:activates ?alarm.
        ?cond onto:hasDescription ?desc.
        ?cond onto:isTriggeredBy ?rootCause.
        ?cond onto:isHandledBy ?user.
        ?cond onto:isRepresentedBy ?agent.
        FILTER(?cond=onto:BeltTearCondition)}
```

Figure 6.30: KFT5 for belt tear condition

Table 6.4: Supervision condition summary by running KFT5

Condition	Type	Alarm	Description	User	Agent
BTC5	AntiHealth	Level 2	No immediate action	Operator 1	MOA
BTC7	Fault	Level 3	Immediate action Requires	Operator 1	MOA

6.4.4 Towards implementations

The overall implementation framework with the software tools is shown in Figure **??**. The **ontoSupervision** ontology is created using open-source project Protege [2] which provides a user-friendly interface for creating, modifying and updating ontology. The ontology inference is implemented with a commercial reasoner pallet, other reasoners like HermiT and KAON2 are also applicable. In order to access data and knowledge from ontology, a query language is required. As discussed previously, a knowledge filtering template is used with a SPARQL language. Another fundamental part is the agent system. a FIPA compliant multi agent framework JADE [3] has been applied to set up the agent system. All supervision methods are implemented and packaged with in agent behaviour. The communication between JADE agents is supported by build-in ACL messages. To manage ontology-agent interaction, the Apache Jena [4] framework is utilised. It is a Java-based open source framework for constructing semantic web applications that allow any Java based systems to interact with ontology. The prototype system implementations can be found in Appendix B.

6.5 Discussion

This chapter aimed to implement the integrated DM system proposed in Chapter 4, to support reliability assessments of large-scale BCS. The contributions of this case study can be summarised as follows:

(1) Proposed and developed an ontology to support information integration and data mining: By developing the **ontoSupervision** ontology, the domain of equipment reliability assessment is modelled in terms of the concepts involved along with their relationships to each other. The **ontoSupervision** has eight classes. This includes **Condition**, **State**, **Systems**, **Resources**, **SupervisionMethod**, **SupervisionAgent**, **Alarm**, and **Users**. Each class can be decomposed into sub-classes in a hierarchical structure. Furthermore, the classes are interconnected with each other via specific properties (i.e., data property and object property). The **ontoSuypervision** is used to support information integration and mining from two perspectives:

- Automatic data categorisation: When a data is inserted to the ontology, it is represented as an instance of a class/sub-class. The ontology automatically links the asserted instance with its related instances based on the defined relationship. For example, when a **TearShape** instance is inserted, the **ontoSupervision** is able to find the instance of **BeltTearSupervisionAgent** this is subscribed to this instance. Furthermore the logic reasoner (in this case Protege) can reconstruct

[2]Protege: http://protege.stanford.edu/
[3]JADE: http://jade.tilab.com/
[4]Jena: https://jena.apache.org/

the ontology hierarchical structure by inferring/discovering new relationships based on the asserted instances.

- Querying and answering: The **ontoSupervision** can help to analyse reliability assessment related knowledge by answering domain queries. The domain objects (classes or instances) in **ontoSupervision** are specified with attributes. A number of queries based on those attributes can be posed by external modules/users. These queries can be answered by the ontology with the assistance of a software platform (e.g. in this case study, Protege was used). In this case study, a knowledge filtering template (KFT) is developed (see Section 6.3.2) and used to query ontology to obtain knowledge and information.

(2) Proposed and implemented a context-aware supervision system to support automatic decision making: The context-aware supervision system is used to enable automatic decision making. The system integrates the multi agent system and ontology into a hierarchical framework. In this system, the role of the multi agent system and ontology is given as follows:

- The multi agent system is used to execute BCS condition supervision tasks through agent intelligence and agent commutations. Each agent is responsible for one BCS condition (e.g., a healthiness or an operational related condition). Agents have distinct tasks but share a common goal. Periodically or on demand, they execute supervision algorithms and deliver condition supervision result. Firstly, the execution requirements are defined. For example, agent intelligence and required parameters . The information about the execution is stored in the knowledge-base (in this case, the ontology) or is requested by another agent of the system.

- The ontology presents the knowledge representation mechanism used by agents. For example, the **ontoSupervision** defines all the parameters that are required by an agent. It also has a registration of the available supervision method an agent can use. The agent can obtain this knowledge by querying **ontoSupervision** through the KFT. Furthermore, the **ontoSupervision** has a common understanding of the problem domain. It is able to track the supervised conditions of a BCS and possibly initiate a new agent task. For example, a low pressure in braking hydraulic unit condition may cause a long braking time of a BCS. As soon as a low pressure is supervised, the **ontoSupervision** may trigger the belt braking agent afterwards. In this sense, the **ontoSupervision** is also responsible for coordinating and controlling the execution of the MAS system. The benefit is to allow end-users to change the agent behaviour through ontology without interfering with the internal agent system.

Compared with the relevant studies (see section 2.5), this research explores the reliability assessment of a large-scale BCS from different perspectives. First, instead of focusing on a specific component condition assessment, this study aimed to develop a generic system suitable for integrating different condition assessments. As a result, it can be applied to different condition supervisions. This was accomplished by developing a context-aware supervision system. Tasks, such as condition supervision, are performed by agents, while end users have control over their execution through manipulation of ontology. Second, instead of developing a new condition measurement/supervision method, this study aimed to provide a method to achieve data mining, namely, categorising and linking heterogeneous data together and improving the decision-making process. This was accomplished by developing an ontology which captures the main entities, relationships, and knowledge in the reliability assessment. Third, this research also contributes the literature by proposing a new method of integrating a MAS with an ontology. It can be noted that in most fields, the agent technology is mainly used as a simulation and modelling tool for analysing their respective domain. The agent technology is rarely used as an enabler in the system implementation phase. As a result, applicability is limited. In this research, a prototype of the proposed integrated DM system was implemented by using open source technologies. For example, JADE was used to implement the agent system and Protege was used to develop ontology. Consequently, the applicability of the proposed ICT technology is demonstrated.

6.6 Conclusion

The content of this chapter provided answers to the sub-question 2.2, as listed in Chapter 1. Concerning the application and implementation of the proposed ICT framework, a context aware system was designed, implemented, and applied in a PL system. It is used to support the reliability assessment of a large scale belt conveyor system which is deployed in a dry bulk terminal. In the implementation,

- Equipment condition data can be (properly) transformed to context and further the context information which are associated with each other to enable an integrated decision-making by means of ontology.
- The reliability assessment (of components/equipment) can be made by an intelligent agent. The assessment algorithms are implemented and associated with the agent. By means of the agent intelligence and the agent communication, a reliability assessment support can be provided.

Given the case study, the advantages of using the implemented integrated DM system include:

- Supports automatic decision-making: This case study addresses the reliability assessment of a belt tear condition. Based on the measurement, the system can collect all the necessary resources (e.g. tear sharp measurement, historical assessment, responsible agent, and end user), and trigger the assessment process. The supervised result can be delivered to relevant users through a user interface with concrete maintenance actions. As such, the goal of automatic decision-making has been achieved.

- Supports information integration: Information integration is supported on two levels. On the condition supervision level, when the supervision process is triggered, heterogeneous data will be integrated within the ontology. This means the agent does not have to find all the necessary information. On the system level, the ontology provides a container where different kind of data sources can be injected and proceeded for the supervision purposes.

- Supports data association: The data association is mainly realised in the ontology. Each relevant entity in the reliability assessment process is modelled as a class. For example, the condition is a class and the measurements are also a class. Entities are associated with each other through object property. For example, a condition *isConditionOf* a component. During the supervision process, the agent can obtain all the context information by running the reasoner or SWRL rules. For example, when a new tear measurement is available, the historical measurements will also be automatically found and sent to the agent for assessment. Furthermore, by means of data association, the relevant entities involved in the supervision process are automatically bundled together.

Chapter 7

Conclusions and recommendations

The main goal of this thesis was to investigate methods for improving the performance of a port logistics system through effective use of ICT technologies. To achieve this goal, an integrated ICT framework has been designed and assessed through its implementation in two case studies. Section 7.1 presents the main conclusions of this thesis and provides answers to the research questions presented in Chapter 1. In section 7.2, recommendations for future research will be put forward.

7.1 Conclusions

Two research questions were proposed in the introduction of this thesis. In order to answer these research questions, the thesis was divided into two main parts. Part one includes Chapters 2 to 4. These chapters aimed to answer research question one. Part two includes Chapters 5 and 6. These chapters aimed to answer research question two. The answers provided are summarised as follows:

Research question 1. How can a new ICT framework be implemented and integrated into a Port Logistics system?

1.1 What is the research status of decision-making problems and the development status of ICT applications in a PL system?

Chapter 2 conducted a review of the most current research. This included an outline of the status of different levels of decision-making problems and the use of ICT in PL systems. As a result of this literature review, it was concluded that there are two challenges at two decision-making levels within a PL system, namely, collaborative hinterland barge transport planning at a tactical level and reliability assessments of large-scale belt conveyor systems at an operational level. Both challenges share the same root causes: (1) they lack an adequate platform for effective information sharing and integration; (2) they lack an effective decision-making approach. The literature review further revealed that ICT had been extensively applied in the PL system. However, the two challenges identified are currently not satisfactorily addressed by ICT.

Therefore, a new ICT framework in the PL system is required. The review also put forward two requirements:(1) support information exchange and sharing;(2) support intelligent decision making.

1.2 What are the potential ICT technologies that can be used to address decision-making problems

In order to solve the decision-making challenges, an ICT platform is required to provide support to information integration and intelligent decision-making. From an ICT perspective, implementation of an effective ICT platform can be broken down into two major elements: a middleware and an intelligent decision-making approach. In chapter 3, these two elements were selected. The middleware was used to support information integration and sharing. By comparing the three available middleware technologies (the object-based approach, the expert system, and the agent-based approach), the agent-based approach was chosen due to its advantages of effective collaboration and coordination, information exchange and sharing, and autonomy. These factors are suitable for the identified decision-making challenges. Concerning an intelligent DM approach, the choice was made according to the context in which the identified problems arise. For the collaborative planning challenge, a planning algorithm was required. By examining the available technological solutions (an exact method, a heuristics method, and a meta-heuristic method), a meta-heuristics approach was chosen as a result of consideration of its applicability. For the reliability assessment challenge, a context-aware system was chosen, and an ontology was selected as the technological solution for context modelling.

1.3 How can ICT technologies be integrated in a PL system?

In order to combine the proposed ICT technologies, an appropriate framework is required. In Chapter 4, an integrated ICT framework was proposed. Three architecture types were compared. These were centralised, distributed, and hierarchical architectures. It was concluded that both the centralised and the fully distributed system have disadvantages. The centralised system lacks flexibility, while the fully distributed system lacks the ability to integrate and coordinate information. A hierarchy architecture is a layered architecture that is designed to integrate the proposed ICT technologies into a generic framework. It can be adapted to address different challenges. This is because both the collaborative planning and reliability assessment issues can be decomposed into sub-problems, meaning decisions can be made by establishing effective coordination and integration of the decomposed sub-problems. It implements functions including information sharing and exchange, coordination and control, and supports decision-making.

Research question 2. What are the benefits of using a new ICT framework in a Port logistics system?

To answer research question two, two case studies were presented in Chapters 5 and 6. The goal was to apply the proposed ICT framework to address practical challenges in a PL system and investigate the potential for performance improvement. The results of the case studies showed that the benefits of integrating the ICT framework in a PL system include:

- improved information sharing and coordination between human-to-human and human-to-machine interactions in a PL system.
- improved decision-making at a tactical and an operational level in a PL system.

2.1 What are the benefits of using a new ICT framework at the tactical decision-making level in a PL system?

To answer this sub-question, a tactical level decision-making challenge in a container terminal was addressed in Chapter 5. More specifically, it focused on the hinterland barge transport planning problem. This case study concluded that the designed ICT framework is a useful tool for improving the coordination between the barge and terminal operators. Three main achievements can be highlighted.

- First, the multi-agent system supports information sharing and decision-making between the barge and terminal operators. Compared with the previous study, which proposed a distributed multi-agent system consists of two types of agents (the barge and terminal agents) (A. M. Douma et al., 2011), a mediator-based agent system was proposed in this research. By introducing a mediator-based MAS structure, the interactions between the barge and terminal agents are controlled and coordinated by a mediator agent. The interaction of the mediator-based agent system includes two phases. In the first phase, terminal agents allocate their available time slots. A barge agent determines its rotation route by collecting available time windows from terminal agents. In the second phase, a mediator is involved to assess the performance of the rotation plan regarding the average waiting time and the terminal utilization level. During barge and terminal agent interaction, the mediator agent can judge the performance by considering both the interests of barges and terminals. In case there is a room for further improvement, a new interaction is initiated. For example, a new round of negotiation can be initiated by the mediator if the terminal utilization can be further increased by adjusting the route of barges. The multi-agent system proposed in this research allows, in contrast with the distributed agent system, for balancing the interests of the barge and terminal operators through a high level of coordination. On the one hand, it allows barge and terminal operators to optimise their operations with limited information. On the other hand, the mediator agent takes the role of coordinator to adjust the interaction between the barge and terminal agents, thus guarantee better global performance.
- Second, to optimise the planning process, intelligent DM approaches are required. In this research, two meta-heuristics approaches, namely, a simulated

annealing and a non-dominant sorting genetic algorithm are selected and implemented. Further two different types of collaboration mechanisms are designed. In collaboration mechanism 1 (CM1), a leader-follower model is proposed. The barge agent (the follower) using a simulated annealing algorithm to explore the possible routes with a goal of minimising its waiting time in the port, where the terminal agents (the leader) generate the available time windows aims to improve its resource utilization. Given the barge rotation sequences and terminal time windows, the mediator agent tries to find an equilibrium of which the competing influences are balanced, namely, the global performance cannot be improved by either compromising barges or terminals interests. In collaboration mechanism 2 (CM2), the planning model is further extended by allowing each terminal has its occupancy level. A non-dominant genetic algorithm is developed to assist the mediator agent. The goal is to find the Pareto front of the results. To evaluate the performance of the proposed algorithms, a simulation study was performed. Section 5.5 proposed three scenarios, and each scenario has different settings, such as a different number of barges and number of terminals. Section 5.5 also compared the proposed collaboration mechanisms with two other strategies (M1 and M2, see section 5.5). Comparing CM1 with M2, the turnaround time (KPI-1) improved by 15 %, 30%, and 35% for each scenario respectively. In addition, the terminal resource utilisation rate (KPI-2) obtained by CM1 outperformed those generated by M1 with a 22%, 35%, and 38% improvement for each scenario respectively. The solution generated by CM2 was compared with CM1 for three designed scenarios. According to the simulation result, CM2 outperformed CM1. For KPI 1, the improvements were 5%, 8%, and 5% for each scenario. For KPI 2, the improvements were 4%, 3%, and 3% for each scenario.

- Finally, in this research, besides simulation studies, a prototype system was implemented which is based on the proposed integrated framework. The JADE was used as the agent management platform. A web-based user interface was implemented to support user interactions with the DM system. Compared with the studies related to the HBT planning problem which stayed on a simulation level with limited applicability (S. Li et al., 2014), the implementations discussed in this case study can be easily extended and implemented in practice.

2.2 What are the benefits of using a new ICT framework at the operational decision-making level in a PL system?

To answer this sub-question, an operational decision-making challenge in a dry bulk terminal was addressed in Chapter 6. This consisted of the problem associated with the reliability assessment of large-scale belt conveyors. This case study demonstrated that the proposed ICT framework is able to enhance the interoperability of human-to-machine interaction in a PL system. More specifically, it was able to transfer a machines status into meaningful information to end users for the purposes of reliability assessment. The main achievements can be highlighted as follows:

- First, the design of the multi-agent system aims to support condition supervision through agent communication and intelligence. This support is concerned with several tasks which include detection of disruptions, identification of condition, and suggestion of reaction decisions. A mediator-based multi-agent system is designed. It decomposed the reliability assessment of BCS into sub-tasks, and each agent is responsible for a single condition-supervision task such as a component condition and operational condition supervision. Using a subscription protocol, each agent subscribes one condition and the parameters related to the condition. The mediator agent provides coordination at a higher level. It is used to integrate the agents skills and knowledge, and supervision results.

- Second, to support knowledge management and information integration, an ontology (**ontoSupervision**) is implemented. The **ontoSupervision** developed in this study is a context-model which captures the reliability assessment related knowledge in the form of objects, concepts, and relationships that exist among them. The information contained in the ontology is linked with other related information. It provides an overview of a specific concept and helps in exploring the domain knowledge in a systematic way. The **ontoSupervision** defines all the parameters that are required by an agent. It also has a registration of the available supervision method used by an agent. The agent can obtain this knowledge by querying **ontoSupervision** through the proposed protocol (knowledge filtering template). Furthermore, the **ontoSupervision** has a common understanding of the problem domain. It can track the supervised conditions of BCS and trigger a new agent task. In light of this, the **ontoSupervision** is also responsible for coordinating and controlling the execution of the MAS system. Moreover, by implementing an agent-ontology integration, it allows end-users changing or modifying the agent behaviour through the ontology without interfering with the internal agent system.

- Finally, it can be noted that in most fields, the agent technology is mainly used as a simulation and modelling tool for analysing their respective domain (Moonen, 2009). The agent technology is rarely used as an enabler in the system implementation phase. As a result, applicability is limited. In this research, a prototype of an agent-ontology integrated DM system was implemented by using open source technologies. For example, JADE was used to implement the agent system, and Protege was used to develop an ontology. Consequently, the applicability of the proposed ICT technology is demonstrated.

7.2 Recommendations

The rapid growth of the PL system and the continuous development of ICT technologies provide opportunities for further integrating ICTs in the PL system. In light of this, several recommendations are made in the following section.

7.2.1 Recommendations for the future researches of the two case studies

This section outlines the research directions for the two case studies presented in this thesis.

For the HBT planning:

- Extension of the coordination model: The case study outlined in Chapter 5 mainly considered the planning activity at sea. For example, it considered the time a barge should arrive at a terminal. The coordination model could be further extended by considering the decision factors on land. For example, the availability of truck transport and storage areas, the location of containers, and the availability of crew on land could be considered. The integration of these factors could expect to result in a more reliable barge rotation plan and a more efficient terminal operation.

- Integration of tactical level planning with operational level control: In this thesis, the hinterland barge transport planning was considered as a tactical level decision-making problem. However, several operational level controls may also be integrated. The potential operational-level controls include ship speed control and ship collision avoidance in a congested port. By considering these factors, it is expected to result in a more reliable, energy-efficient barge transportation system.

For the reliability assessment of a large-scale BCS:

- Improvement to interoperability of equipment control: This case study concerns the reliability control of a belt conveyor system. An ontology was developed to improve the interoperability of human-machine interactions by integrating a domain concept and domain knowledge. However, the developed ontology was designed at a specific level. It only considered one type of equipment in the PL system. Extending the ontology model to incorporate more general knowledge about equipment reliability assessments in a PL system can be recommended for further research. Furthermore, research could be extended to include different equipment in order to verify the applicability of the framework.

- Integration of equipment control with tactical level decision making: By applying the framework, meaningful information regarding system conditions can be obtained. This category of information is important for maintenance work at an operational level. It can also be a valuable resource for risk management and resource planning at a tactical and strategic level. In light of this, future research could consider investigating how to integrate operational level information to the tactical and strategic level to enable more robust and efficient operations in a port.

7.2.2 Recommendations for ICT developments in future PL

This section proposes the potential future research direction with respect to ICT developments in future PL systems.

- Improvement of interoperability of the overall port logistics chain: By examining the recent developments of a PL system, the interoperability is expected to be further improved. Kalogeraki et al. (2018) argued that the growing complexity and the heterogeneity of critical infrastructures challenge the existing approaches and tools to respond to the frequent change of information dynamically. This results in a research gap named semantic gap. It reveals a lack of efficient knowledge sharing approach in the current maritime logistics chain, and this gap can potentially result in issues such as information security. Meanwhile, Glöckner and Ludwig (2017) also emphasised the challenge of the semantic gap in cloud logistics, when all the information from different logistics service providers need to be integrated. The challenge is caused by different descriptions offered by different providers.

  This thesis proposed an integrated DM system which integrates a MAS and ontology to improve the reliability assessment of equipment in a PL system. It aimed to improve the interoperability between human-machine interactions by addressing a specific case. To overcome the semantic gap and improve interoperability in a broader scope, two potential research directions are suggested: (1) a proposal of the theoretical and methodological approach of ontology-design in the PL domain with a focus on precision, time-efficiency, flexibility, and scalability. (2) enhancement of the decision-making capability of ontology reasoning by incorporating artificial intelligence. For example, introducing randomness into the ontology model with the assistance of the fuzzy logic approach (Bobillo & Straccia, 2016).

- ICT system architecture design: With respect to future ICT developments, increased attention should be paid on the design of the logistics information system architecture. In a recent position paper by Grefen et al. (2018), the need of an IT system design that supports integrated logistics service was emphasised. Challenges include the integration of different ICT technologies, and the information sharing and communication standards on a generic platform. In the review paper by Heilig and Voß (2016), a review of existing IT developments in container terminals was presented. The authors also suggested that interdisciplinary researches were required to further understand the requirements and implications of integrated electronic logistics platforms. In line with these challenges, a design-science related research is recommended to explore the requirements and needs of developing a more generic information system architecture in Port Logistics system. The proposed research should be able to examine whether, or to what extent and how, the design principles can be transferred or adapted to the

Port Logistics system. In other words, it should assess how to best apply different ICT technologies together with the architecture. Evaluation should focus on system feasibility, flexibility, scalability, and user acceptance.

7.2.3 Recommendations for applying emerging ICT technologies

This section proposes research directions in order to enable the application of emerging ICT technologies to improve the performance of a PL system further. This includes three major aspects which are presented in the section below.

Adapt the concept of the Internet of Things (IoT) to a PL system

In a PL system, a quality decision-making process relies on an accurate and timely acquisition of data. This includes container data, position-specific data, equipment specific data, and gate related data. Currently, not all data can be acquired in a timely manner. This is because current data collection methods lack a data-centric platform. A potential method to address this challenge is to adapt the concept of the Internet of Things (IoT). The concept of the IoT is to develop a network of devices by deploying smart sensors on a machine. This enables more efficient machine-to-machine and machine-to-human communication. The use of the IoT can enable smarter decision-making by utilising an immense amount of data collected from different contexts. For example, using sensor data acquired from machines can result in preventative maintenance once the right application is designed, such as the ICT framework designed in Chapter 6. The use of the IOT in port logistics is not as widespread compared to other industries. To bring the IOT into daily PL practice, several issues require further investigation. This includes research into smarter sensor design, wireless network infrastructure support, and data security.

Introduction of the concept of Machine learning and Big Data in the context of a PL system

Alongside increased digitisation, data-driven management and planning processes are expected to become more widespread in future PL practice. The core of a data-driven process includes two parts: (1) to enable a distributed and scalable data collection network (e.g. use of the IoT); (2) to develop an intelligent model to study the data and derive decisions out of the data. For the second aspect, machine learning techniques can be useful. Machine learning is an emerging methodology used to study and analyse historical data and derive data patterns. Put simply, the derived patterns are a valuable resource which helps decision makers to make decisions. Machine learning techniques have been well developed in different domains such as market demand predictions and facial recognition. More recently, there has been a demand from industry to investigate the potential benefits of applying machine learning techniques in a PL system.

In light of this demand, three topics at three decision levels are recommended for future research: (1) at a strategic level, research could consider investigating how best to apply ML techniques to predict the risks and uncertainties in the market; (2) at a tactical level, research could consider investigating how best to apply ML techniques to resource planning and reaching maximum capacity by predicting the possible factors such as customer demand and traffic volume; (3) at an operational level, research could consider investigating how best to apply ML techniques to predict machine conditions and create accurate preventive maintenance plans.

Investigation of the benefits of migrating from a legacy ICT system to a cloud-based system

In recent decades, different types of ICT systems have been developed and integrated to support the daily operations of a PL system. This includes, for example, in terminal operation systems and gate management systems. To cope with increasing customer demands and a highly competitive market, the existing ICT systems are expected to be updated. The goal is to fulfil the requirements, such as handling an increasing amount of data, enable timely and accurate information integration, and incorporate a more advanced and intelligent model. Therefore, a possible research direction is to develop a new ICT infrastructure that could enable this migration. One possible solution is to use cloud technology. It is recommended that an investigation into the possibility of migrating an existing legacy ICT system to a cloud-based system be conducted. Issues such as security, usability, and scalability require further exploration.

Bibliography

Abbate, S., Avvenuti, M., Corsini, P., & Vecchio, A. (2009). Localization of shipping containers in ports and terminals using wireless sensor networks. *2009 International Conference on Computational Science and Engineering*, *2*, 587-592.

Aksentijević, S., Tijan, E., & Hlaća, B. (2009). Importance of organizational information security in port community systems. *Information System Security, MIPRO*.

Amen, M. (2000). Heuristic methods for cost-oriented assembly line balancing: A survey. *International Journal of Production Economics*, *68*(1), 1–14.

Anicic, D., Fodor, P., Rudolph, S., Stühmer, R., Stojanovic, N., & Studer, R. (2010). A rule-based language for complex event processing and reasoning. *International Conference on Web Reasoning and Rule Systems*, 42–57.

ASEAN. (2011). Guidelines on strategic maintenance for port structures. *Ports and Harbours Bureau, Ministry of Land, Infrastructure, Transport and Tourism(MLIT), Japan*.

Aydogdu, Y. V., & Aksoy, S. (2015). A study on quantitative benefits of port community systems. *Maritime Policy & Management*, *42*(1), 1–10.

Aylett, R., Brazier, F., Jennings, N. R., Luck, M., Nwana, H., & Priest, C. (1998). Agent systems and applications. *The Knowledge Engineering Review*, *13*(3), 303–308.

Baader, F. (2003). *The description logic handbook: Theory, implementation and applications*. Cambridge university press.

Bäck, T., & Schwefel, H. P. (1993). An overview of evolutionary algorithms for parameter optimization. *Evolutionary computation*, *1*(1), 1–23.

BañOs, R., Ortega, J., Gil, C., FernáNdez, A., & De Toro, F. (2013). A simulated annealing-based parallel multi-objective approach to vehicle routing problems with time windows. *Expert Systems with Applications*, *40*(5), 1696–1707.

Baumgartner, L., Strack, C., Hossbach, B., Seidemann, M., Seeger, B., & Freisleben, B. (2015). Complex event processing for reactive security monitoring in virtualized computer systems. *Proceedings of the 9th ACM International Conference on Distributed Event-Based Systems*, 22–33.

Baykasoğlu, A., & Kaplanoğlu, V. (2015). An application oriented multi-agent based approach to dynamic load/truck planning. *Expert Systems with Applications*, *42*(15), 6008–6025.

Bellifemine, F., Poggi, A., & Rimassa, G. (2001). Developing multi-agent systems with a fipa-compliant agent framework. *Software: Practice and Experience*, *31*(2), 103–128.

Bellifemine, F. L., Caire, G., & Greenwood, D. (2007). *Developing multi-agent systems with jade*.

Bierwirth, C., & Meisel, F. (2010). A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, *202*(3), 615–627.

Bobillo, F., & Straccia, U. (2016). Optimising fuzzy description logic reasoners with general concept inclusion absorption. *Fuzzy Sets and Systems*, *292*, 98–129.

Bode, O., & Ferreira, F. (2014). Patent licensing: Stackelberg versus cournot models. *Discontinuity and Complexity in Nonlinear Physical Systems*, 409–420.

Bohanec, M., & Rajkovic, V. (1990). Expert system for decision making. *Sistemica*, *1*(1), 145–157.

Bricon Souf, N., & Newman, C. R. (2007). Context awareness in health care: A review. *international journal of medical informatics*, *76*(1), 2–12.

Brownlee, J. (2011). *Clever algorithms: nature-inspired programming recipes*. Jason Brownlee.

Bruzzone, A., & Signorile, R. (1998). Simulation and genetic algorithms for ship planning and shipyard layout. *Simulation*, *71*(2), 74–83.

Bugaric, U. S., Petrovic, D. B., Jeli, Z. V., & Petrovic, D. V. (2012). Optimal utilization of the terminal for bulk cargo unloading. *Simulation*, *88*(12), 1508–1521.

Cao, J. X., Lee, D.-H., Chen, J. H., & Shi, Q. (2010). The integrated yard truck and yard crane scheduling problem: Benders decomposition-based methods. *Transportation Research Part E: Logistics and Transportation Review*, *46*(3), 344–353.

Carlan, V., Sys, C., & Vanelslander, T. (2016). How port community systems can contribute to port competitiveness: Developing a costbenefit framework. *Research in Transportation Business & Management*, *19*, 51 - 64.

Carlo, H. J., Vis, I. F., & Roodbergen, K. J. (2014). Storage yard operations in container terminals: Literature overview, trends, and research directions. *European Journal of Operational Research*, *235*(2), 412–430.

Cepolina, S., & Ghiara, H. (2013). New trends in port strategies. Emerging role for ICT infrastructures. *Research in Transportation Business and Management*, *8*, 195–205.

Chan, F. T. S., Jha, A., & Tiwari, M. K. (2016). Bi-objective optimization of three echelon supply chain involving truck selection and loading using NSGA-II with heuristics algorithm. *Applied Soft Computing Journal*, *38*, 978–987.

Chang, S.-J. (2004). Development and analysis of ais applications as an efficient tool for vessel traffic service. *OCEANS'04. MTTS/IEEE TECHNO-OCEAN'04*, *4*, 2249–2253.

Chang, Y. S., Fan, C. T., Lo, W. T., Hung, W. C., & Yuan, S. M. (2015). Mobile cloud-based depression diagnosis using an ontology and a Bayesian network. *Future Generation Computer Systems*, *43-44*, 87–98.

Chao, S. L., & Lin, Y. L. (2010). Applying fuzzy ahp to select an automatic container number identification system in port terminals. *8th International Conference on Supply Chain Management and Information Systems (SCMIS)*, 1–7.

Chen, D., & Daclin, N. (2007). Barriers driven methodology for enterprise interoperability. *IFIP International Federation for Information Processing*, *243*(1), 453–460.

Chen, G., Govindan, K., & Yang, Z. (2013). Managing truck arrivals with time windows to alleviate gate congestion at container terminals. *International Journal of Production Economics*, *141*(1), 179–188.

Chioreanu, A., Brad, S., Porumb, C., & Porumb, S. (2014). E-maintenance ontology-based approach for heterogeneous distributed robotic production capabilities. *International Journal of Computer Integrated Manufacturing*, *28*(2), 200–212.

Christiansen, M., Fagerholt, K., Nygreen, B., & Ronen, D. (2007). Maritime transportation. *Handbooks in operations research and management science*, *14*, 189–284.

Christiansen, M., Fagerholt, K., & Ronen, D. (2004). Ship routing and scheduling: Status and perspectives. *Transportation science*, *38*(1), 1–18.

Chu, Y., You, F., Wassick, J. M., & Agarwal, A. (2015). Integrated planning and scheduling under production uncertainties: Bi-level model formulation and hybrid solution method. *Computers and Chemical Engineering*, *72*, 255–272.

Corry, P., & Kozan, E. (2006). An assignment model for dynamic load planning of intermodal trains. *Computers and Operations Research*, *33*(1), 1–17.

Crossland, A., Jones, D., & Wade, N. (2014). Planning the location and rating of distributed energy storage in lv networks using a genetic algorithm with simulated annealing. *International Journal of Electrical Power & Energy Systems*, *59*, 103–110.

Dai, Y., & Gao, Y. (2014). Real-time pricing decision making for retailer-wholesaler in smart grid based on game theory. *Abstract and Applied Analysis*, *2014*.

Darbra, R. M., Crawford, J., Haley, C., & Morrison, R. (2007). Safety culture and hazard risk perception of australian and new zealand maritime pilots. *Marine Policy*, *31*(6), 736–745.

Davidsson, P., Henesey, L., Ramstedt, L., Törnquist, J., & Wernstedt, F. (2005). An analysis of agent-based approaches to transport logistics. *Transportation Research Part C: Emerging Technologies*, *13*(4), 255–271.

Dawson, R. J., Peppe, R., & Wang, M. (2011). An agent-based model for risk-based flood incident management. *Natural Hazards*, *59*(1), 167–189.

De Langen, P. W., Van Den Berg, R., & Willeumier, A. (2012). A new approach to granting terminal concessions: the case of the Rotterdam World Gateway terminal. *Maritime Policy and Management*, *39*(1), 79–90.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, *6*(2), 182–197.

De La Cruz, A. L., Veeke, H., & Lodewijks, G. (2006). Prognostics in the control of logistics systems. *2006 IEEE International Conference on Service Operations* *and Logistics, and Informatics*, 1–5.

Dey, A. K. (2001). Understanding and using context. *Personal and ubiquitous computing*, *5*(1), 4–7.

Dibley, M., Li, H., Rezgui, Y., & Miles, J. (2012). An ontology framework for intelligent sensor-based building monitoring. *Automation in Construction*, *28*, 1–14.

Dibley, M., Li, H., Rezgui, Y., & Miles, J. (2015). An integrated framework utilising software agent reasoning and ontology models for sensor based building monitoring. *Journal of Civil Engineering and Management*, *21*(June), 356–375.

Ding, H., Benyoucef, L., & Xie, X. (2006). A simulation-based multi-objective genetic algorithm approach for networked enterprises optimization. *Engineering Applications of Artificial Intelligence*, *19*(6), 609–623.

Dong, C.-S. J., & Srinivasan, A. (2013). Agent-enabled service-oriented decision support systems. *Decision Support Systems*, *55*(1), 364–373.

Dorneles, Á. P., de Araújo, O. C., & Buriol, L. S. (2014). A fix-and-optimize heuristic for the high school timetabling problem. *Computers & Operations Research*, *52*, 29–38.

Douma, A., Schutten, M., & Schuur, P. (2009). Waiting profiles: An efficient protocol for enabling distributed planning of container barge rotations along terminals in the port of Rotterdam. *Transportation Research Part C: Emerging Technologies*, *17*(2), 133–148.

Douma, A., Van Hillegersberg, J., & Schuur, P. (2012, June). Design and evaluation of a simulation game to introduce a Multi-Agent system for barge handling in a seaport. *Decision Support Systems*, *53*(3), 465–472.

Douma, A. M. (2008). *Aligning the Operations of Barges and Terminals through Distributed Planning*. PHD thesis, University of Twente.

Douma, A. M., Schuur, P. C., & Schutten, J. M. J. (2011). Aligning barge and terminal operations using service-time profiles. *Flexible Services and Manufacturing Journal*, *23*(4), 385–421.

Drummond, N., Horridge, M., Stevens, R., Wroe, C., & Sampaio, S. (2007). Pizza ontology. *The University of Manchester*, *2*.

Duchessi, P., & Chengalur-Smith, I. (1998). Client/server benefits, problems, best practices. *Communications of the ACM*, *41*(5), 87–94.

Duinkerken, M. B., Evers, J. J., & Ottjes, J. A. (2002). Improving quay transport on automated container terminals. *Proceedings of the IASTED International Conference Applied Simulation and Modelling (ASM 2002)*.

Duinkerken, M. B., Ottjes, J. A., & Lodewijks, G. (2006). Comparison of routing strategies for agv systems using simulation. *Proceedings of the 38th conference on Winter simulation*, 1523–1530.

Dullaert, W., Neutens, T., Vanden Berghe, G., Vermeulen, T., Vernimmen, B., & Witlox, F. (2009). MamMoeT: An intelligent agent-based communication support platform for multimodal transport. *Expert Systems with Applications*, *36*(7), 10280–10287.

Ejigu, D., Scuturici, M., & Brunie, L. (2007). An ontology-based approach to context modeling and reasoning in pervasive computing. *Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops, 2007*, 14–19.

Elhdad, R., Chilamkurti, N., & Torabi, T. (2013). An ontology-based framework for process monitoring and maintenance in petroleum plant. *Journal of Loss Prevention in the Process Industries*, *26*(1), 104–116.

European Commission. (2008a). *An economic assessment of ict adoption and its impact on innovation and performance*. Berlin/Brussels.

Evchina, Y., Puttonen, J., Dvoryanchikova, A., & Lastra, J. L. M. (2015). Context-aware knowledge-based middleware for selective information delivery in data-intensive monitoring systems. *Engineering Applications of Artificial Intelligence*, *43*, 111–126.

Expósito-Izquierdo, C., Melián-Batista, B., & Moreno-Vega, M. (2012). Pre-marshalling problem: Heuristic solution method and instances generator. *Expert Systems with Applications*, *39*(9), 8337–8349.

Faccio, M., Ries, J., & Saggiorno, N. (2015). Simulated annealing approach to solve dual resource constrained job shop scheduling problems: layout impact analysis on solution quality. *International Journal of Mathematics in Operational Research*, *7*(6), 609–629.

Feng, F., Pang, Y., & Lodewijks, G. (2014). An intelligent agent-based information integrated platform for hinterland container transport. *2014 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, 84–89.

Feng, F., Pang, Y., & Lodewijks, G. (2015a). Application of hybrid algorithm to joint decision making in hinterland barge transport planning. *Proceedings of the 45th international conference on Computers & Industrial Engineering (CIE45)*.

Feng, F., Pang, Y., & Lodewijks, G. (2015b). Integrate multi-agent planning in hinterland transport: Design, implementation and evaluation. *Advanced Engineering Informatics*, *29*(4), 1055–1071.

Feng, F., Pang, Y., & Lodewijks, G. (2016). An intelligent context-aware system for logistics asset supervision service. *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems*, *8*, 1147–1152.

Feng, F., Pang, Y., & Lodewijks, G. (2018). Context-aware supervision framework for reliability control of large scale material handling system: an ontology-agent integrated approach. *Computers in industry (submitted)*.

Feng, F., Pang, Y., Lodewijks, G., & Li, W. (2015). Agent-based negotiation and decision-making for efficient hinterland transport plan. *2015 IEEE 19th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 395–400.

Feng, F., Pang, Y., Lodewijks, G., & Li, W. (2017). Collaborative framework of an intelligent agent system for efficient logistics transport planning. *Computers & Industrial Engineering*.

Fischer, K., Mu¨ ller, J. R. P., & Pischel, M. (1996). Cooperative transportation scheduling: an application domain for dai. *Applied Artificial Intelligence*, *10*(1), 1–34.

Fourie, J., Alport, M., Basson, J., & Padayachee, T. (2005). Condition monitoring of fabric-reinforced conveyor belting using digital x-ray imaging. *Bulk Solids Handling*, *25*(5), 290–294.

Franc, P., & Van der Horst, M. (2010). Understanding hinterland service integration by shipping lines and terminal operators: a theoretical and empirical analysis. *Journal of Transport Geography*, *18*(4), 557–566.

Galar, D., Thaduri, A., Catelani, M., & Ciani, L. (2015). Context awareness for maintenance decision making: A diagnosis and prognosis approach. *Measurement: Journal of the International Measurement Confederation*, *67*, 137–150.

Gao, R., Wang, L., Teti, R., Dornfeld, D., Kumara, S., Mori, M., & Helu, M. (2015). Cloud-enabled prognosis for manufacturing. *CIRP Annals - Manufacturing Technology*, *64*(2), 749–772.

Gaunitz, B., Roth, M., & Franczyk, B. (2015). Dynamic and scalable real-time analytics in logistics combining apache storm with complex event processing for enabling new business models in logistics. *International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE), 2015*, 289–294.

Gershwin, S. B. (1989). Hierarchical flow control: a framework for scheduling and planning discrete events in manufacturing systems. *Proceedings of the IEEE*, *77*(1), 195–209.

Giallombardo, G., Moccia, L., Salani, M., & Vacca, I. (2010). Modeling and solving the tactical berth allocation problem. *Transportation Research Part B: Methodological*, *44*(2), 232–245.

Glöckner, M., & Ludwig, A. (2017). Ontological structuring of logistics services. In *Proceedings of the international conference on web intelligence* (pp. 146–153).

Gordon, J. R., Lee, P.-M., & Lucas, H. C. (2005). A resource-based view of competitive advantage at the port of singapore. *The Journal of Strategic Information Systems*, *14*(1), 69–86.

Grefen, P., Hofman, W., Dijkman, R., Veenstra, A., & Peters, S. (2018). An integrated view on the future of logistics and information technology. *arXiv preprint arXiv:1805.12485*.

Groningen, V. (2006). *Performance measurement in barge planning*. Master Thesis, RSM Erasmus University.

Grunow, M., Günther, H.-O., & Lehmann, M. (2006). Strategies for dispatching agvs at automated seaport container terminals. *OR spectrum*, *28*(4), 587–610.

Guan, C., & Liu, R. R. (2009). Container terminal gate appointment system optimization. *Maritime Economics & Logistics*, *11*(4), 378–398.

Guarino, N., Oberle, D., & Staab, S. (2009). What is an ontology? *Handbook on ontologies*, 1–17.

Guimaraes, T. (2007). Comparing expert systems and agent technology for km. *Emerging Information Resources Management and Technologies*, 321–343.

Guo, Q. L., & Zhang, M. (2010). An agent-oriented approach to resolve scheduling optimization in intelligent manufacturing. *Robotics and Computer-Integrated Manufacturing*, *26*(1), 39 - 45.

Gutierrez, S. A., & Branch, J. W. (2011). A comparison between expert systems and autonomic computing plus mobile agent approaches for fault management. *Dyna*, *78*(168), 173–180.

Hadzic, M., Chang, E., Wongthongtham, P., & Dillon, T. (2009). *Ontology-based multi-agent systems*.

Hansen, I. (2004). Automated shunting of rail container wagons in ports and terminal areas. *Transportation Planning and Technology*, *27*(5), 385–401.

Harris, I., Mumford, C. L., & Naim, M. M. (2014). A hybrid multi-objective approach to capacitated facility location with flexible store allocation for green logistics modeling. *Transportation Research Part E*, *66*, 1–22.

Harris, I., Wang, Y., & Wang, H. (2015). ICT in multimodal transport and technological trends: Unleashing potential for the future. *International Journal of Production Economics*, *159*, 88–103.

Hartmann, S. (2004). Generating scenarios for simulation and optimization of container terminal logistics. *OR Spectrum*, *26*(2), 171–192.

He, D., Pang, Y., & Lodewijks, G. (2016). Speed control of belt conveyors during transient operation. *Powder Technology*, *301*, 622–631.

Heilig, L., Lalla Ruiz, E., & Voß, S. (2016). Port-io: a mobile cloud platform supporting context-aware inter-terminal truck routing. *Proceedings of the 24th European Conference on Information Systems*.

Heilig, L., & Voß, S. (2016). Information systems in seaports: a categorization and overview. *Information Technology and Management*, 1–23.

Hendriks, M. M. (2009). *Multi-step optimization of logistics networks:strategic, tactical, and operational decisions*. PHD thesis, Technology Unviersity of Einhoven.

Henesey, L., Davidsson, P., & Persson, J. A. (2006). Agent based simulation architecture for evaluating operational policies in transshipping containers. *German Conference on Multiagent System Technologies*, 73–85.

Hernández, J. E., Lyons, A. C., Mula, J., Poler, R., & Ismail, H. (2014). Supporting the collaborative decision-making process in an automotive supply chain with a multi-agent system. *Production Planning & Control*, *25*(8), 662–678.

Herz, N., & Flämig, H. (2014). Understanding supply chain management concepts in the context of port logistics: an explanatory framework. *Transport*, *29*(4), 376–385.

Horling, B., & Lesser, V. (2004). A survey of multi-agent organizational paradigms. *The Knowledge Engineering Review*, *19*(04), 281–316.

Hu, D., & Yao, Z. (2012). Stacker-reclaimer scheduling in a dry bulk terminal. *International Journal of Computer Integrated Manufacturing*, *25*(11), 1047–1058.

Huang, C. J., & Liao, L. M. (2012). A multi-agent-based negotiation approach for parallel machine scheduling with multi-objectives in an electro-etching process. *International Journal of Production Research*, *50*(20), 5719–5733.

Introna, L. D. (1991). The impact of information technology on logistics. *International*

Journal of Physical Distribution & Logistics Management, *21*(5), 32–37.

Jeinsch, T., Sader, M., Noack, R., Barber, K., Ding, S., Zang, P., & Zhong, M. (2002). A robust model-based information system for monitoring and fault detection of large scale belt conveyor systems. , *4*, 3283–3287.

Jennings, N. R. (2000). On agent-based software engineering. *Artificial intelligence*, *117*(2), 277–296.

Jourdan, L., Basseur, M., & Talbi, E. (2009). Hybridizing exact methods and metaheuristics: A taxonomy. *European Journal of Operational Research*, *199*(3), 620–629.

Junghans, L., & Darde, N. (2015). Hybrid single objective genetic algorithm coupled with the simulated annealing optimization method for building optimization. *Energy and Buildings*, *86*, 651–662.

Kadir, E. A., Rosa, S. L., & Gunawan, H. (2016). Application of rfid technology and e-seal in container terminal process. *International Conference on Information and Communication Technology (ICoICT), 2016 4th*, 1–6.

Kalogeraki, E.-M., Apostolou, D., Polemi, N., & Papastergiou, S. (2018). Knowledge management methodology for identifying threats in maritime/logistics supply chains. *Knowledge Management Research & Practice*, *16*(4), 508–524.

Kantamneni, A., Brown, L. E., Parker, G., & Weaver, W. W. (2015). Survey of multi-agent systems for microgrid control. *Engineering Applications of Artificial Intelligence*, *45*, 192–203.

Kao, S. L., Lee, K. T., Chang, K. Y., & Ko, M. D. (2007). A fuzzy logic method for collision avoidance in vessel traffic service. *Journal of Navigation*, *60*(01), 17–31.

Keceli, Y., Choi, H. R., Cha, Y. S., & Aydogdu, Y. V. (2008). A study on adoption of port community systems according to organization size. *Third International Conference on Convergence and Hybrid Information Technology, 2008. ICCIT'08.*, *1*, 493–501.

Khozium, M. O., Abuarafah, A. G., & Mohammed, H. A. (2013). Agent vs object with an in-depth insight to multi-agent systems. *International Journal of Advanced Research in Computer Science*, *4*(8).

Kia, M., Shayan, E., & Ghotb, F. (2000). The importance of information technology in port terminal operations. *International Journal of Physical Distribution & Logistics Management*, *30*(3/4), 331–344.

Kim, K., Kim, H., Kim, S. K., & Jung, J. Y. (2015). i-RM: An intelligent risk management framework for context-aware ubiquitous cold chain logistics. *Expert Systems with Applications*, *46*, 463–473.

Kim, K. H., & Hong, G. P. (2006). A heuristic rule for relocating blocks. *Computers & Operations Research*, *33*(4), 940–954.

Kim, K. H., & Lee, H. (2015). Container terminal operation: current trends and future challenges. *Handbook of Ocean Container Transport Logistics*, 43–73.

Kim, Y., Kim, J., Wang, Y., Chang, K., Park, J. W., & Lim, Y. (2009). Application scenarios of nautical ad-hoc network for maritime communications. *OCEANS*

2009, 1–4.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671–680.

Konings, R. (2007). Opportunities to improve container barge handling in the port of Rotterdam from a transport network perspective. *Journal of Transport Geography*, *15*(6), 443–454.

Krummenacher, R., & Strang, T. (2007). Ontology-Based Context Modeling. *Transactions On Information And Systems*(8), 1262–1270.

Kumar, U., Ahmadi, A., Verma, A. K., & Varde, P. (2015). *Current trends in reliability, availability, maintainability and safety: An industry perspective*. Springer.

Lai, K. h., Wong, C. W., & Cheng, T. E. (2006). Institutional isomorphism and the adoption of information technology for supply chain management. *Computers in Industry*, *57*(1), 93–98.

Lang, N., Moonen, H., Srour, J., & Zuidwijk, R. (2008). Multi agent systems in logistics: A literature and state-of-the-art review. *ERIM Report Series Research in Management*(ERS-2008-043-LIS).

Langley Jr, C. J. (1985). Information-based decision making in logistics management. *International Journal of Physical Distribution & Materials Management*, *15*(7), 41–55.

Lee, D.-H., Wang, H. Q., & Miao, L. (2008). Quay crane scheduling with non-interference constraints in port container terminals. *Transportation Research Part E: Logistics and Transportation Review*, *44*(1), 124–135.

Leech, J. (2010). Design of an efficient coal export terminal. *Proceedings of the Queensland Mining and Engineering exhibition*.

Lenstra, J. K., & Kan, A. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, *11*(2), 221–227.

Lesser, V. R. (1999). Cooperative multiagent systems: A personal view of the state of the art. *IEEE Transactions on knowledge and data engineering*, *11*(1), 133–142.

Leung, C., Wong, T., Mak, K., & Fung, R. (2010). Integrated process planning and scheduling by an agent-based ant colony optimization. *Computers and Industrial Engineering*, *59*(1), 166 - 180.

Li, Q., & Fan, H. (2011). 18. a simulation model for detecting vessel conflicts within a seaport. *Navigational Systems and Simulators: Marine Navigation and Safety of Sea Transportation*, 133.

Li, S. (2016). *Coordinated planning of inland vessels for large seaports*. TRAIL Thesis Series, The Netherlands TRAIL Research School.

Li, S., Negenborn, R. R., & Liu, J. (2018). Stimulating inland waterway transport between seaports and the hinterland from a coordination perspective. *International Conference on Computer Languages*, 67–85.

Li, S., Negenborn, R. R., & Lodewijks, G. (2014). A distributed constraint optimization approach for vessel rotation planning. *Computational Logistics*, 61–80.

Li, S., Negenborn, R. R., & Lodewijks, G. (2016a). Approach integrating mixed-integer programming and constraint programming for planning rotations of inland vessels in a large seaport. *Transportation Research Record: Journal of the Transportation Research Board*(2549), 1–8.

Li, S., Negenborn, R. R., & Lodewijks, G. (2016b). Distributed constraint optimization for addressing vessel rotation planning problems. *Engineering Applications of Artificial Intelligence*, *48*, 159–172.

Li, S., Negenborn, R. R., & Lodewijks, G. (2017). Planning inland vessel operations in large seaports using a two-phase approach. *Computers & Industrial Engineering*, *106*, 41–57.

Li, W., Wu, Y., Petering, M. E., Goh, M., & de Souza, R. (2009). Discrete time model and algorithms for container yard crane scheduling. *European Journal of Operational Research*, *198*(1), 165–172.

Lin, S.-W., & Ting, C.-J. (2014). Solving the dynamic berth allocation problem by simulated annealing. *Engineering Optimization*, *46*(3), 308–327.

Liu, D., & Stewart, T. J. (2004). Object-oriented decision support system modelling for multicriteria decision making in natural resource management. *Computers & Operations Research*, *31*(7), 985–999.

Liu, L., Logan, K. P., Cartes, D., Srivastava, S. K., et al. (2007). Fault detection, diagnostics, and prognostics: software agent solutions. *IEEE Transactions on Vehicular Technology,*, *56*(4), 1613–1622.

Liu, N., Abdelrahman, M. A., & Ramaswamy, S. (2007). A complete multiagent framework for robust and adaptable dynamic job shop scheduling. *IEEE Transactions On Systems Man And Cybernetics Part C Applications And Reviews*, *37*(5), 904.

Liu, X., Pang, Y., & Lodewijks, G. (2016). Theoretical and experimental determination of the pressure distribution on a loaded conveyor belt. *Measurement*, *77*, 307–316.

Liu, Y., Ji, Y., & Jiao, R. J. (2013). A Stackelberg solution to joint optimization problems: A case study of green design. *Procedia Computer Science*, *16*, 333–342.

Lodewijks, G. (2005). Large-scale conveyor belt system maintenance decision-making by using fuzzy causal modeling. *IEEE Intelligent Transportation Systems*, 402–406.

Lodewijks, G., Li, W., Pang, Y., & Jiang, X. (2016). An application of the iot in belt conveyor systems. *International Conference on Internet and Distributed Computing Systems*, 340–351.

Lodewijks, G., & Ottjes, J. (2004). optimisation of maintenance strategies for large scale transport systems lodewijks. *International Materials Handling Conference (Beltcon)*.

Lodewijks, G., & Ottjes, J. (2005a). Application of Fuzzy Logic in belt conveyor monitoring and control. *International Materials Handling Conference (Beltcon) 13*, 1–13.

Lodewijks, G., & Ottjes, J. A. (2005b). Intelligent belt conveyor monitoring & control: Theory and applications. *proceedings of the BeltCon 13 conference*.

Lodewijks, G., Schott, D. L., Ottjes, J. a., & Engineering, T. (2009). Dry bulk terminal expansion or redesign? *Port Technology International*, *1*(43), pp. 87–94.

Lun, Y. V., Wong, C. W., Lai, K.-H., & Cheng, T. (2008). Institutional perspective on the adoption of technology for the security enhancement of container transport. *Transport Reviews*, *28*(1), 21–33.

Madavan, N. K. (2002). Multiobjective optimization using a pareto differential evolution approach. *Proceedings of the 2002 Congress on Evolutionary Computation, 2002. CEC'02.*, *2*, 1145–1150.

Mahdavi, I., Shirazi, B., Ghorbani, N., & Sahebjamnia, N. (2013). IMAQCS: Design and implementation of an intelligent multi-agent system for monitoring and controlling quality of cement production processes. *Computers in Industry*, *64*(3), 290–298.

Manoufali, M., Alshaer, H., Kong, P. Y., & Jimaa, S. (2013). Technologies and networks supporting maritime wireless mesh communications. *Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP*, 1–8.

Marik, V., & McFarlane, D. (2005). Industrial adoption of agent-based technologies. *IEEE Intelligent Systems*, *20*(1), 27–35.

Maturana, F. P., & Norrie, D. H. (1996). Multi-agent mediator architecture for distributed manufacturing. *Journal of Intelligent Manufacturing*, *7*(4), 257–270.

McArthur, S. D., Booth, C. D., McDonald, J., & McFadyen, I. T. (2005). An agent-based anomaly detection architecture for condition monitoring. *IEEE Transactions on Power Systems*, *20*(4), 1675–1682.

Meisel, F., & Bierwirth, C. (2013). A framework for integrated berth allocation and crane operations planning in seaport container terminals. *Transportation Science*, *47*(2), 131–147.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, *21*(6), 1087–1092.

Milà, S. G. (2007). Information technologies in ports: Port community systems. *IAPS Conference*.

Min, H., Lim, Y. K., & Park, J. W. (2013). X-ray cargo inspection for enhancing the maritime safety and security: a case study of international container ports. *International Journal of Logistics Systems and Management*, *16*(1), 98–112.

Mokhtari, K., Ren, J., Roberts, C., & Wang, J. (2012). Decision support framework for risk management on sea ports and terminals using fuzzy set theory and evidential reasoning approach. *Expert Systems with Applications*, *39*(5), 5087–5103.

Moonen, H. (2009). *Multi-agent systems for transportation planning and coordination*. RIM Ph.D. Series Research in Management. Erasmus Research Institute of Management.

Moonen, H., Van de Rakt, B., Miller, I., Van Nunen, J., & Van Hillegersberg, J. (2005). Agent technology supports inter-organizational planning in the port. *ERIM Report Series Reference No*.

Moorthy, R., & Teo, C.-P. (2006). Berth management in container terminal: the template design problem. *OR spectrum*, *28*(4), 495–518.

Murty, K. G., Liu, J., Wan, Y.-w., & Linn, R. (2005). A decision support system for operations in a container terminal. *Decision support systems*, *39*(3), 309–332.

Murty, K. G., Wan, Y. W., Liu, J., Tseng, M. M., Leung, E., Lai, K.-K., & Chiu, H. W. (2005). Hongkong international terminals gains elastic capacity using a data-intensive decision-support system. *Interfaces*, *35*(1), 61–75.

Naderi, B., Zandieh, M., & Roshanaei, V. (2009). Scheduling hybrid flowshops with sequence dependent setup times to minimize makespan and maximum tardiness. *International Journal of Advanced Manufacturing Technology*, *41*(11-12), 1186–1198.

Nadoveza, D., & Kiritsis, D. (2014). Ontology-based approach for context modeling in enterprise applications. *Computers in Industry*, *65*(9), 1218–1231.

Narayanaswami, S., & Rangaraj, N. (2015). A MAS architecture for dynamic, realtime rescheduling and learning applied to railway transportation. *Expert Systems with Applications*, *42*(5), 2638–2656.

Natarajan, S., Ghosh, K., & Srinivasan, R. (2012). An ontology for distributed process supervision of large-scale chemical plants. *Computers & Chemical Engineering*, *46*, 124–140.

Ng, W., & Mak, K. (2004). Sequencing of container pickup trucks in container yard blocks. *International Journal of Industrial Engineering: Theory Applications and Practice*, *11*(1), 82-89.

Ng, W., & Mak, K. (2005). Yard crane scheduling in port container terminals. *Applied mathematical modelling*, *29*(3), 263–276.

Ngai, E., Li, C. L., Cheng, T., Lun, Y. V., Lai, K. H., Cao, J., & Lee, M. (2011). Design and development of an intelligent context-aware decision support system for real-time monitoring of container terminal operations. *International Journal of Production Research*, *49*(12), 3501–3526.

Ngai, E., Moon, K. K., Riggins, F. J., & Candace, Y. Y. (2008). Rfid research: An academic literature review (1995–2005) and future research directions. *International Journal of Production Economics*, *112*(2), 510–520.

Notteboom, T. E. (1997). Concentration and load centre development in the european container port system. *Journal of transport geography*, *5*(2), 99–115.

Nozick, L. K., & Morlok, E. K. (1997). A model for medium-term operations planning in an intermodal rail-truck service. *Transportation research part a: policy and practice*, *31*(2), 91–107.

Nwana, H., & Ndumu, D. (1997). *An introduction to agent technology, software agents and soft computing, hs nwana and n. azarmi*. Springer-Verlag, Berlin, Germany.

Nwana, H. S., Ndumu, D. T., Lee, L. C., & Collis, J. C. (1999). Zeus: A toolkit and approach for building distributed multi-agent systems. *Proceedings of the Third Annual Conference on Autonomous Agents*, 360–361.

O'Brien, P. D., & Nicol, R. C. (1998). Fipatowards a standard for software agents. *BT Technology Journal*, *16*(3), 51–59.

Ohazulike, A. E., & Brands, T. (2013). Multi-objective optimization of traffic externalities using tolls: A comparison of genetic algorithm and game theoretical approach. *2013 IEEE Congress on Evolutionary Computation, CEC 2013*, 2465–2472.

Osman, I. H., & Laporte, G. (1996). Metaheuristics: A bibliography. *Annals of Operations research*, *63*(5), 511–623.

Pang, Y. (2010). *Intelligent belt conveyor monitoring and control*. TU Delft, Delft University of Technology.

Pang, Y., & Lodewijks, G. (2005, Sept). Large-scale conveyor belt system maintenance decision-making by using fuzzy causal modeling. *Proceedings of IEEE Intelligent Transportation Systems, 2005*, 563-567.

Pang, Y., & Lodewijks, G. (2011). The application of rfid technology in large-scale dry bulk material transport system monitoring. *Environmental Energy and Structural Monitoring Systems (EESMS), 2011 IEEE Workshop on*, 1–5.

Pang, Y., & Lodewijks, G. (2012). Agent-based intelligent monitoring in large-scale continuous material transport. *Proceedings of 2012 9th IEEE International Conference on Networking, Sensing and Control*, 79–84.

Pang, Y., & Lodewijks, G. (2013). A Remote Intelligent Belt Conveyor Inspection Tool. *11th Int. congress bulk materials storage, handling and transportation*(July).

Panzarasa, P., Jennings, N. R., & Norman, T. J. (2002). Formalizing collaborative decisionmaking and practical reasoning in multiagent systems. *Journal of Logic and Computation*, *12*(1), 55-117.

Park, N., & Bang, H. C. (2016). Mobile middleware platform for secure vessel traffic system in iot service environment. *Security and Communication Networks*, *9*(6), 500–512.

Perego, A., Perotti, S., & Mangiaracina, R. (2011). ICT for logistics and freight transportation: a literature review and research agenda. *International Journal of Physical Distribution & Logistics Management*, *41*(5), 457–483.

Phan, M. H., & Kim, K. H. (2015). Negotiating truck arrival times among trucking companies and a container terminal. *Transportation Research Part E: Logistics and Transportation Review*, *75*, 132–144.

Pielage, B. J., Konings, R., Rijsenbrij, J., & Schuylenburg, M. v. (2007). Barge hub terminals: a perspective for more efficient hinterland container transport for the port rotterdam. *48th Annual Transportation Research Forum, Boston University, Boston, MA*, 19–27.

Pistofidis, P., & Emmanouilidis, C. (2012). Profiling context awareness in mobile and cloud based engineering asset management. *IFIP International Conference on Advances in Production Management Systems*, 17–24.

Posti, A., Häkkinen, J., & Tapaninen, U. (2011). Promoting information exchange with a port community system–case finland. *International Supply Chain Management and Collaboration Practices*, *4*, 455–471.

Ramos, L. (2015). Semantic Web for manufacturing, trends and open issues: Toward a state of the art. *Computers and Industrial Engineering*, *90*, 444–460.

Ramot, D., Friedman, M., Langholz, G., & Kandel, A. (2003). Complex fuzzy logic. *IEEE Transactions on Fuzzy Systems*, *11*(4), 450–461.

Rani, M., Nayak, R., & Vyas, O. P. (2015). An ontology-based adaptive personalized e-learning system, assisted by software agents on cloud storage. *Knowledge-Based Systems*, *90*, 33–48.

Ruiz, R., & Maroto, C. (2005). A comprehensive review and evaluation of permutation flowshop heuristics. *European Journal of Operational Research*, *165*(2), 479–494.

Russell, S. J., Norvig, P., Canny, J. F., Malik, J. M., & Edwards, D. D. (2003). *Artificial intelligence: a modern approach* (Vol. 2). Prentice hall Upper Saddle River.

Sarrafha, K., Rahmati, S. H. A., Niaki, S. T. A., & Zaretalab, A. (2014). A bi-objective integrated procurement, production, and distribution problem of a multi-echelon supply chain network design: A new tuned MOEA. *Computers and Operations Research*, *54*, 35–51.

Schneeweiss, C. (2012). *Distributed decision making*. Springer Science & Business Media.

Schut, M., Kentrop, M., Leenaarts, M., Melis, M., & Miller, I. (2004). Approach: Decentralised rotation planning for container barges. *Proceedings of the 16th European Conference on Artificial Intelligence*, 755–759.

Sels, V., & Vanhoucke, M. (2012). A hybrid genetic algorithm for the single machine maximum lateness problem with release times and family setups. *Computers and Operations Research*, *39*(10), 2346–2358.

Shen, W., & Norrie, D. (Nov,1997). Facilitators , Mediators or Autonomous Agents. *Proceeding of the Second International Workshop on CSCW in design, Bangbok, Thailand*(1), 119-124.

Shi, X., Tao, D., & Voß, S. (2011). Rfid technology and its application to port-based container logistics. *Journal of Organizational Computing and Electronic Commerce*, *21*(4), 332–347.

Simsir, U., Amasyalı, M. F., Bal, M., Çelebi, U. B., & Ertugrul, S. (2014). Decision support system for collision avoidance of vessels. *Applied Soft Computing*, *25*, 369–378.

Siror, J. K., Huanye, S., & Dong, W. (2011). Rfid based model for an intelligent port. *Computers in industry*, *62*(8), 795–810.

Snyder, R. D., Douglas, D., & Mackenzie, C. (2004). Cougaar agent communities. *Proceedings of Open Cougaar*.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, *35*(2), 254–265.

Sprenger, R., & Mönch, L. (2014, sep). A decision support system for cooperative transportation planning: Design, implementation, and performance assessment. *Expert Systems with Applications*, *41*(11), 5125–5138.

Srinivas, N., & Deb, K. (1994). Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, *2*(3), 221–248.

Srour, F. J., van Oosterhout, M., van Baalen, P., & Zuidwijk, R. (2008). Port community system implementation: Lessons learned from international scan. *Transportation Research Board 87th Annual Meeting, Washington DC*.

Staab, S., & Studer, R. (2010). *Handbook on ontologies*. Springer Science & Business Media.

Stahlbock, R., & Voß, S. (2008). Operations research at container terminals: A literature update. *OR Spectrum*, *30*(1), 1–52.

Stallings, W., & Manna, M. M. (1997). *Data and computer communications* (Vol. 6). Prentice hall Englewood Cliffs, NJ.

Steenken, D., Voß, S., & Stahlbock, R. (2004). Container terminal operation and operations research-a classification and literature review. *OR spectrum*, *26*(1), 3–49.

Tapia, D. I., Bajo, J., & Corchado, J. M. (2009). Distributing functionalities in a soa-based multi-agent architecture. *7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009)*, 20–29.

Terroso-Saenz, F., Valdes-Vela, M., Campuzano, F., Botia, J. A., & Skarmeta-Gomez, A. F. (2015). A complex event processing approach to perceive the vehicular context. *Information Fusion*, *21*, 187–209.

Tseng, Y., Yue, W., & Taylor, M. (2005, 01). The role of transportation in logistics chain. *Proceedings of the Eastern Asia Society for Transportation Studies*, *5*, 1657-1672.

Tu, E., Zhang, G., Rachmawati, L., Rajabally, E., & Huang, G. B. (2017). Exploiting ais data for intelligent maritime navigation: A comprehensive survey from data to methodology. *IEEE Transactions on Intelligent Transportation Systems*, 1-24.

UNCTAD. (1985). *Port development: A handbook for planners in developing countries*. United Nations Conference on Trade and Development. New York: UNITED NATIONS.

UNCTAD. (1990). The management of port equipment maintenance. *Infrastructure and Urban Development Department technical paper*.

UNCTAD. (2006). *Report of the expert meeting on ict solutions to facilitate trade at border crossings and ports*. Geneva: UNCTAD/RMT/2006.

UNCTAD. (2014). *Review of maritime transport 2013*. Geneva: UNCTAD/RMT/2013.

Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, methods and applications. *The knowledge engineering review*, *11*(2), 93–136.

Van Der Horst, M. R., & De Langen, P. W. (2008). Coordination in Hinterland Transport Chains: A Major Challenge for the Seaport Community. *Maritime Economics & Logistics*, *10*(1-2), 108–129.

Van Dam, K. H., Verwater-Lukszo, Z., Ottjes, J. A., & Lodewijks, G. (2006). Distributed intelligence in autonomous multi-vehicle systems. *International journal of critical infrastructures*, *2*(2-3), 261–272.

Van der Horst, M., & Van der Lugt, L. (2009). Coordination in railway hinterland chains: an institutional analysis. *Internationalization of Infrastructures: Pro-*

ceedings of the 12th Annual International Conference on the Economics of Infrastructures, Delft, The Netherlands.

Van der Horst, M. R., & Van der Lugt, L. M. (2014). An Institutional Analysis of Coordination in Liberalized Port-related Railway Chains: An Application to the Port of Rotterdam. *Transport Reviews*, *34*(1), 68–85.

Van Rijswijck, H., & Davydenko, I. (2007). *Intelligent agents in transportation planning*. TNO.

Van Vianen, T. A., Ottjes, J. A., Negenborn, R. R., Lodewijks, G., & Mooijman, D. L. (2012). Simulation-based operational control of a dry bulk terminal. , 73–78.

Van Zijverden, R. J., & Negenborn, R. R. (2012). Survey of approaches for integrated control of intermodal container terminals. *Proccedings of 2012 9th IEEE International Conference on Networking, Sensing and Control (ICNSC)*, 67–72.

Vieira, G. B. B., Pasa, G. S., do Oliveira Borsa, M. B. N., Milan, G. S., & Pandolfo, A. (2011). Materials handling management: A case study. *Journal of Operations and Supply Chain Management*, *4*(2), 19–30.

Villegas, J. G., Palacios, F., & Medaglia, A. L. (2006). Solution methods for the bi-objective (cost-coverage) unconstrained facility location problem with an illustrative example. *Annals of Operations Research*, *147*(1), 109–141.

Vis, I. F., & Roodbergen, K. J. (2009). Scheduling of container storage and retrieval. *Operations Research*, *57*(2), 456–467.

Visser, J., Konings, R., Pielage, B.-J., & Wiegmans, B. (2007). A new hinterland transport concept for the port of rotterdam: organisational and/or technological challenges. *Proceedings of the Transportation Research Forum, North Dakota State University*.

Von Stackelberg, H. (1952). *The theory of the market economy*. London, England: William Hodge.

Wang, G. (2012). A wireless sensor network in port security. *2012 International Conference on System Science and Engineering (ICSSE)*, 199–204.

Wang, G., Wong, T. N., & Wang, X. (2014). A hybrid multi-agent negotiation protocol supporting agent mobility in virtual enterprises. *Information Sciences*, *282*, 1–14.

Weiss, G. (1999). *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT press.

Weyns, D., & Georgeff, M. (2010). Self-adaptation using multiagent systems. *IEEE Software*, *27*(1), 86-91.

Xie, L., Chu, X., Huang, M., & Yan, Z. (2011). Vessel traffic flow detection and tracking in river. *Proceedings of the First International Conference on Transportation Information and Safety, 2011, Wuhan, China*.

Xin, J., Negenborn, R. R., Corman, F., & Lodewijks, G. (2015). Control of interacting machines in automated container terminals using a sequential planning approach for collision avoidance. *Transportation Research Part C: Emerging Technologies*, *60*, 377–396.

Xin, J., Negenborn, R. R., & Lodewijks, G. (2014). Energy-aware control for automated container terminals using integrated flow shop scheduling and optimal control. *Transportation Research Part C: Emerging Technologies*, *44*, 214–230.

Xu, G., Shen, W., & Wang, X. (2014). Applications of wireless sensor networks in marine environment monitoring: A survey. *Sensors*, *14*(9), 16932–16954.

Yılmaz, Ö., & Erdur, R. C. (2012). iConAwa–an intelligent context-aware system. *Expert Systems with Applications*, *39*(3), 2907–2918.

Yin, X. F., Khoo, L. P., & Chen, C.-H. (2011). A distributed agent system for port planning and scheduling. *Advanced Engineering Informatics*, *25*(3), 403–412.

Youn, S., & McLeod, D. (2006). Ontology development tools for ontology-based knowledge management. *Encyclopedia of E-Commerce, E-Government, and Mobile Commerce*, 858–864.

Yu, R., Iung, B., & Panetto, H. (2003). A multi-agents based e-maintenance system with case-based reasoning decision support. *Engineering applications of artificial intelligence*, *16*(4), 321–333.

Zachman, J. A. (1987). A framework for information systems architecture. *IBM systems journal*, *26*(3), 276–292.

Zhang, L., Gao, L., & Li, X. (2013). A hybrid genetic algorithm and tabu search for a multi-objective dynamic job shop scheduling problem. *International Journal of Production Research*, *51*(12), 3516–3531.

Zhang, Y., Huang, G. Q., Sun, S., & Yang, T. (2014). Multi-agent based real-time production scheduling method for radio frequency identification enabled ubiquitous shopfloor environment. *Computers & Industrial Engineering*, *76*, 89–97.

Zheng, H., Negenborn, R. R., & Lodewijks, G. (2017). Closed-loop scheduling and control of waterborne agvs for energy-efficient inter terminal transport. *Transportation Research Part E: Logistics and Transportation Review*, *105*, 261–278.

Zio, E., & Bazzo, R. (2012). A comparison of methods for selecting preferred solutions in multiobjective decision making. *Computational intelligence systems in industrial engineering*, 23–43.

Zuo, X., Chen, C., Tan, W., & Zhou, M. (2015). Vehicle scheduling of an urban bus line via an improved multiobjective genetic algorithm. *IEEE Transactions on Intelligent Transportation Systems*, *16*(2), 1030–1041.

Appendix A

The basis of a fuzzy-logic method

The basis of fuzzy-logic method

A unique feature of a fuzzy-logic method is its ability to transform a subjective or linguistic knowledge into a mathematics model (Ramot et al., 2003). A set of rules is defined to transform the input data to a scalar output. The basic flow of the fuzzy-logic method is illustrated in Figure A.1.

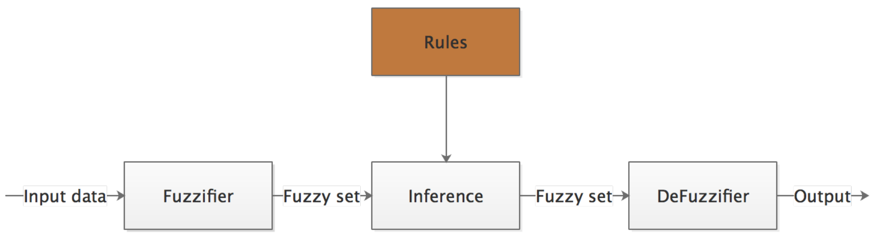

Figure A.1: A fuzzy logic method framework

A set of input data is transformed to a fuzzy set by applying a mapping function. This process is also called fuzzification and the mapping function is called fuzzifier. Afterward, a set of rules is applied to the fuzzy set. Finally, the fuzzy set is converted to a set of output in a defuzzification process. Given a set of input X, the fuzzy set A of X is given in Equation A.1:

$$A : X \Rightarrow [0,1] \tag{A.1}$$

The $A(x)$ is defined as the membership degree of input element x in fuzzy set A for each $x \in X$. The value 0 indicates a non-membership and value 1 indicates a complete membership. The value between 0 and 1 indicates the intermediate degree of membership.

To illustrate the application of the fuzzy logic method, a simple example is given. Suppose the condition of a tear on the belt surface is measured. Based on the measurement, the condition needs to be evaluated and the decision is needed for maintenance or replacement based on the condition. A simple application of fuzzy logic can be used to mapping the input data to the scale of [0,1] where 0 indicates a non-relevant tear shape and 1 indicates a server crack that needs immediate actions. The complete membership function can be roughly given in Figure A.2

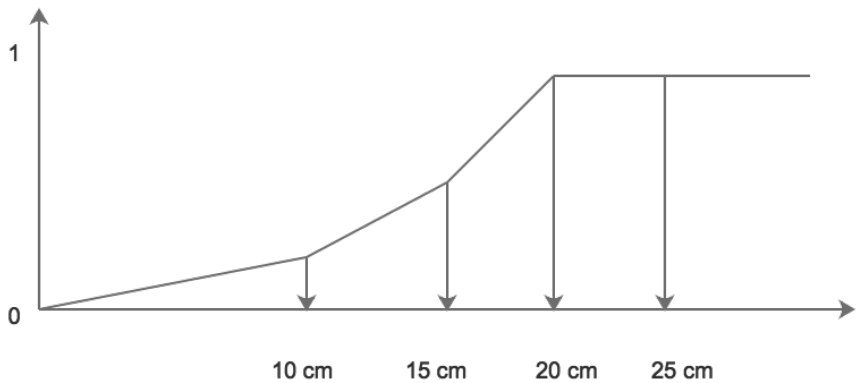

Figure A.2: An simple membership function of belt tear condition

Belt wear index

Belt wear index is a reliability indicator. It is used to quantify the damage level of wear (e.g., a tear on the belt surface) on the belt, and determine the proper maintenance actions: plan a subsequent inspection, immediate repair of the system (e.g., replace the fault belt section). Based on (Lodewijks & Ottjes, 2005a), the belt wear index is determined by three factors: a location factor, an intensity factor, and an extension factor.

- Location Factor (LF): assess the location factor that could contribute to the damage. For instance, the tear found at the edge of the belt is more likely lead to significant damage than the one in the center. The membership function is depicted in the left graph in Figure A.3. The horizontal axis represents the relative position to the center of the belt. It can be seen from the plot that the closer the damage to the center (where x equal to 0.5), the less damage it creates.
- Intensity Factor (IF): represents the type of damage. The membership function of IF is given in Table A.1. It is based on the damage type of the tear on the belt. The more irregular and heavy damage, the higher the degree is given.
- Extension Factor (EF): assess the size of a damage. The EF is determined based on the broken cables inside the belt. The membership function can be seen on

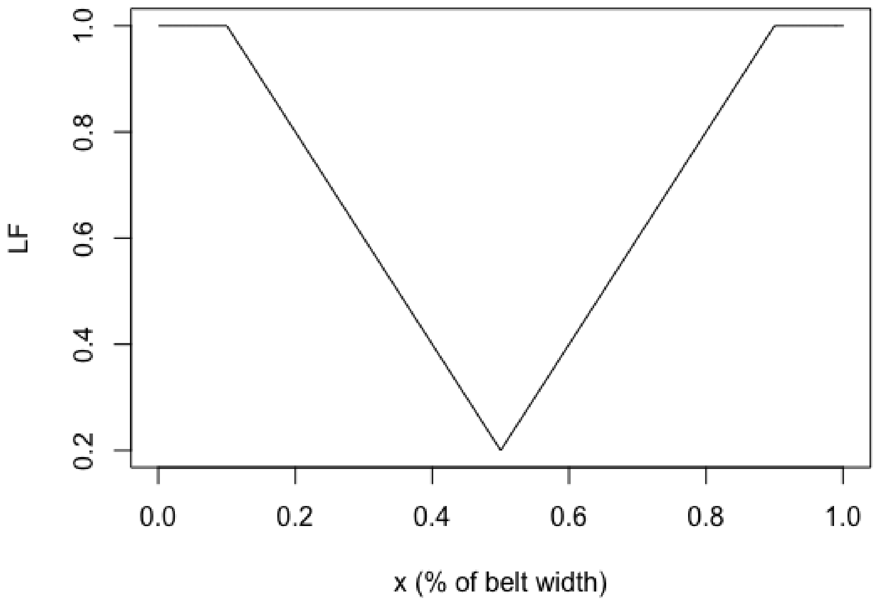

Figure A.3: Membership functions: location factor

Table A.1: Memberships of IF

Type	Slight	Smooth	Irregular	Uneven	Broken
IF	0.2	0.4	0.6	0.8	1

the right plot in Figure A.4. The x represents the percentages of broken cables. If more than 10 % of cables are broken, then the EF is given as 1 which indicate an server damage is developed for the tear condition.

Combine above-defined factors, the belt wear index is given in Equation A.2. The calculation of WI is given in Equation A.2 where DF is the decision factors described previously include EF, IF and LF. The w_i is the weight that applied to each DF_i. According to (Lodewijks, 2005), the weight for each factor is given as: $w_{LF} = 2.0, w_{EF} = 3.0, w_{IF} = 1.0$.

$$WI = \sum_i w_i DF_i / \sum_i w_i \tag{A.2}$$

The calculation of a belt wear index is given. The remaining question is how this value means in terms of maintenance decisions. To achieve this purpose, a defuzzification process is needed. Figure A.5 presents a defuzzification of a belt wear index to a repair index. If repair index is 1.0, an immediate repair is needed. If the value is 0, no immediate repair is needed. From the figure, it can be seen that if the wear index

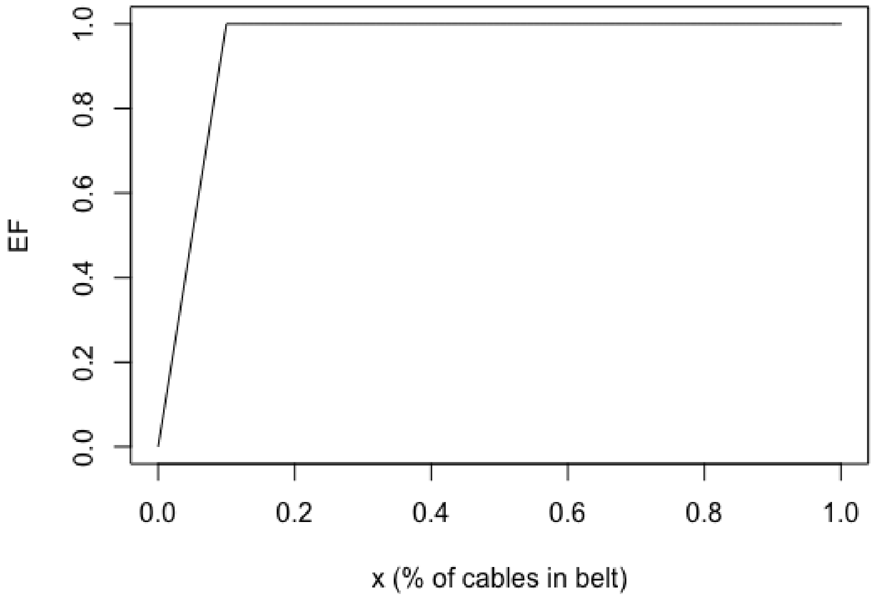

Figure A.4: Membership functions: extension factor

exceeds 0.55, the damage is identified and a repair is needed. If the wear index is below 0.55, no immediate action is required, a further inspection should be planned. To determine the inspection interval, an inspection frequency index is introduced as described in next sections.

Belt inspection frequency index

To determine and adjust the inspection interval, a belt inspection frequency index is introduced. As shown in Figure 6.25, an inspection frequency index is calculated by an accumulation factor. An accumulation factor (AF): quantify the damage propagation pace between two consecutive inspections with respect to all the decision factors. For instance, the accumulation factor of EF can be calculated by Equation A.3 and it also applies to LF and EF.

$$AF_{\{EF,LF,IF\}} = \frac{\{EF,LF,IF\}_{t+\Delta t} - \{EF,LF,IF\}_t}{\Delta t} \tag{A.3}$$

where Δt is the time interval between two consecutive inspections.

The frequency index FI is a function of inspection frequency index IF. The calculation of IF is given in Equation A.4.

$$IF = \sum_i w_i^{'} AF_i / \sum_i w_i^{'} \tag{A.4}$$

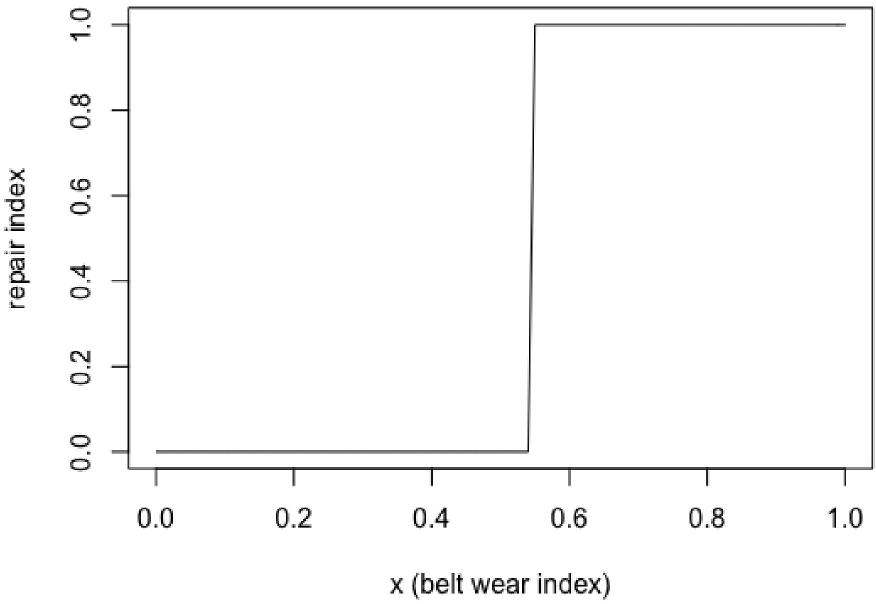

Figure A.5: Defuzzyfication of wear index

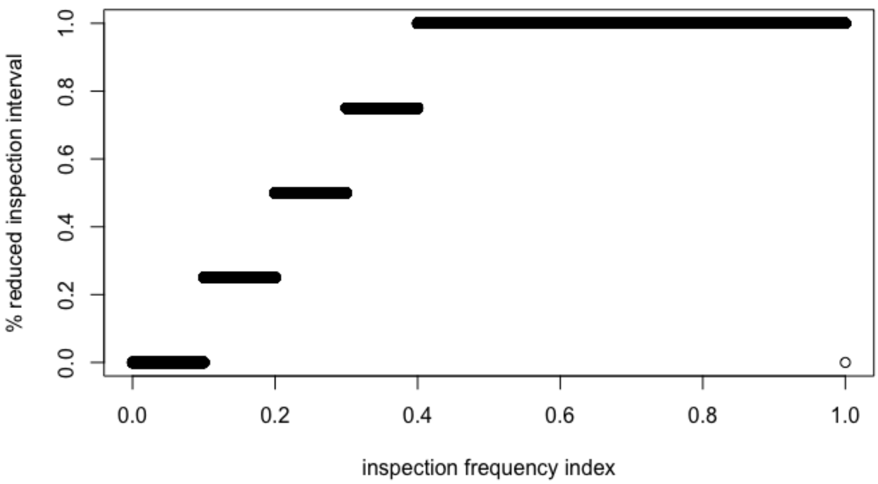

Figure A.6: Defuzzyfication of frequency index

Similar to the belt wear index, the inspection frequency index is a numerical number. To transform the number into a proper decision ,a defuzzyfication is needed. The defuzzification of FI deliver the decisions of optimized strategy for further inspections. The defuzzyfication result is shown in Figure A.6. For example, if $FI = 0.25$, it indicates the inspection interval should be reduced by 12.5 %, similarly if $FI = 1$ indicate the inspection interval should be reduced by 50 %.

Appendix B

System implementation

Prototype implementations for the integrated DM system for HBT planning

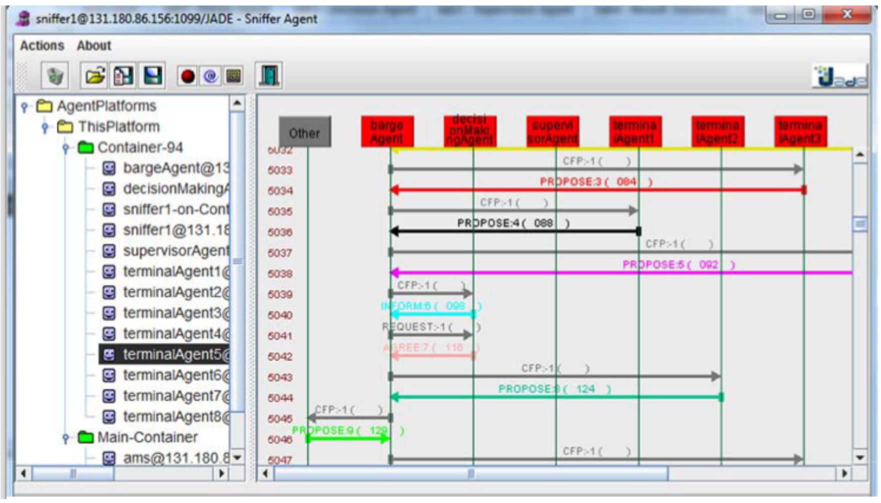

Figure B.1: A demonstration of agent message exchange which includes a barge agent, 8 terminal agents , and the mediator agent. This figure is captured from the JADE agent platform system.

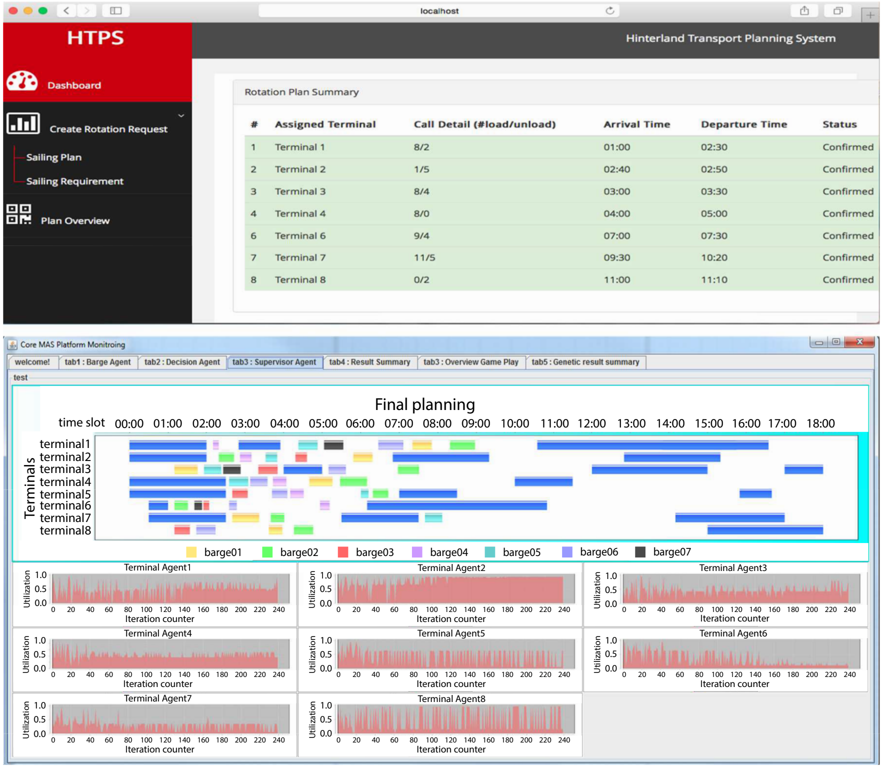

Figure B.2: The web-based user-interface and the backend decision-making thread of the implemented HBT planning DM system. The figures are captured from the developed prototype system.

Prototype implementations for the integrated DM system for BCS reliability assessments

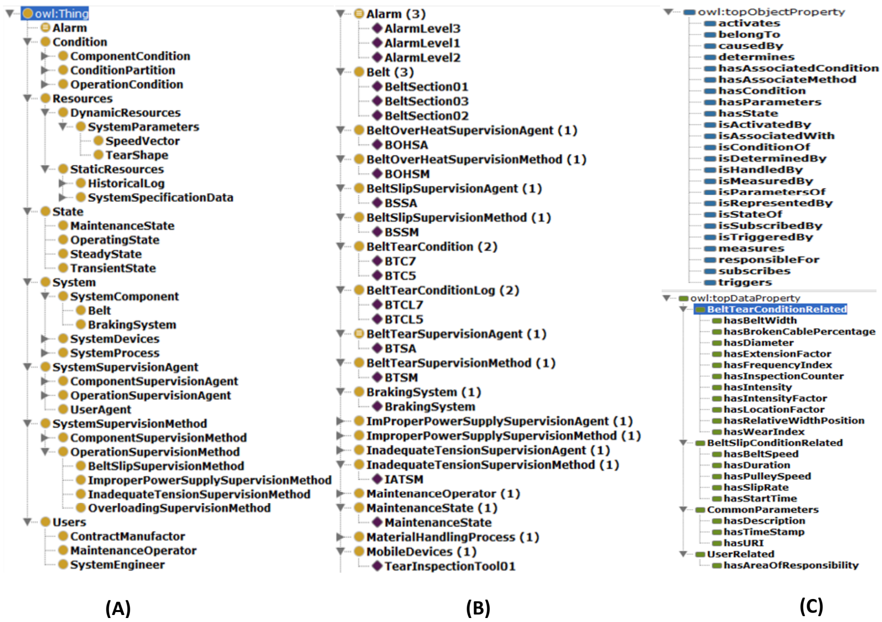

Figure B.3: A demonstration of the developed **ontoSupervision** ontology, the figure is captured from the Protege software platform.

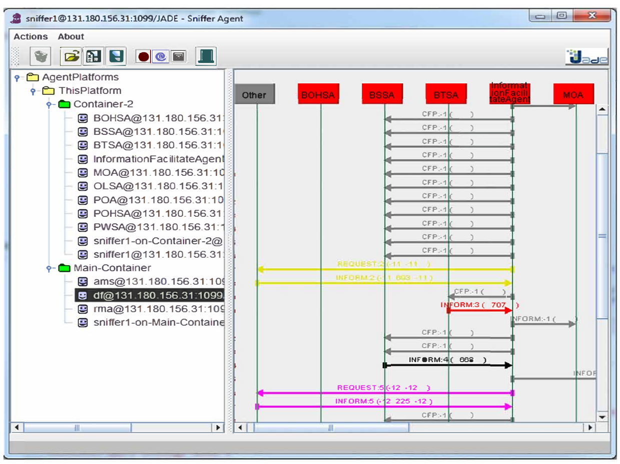

Figure B.4: A demonstration of agent message exchange which includes different types of supervision agent and the Information Facilitate agent. This figure is captured from the JADE agent platform system.

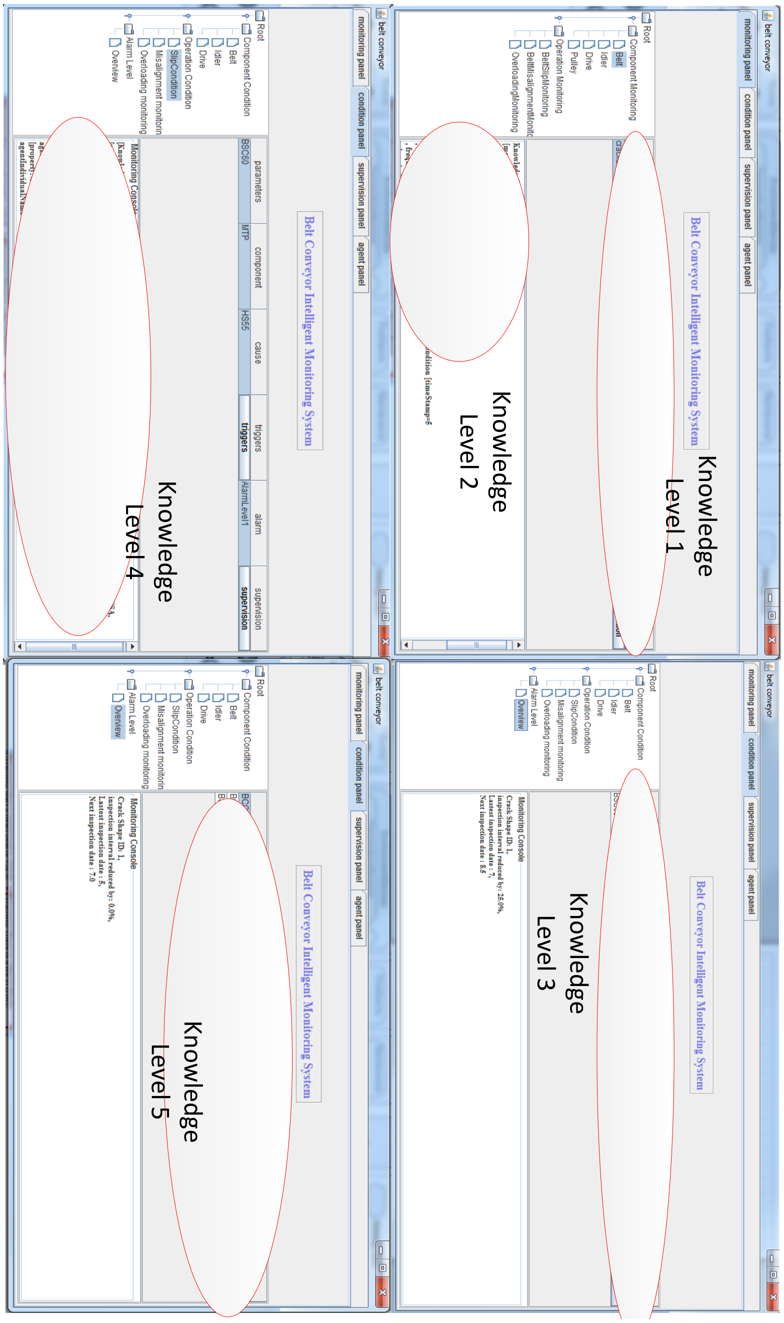

Figure B.5: A demonstration of the developed reliability assessment DM system. Figures from upper right to lower left are: (1) context acquisition and identification by applying KFT1 and KFT2; (2) context supervision employing KFT3; (3) check condition casualty applying KFT4; (4) deliver supervision result to corresponding user.

Glossary

The abbreviations and symbols that are used in this thesis are presented as follows:

Abbreviations

ABA	:	Agent-Based Approach
BA	:	Barge Agent
BCS	:	Belt Conveyor System
CAS	:	Context Aware System
CASS	:	Context Aware Supervision System
CM	:	Collaboration Mechanism
DL	:	Descriptive Language
DM	:	Decision Making
ES	:	Expert System
FFE	:	Fractional Factorial Experiment
HBT	:	Hinterland Barge Transport
ICT	:	Information and Communication Technology
IDM	:	Intelligent Decision-Making
IoT	:	Internet of Things
MAS	:	Multi-Agent System
MHS	:	Material Handling System
MILP	:	Mixed Integer Linear Programming
ML	:	Machine Learning
MOO	:	Multi-Objective Optimisation
NSGA	:	Non Dominate Sorting Genetic Algorithm
OBA	:	Object Based Approach
PL	:	Port Logistics
SA	:	Simulated Annealing
SOA	:	Service Oriented Architecture
TA	:	Terminal Agent
TOS	:	Terminal Operation System
VRPTW	:	Vehicle Routing Problem With Time Windows

Symbols

I	:	a set of barges that are considered in a HBT planning model
N^i	:	a set of terminals that the barge i has be visited
AT^i	:	arrival time of i
DT^i	:	latest departure time of i
$S^i_{q,p}$	:	sailing time of barge i from terminal p to q
L^i_p	:	container load number of i at p
UL^i_p	:	container unload number of i at p
M	:	a very large number
a^i_p	:	operation start time of barge i at terminal q
b^i_p	:	operation end time of barge i at terminal q
∂_i	:	turn around time of the barge i
AR^i_p	:	arrival time of barge i at terminal p
D^i_p	:	departure time of barge i at terminal p
OS^i_p	:	operation start time of barge i at terminal p
σ^i_p	:	binary variable determines if terminal p is the first terminal to visit by barge i
ς^i_p	:	binary variable determine if terminal p is the last terminal to visit by barge i
D_i	:	departure time of barge i at last terminal.
$X^i_{p,q}$	:	binary variable that determine if barge i travel from terminal p to terminal q
P	:	a set of terminals in port region
K	:	number of planned events in single terminal
OR_p	:	operation rate, time taken to load or unload a container
E^p	:	a set of events which have been planned in terminal p
OS^p_k	:	operation start time of event k in terminal p
OF^p_k	:	operation finish time of event k in terminal p
a^i_p	:	operation start time of barge i provided by terminal p
b^i_p	:	operation end time of barge i provided by terminal p
$I^p_{k,k+1}$	:	idle time between event k and $k+1$ in terminal p
A^p_k	:	k_{th} available start time in terminal p
B^p_k	:	k_{th} available end time in terminal p
α_p	:	terminal utilization level of p
β_p	:	buffer time of terminal p
C_p	:	compactness of terminal p
ϕ^i_p	:	waiting time of barge i at terminal p
r	:	signal to noise ratio

Samenvatting

Havenlogistiek (E. port logistics, PL) kan worden gedefinieerd als het proces van planning, implementatie en besturing van de stroom van goederen en informatie via havens van de zee naar het achterland, en omgekeerd. Ten behoeve van PL-systemen moeten functies worden ontwikkeld voor de ondersteuning van activiteiten als zeezijdig en landzijdig transport, opslag van vracht, verwerking van orders en distributie. Door de groeiende vraag en de sterk competitieve markt moeten de prestaties van PL-systemen, zoals de operationele efficintie en de betrouwbaarheid, voortdurend worden verbeterd. Een belangrijk onderdeel is de verbetering van de besluitvorming. Beslissingssystemen zijn een belangrijk onderdeel van PL-systemen, met name bij het nagaan hoe verschillende processen, operaties en equipment kunnen worden bestuurd en gecordineerd. Er zijn met ICT-technieken belangrijke bijdragen geleverd aan de ontwikkeling van beslissingssystemen, maar er is nog niet genoeg ICT-ondersteuning in alle systemen. Het is onduidelijk wat de toegevoegde waarde kan zijn van nieuwe opties voor ICT-ondersteuning, zoals voor bijvoorbeeld voor de cordinatie van binnenvaart en voor de schatting van betrouwbaarheid van equipment.

Doel van dit proefschrift is de ontwikkeling van een ICT-framework voor de ondersteuning van beslissingsprocessen, met het uiteindelijke doel de prestaties van PL-systemen te verbeteren. Om dat te bereiken is een hirarchisch ICT-framework opgezet dat bestaat uit twee delen: een communicatie-deel (E: middleware) en gedeelte met intelligente besluitvorming. Voor het middleware-deel is een agent-systeem gekozen. Voor het beslissingsdeel is gekozen voor een meta-heuristiek ter ondersteuning van de collaboratieve planning, en een contextgevoelig systeem voor het schatten van de betrouwbaarheid. Voor de verdere integratie van de voorgestelde ICT-technieken is een hirarchisch framework gemaakt dat bestaat uit drie lagen: agent-modellering, agent-besturing en agent-management. In de modelleringslaag worden de problemen ontleed in onderdelen en gemodelleerd als agent. Op de beslissingsniveau wordt een cordinatie-agent gentegreerd met de besluitvorming. De agent-management-laag verzorgt de communicatie.

Het ontworpen framework is eerst toegepast op een containerterminal, voor de ondersteuning van de planning van het achterlandtransport per binnenschip. Er is een cordinatie- en beslissingsmechanisme gemaakt voor de toepassing van autonome besturing met een multi-agentsysteem en meta-heuristieken. Voor de ondersteuning van de besluitvormig is een hirarchisch algoritme gekozen, een genetisch algoritme gecombineerd met simulated annealing. Boven het beslissingsdeel is een mediator-based agent gemaakt voor de ondersteuning van de communicatie en de cordinatie tussen de terminal en het binnenvaartbedrijf. Een simulatiestudie laat zien dat met het geimplementeerde systeem zowel de communicatie en de cordinatie tussen de terminal en het binnenvaartbedrijf kunnen worden verbeterd, alsook de kwaliteit van het rotatieplan in termen van totale turnaround-tijd en benutting van terminalcapaciteit.

Het framework is ook gebruikt voor de ondersteuning van de bepaling van de betrouwbaarheid van een groot bandtransportsysteem op een terminal voor droge bulkgoederen. Er is een contextgevoelig inspectiesysteem gemaakt ter ondersteuning van de schatting van de betrouwbaarheid van het bandtransportsysteem en de planning van het onderhoud daarvan. Eerst wordt het bandtransportsysteem opgedeeld in componenten en operaties; elke component en elke operatie wordt gemodelleerd als agent. Vervolgens is een supervisie-ontologie ontwikkeld voor het modelleren van de bepaling van de betrouwbaarheid van het equipment. Specifieke kennis wordt verwerkt in de ontologische kennisbank. Door integratie van agent-ontology wordt een nieuw soort ICT-ondersteuning gecreerd. Het voorgestelde systeem is gemplementeerd en gevalideerd voor de inspectie van scheuren in transportbanden, om te laten zien dat het systeem alle metingen en systeeminformatie aan elkaar kan relateren. De metingen en de systeeminformatie kunnen verder worden verwerkt met behulp van een intelligente methode van de agent. Door communicatie en cordinatie van de agents kan worden besloten tot een onderhoudactie. Met de hiervoor beschreven methode worden bestaande inspectiesystemen voor grote bandtransportsystemen verbeterd door een geautomatiseerd systeem waarmee verschillende soorten gegevens kunnen worden onderzocht en met elkaar in verband gebracht voor gentegreerde besluitvorming. Een dergelijk framework kan worden uitgebreid om rekening te kunnen houden met betrouwbaarheidsschattingen van ander equipment.

Samenvattend: In dit proefschrift is onderzoek gedaan aan informatie-integratie en intelligente regeling van PL-systemen door invoering van nieuwe ICT-technieken. De resultaten van het onderzoek laten zien dat gebruik van het opgezette ICT-framework en de bijbehorende technieken mogelijkheden geeft voor verbetering van het PL-systeem op tactisch en operationeel niveau. Met behulp van case studies zijn de realiseerbaarheid en de toepasbaarheid van het voorgestelde geintegreerde ICT-framework en de gekozen ICT-technieken gedemonstreerd.

Summary

Port logistics (PL) can be defined as the process of planning, implementing and controlling the flow of goods and information between the sea and inland via ports and the other way around. PL systems concern the development of functions to support activities including sea side and land side transportation, cargo storage, order processing, and distribution. Increasing demand and a highly competitive market have forced PL systems to continuously improve their performance, including their operational efficiency and reliability. A key issue is the improvement of decision-making abilities. Decision-making systems play an important role within PL systems, especially as they consider the ways that different processes, operations, and equipment can be controlled and coordinated. With the support of ICT technologies, the decision-making systems have significantly developed. However, several decision making processes lack sufficient ICT support. As a result, the benefits of integrating new ICT supports are unknown, including their benefits for inland vessel coordination and the equipment reliability assessments.

The goal of this thesis is to develop an ICT framework to support the decision-making processes and ultimately improve the performance of PL systems. To do so, a hierarchical ICT framework is designed, which consists of two major components: a middleware and an intelligent decision-making approach. With regards of selecting middleware, an agent system is chosen. Likewise, for the selection of intelligent decision-making approach, the meta-heuristics approach is chosen to aid the collaborative planning, whereas context-aware system is chosen for the reliability assessment. To further integrate the selected ICT technologies, a hierarchical framework is designed, which contains three layers: an agent model layer, an agent control layer, and an agent management layer. At the agent model layer, the problems are decomposed and modelled as an agent. At the agent control layer, a coordinate agent is integrated with the intelligent decision-making approach to establish control and coordination. Finally, at the agent management layer, the agent communication facility is established.

The designed framework is first applied to support hinterland barge transport planning at a container terminal. A collaboration and decision-making mechanism is designed for the implementation of autonomous control by means of a multi agent system and meta-heuristics. A bi-level algorithm is also designed and implemented to assist the barge rotation planning. The bi-level algorithm contains a hierarchical structure which combines a genetic algorithm with a simulated annealing algorithm. On top of the the

decision making approach, a mediator-based agent framework is implemented to support the communication and coordination between the terminal and the barge party. A simulation study concludes that the implemented system can improve the coordination between the terminal and the barge, as well as improving the quality of rotation plan in terms of the total turn around time and terminal capacity utilisation.

The designed framework is further applied to support reliability assessments of a large-scale belt conveyor system in a dry bulk terminal. A context-aware supervision system is designed to support condition assessment of a belt conveyor system and its maintenance planning. To begin, the belt conveyor system is decomposed into components and operations. Each component and operation is modelled as an agent. Then, a supervision ontology is developed to model the equipment reliability assessment domain. Specific knowledge is incorporated into the ontological knowledge-base. By means of agent-ontology integration, a novel ICT support is delivered. By studying a case of a belt tear condition supervision, the system is implemented and validated to demonstrate that the system can automatically associate all measurements and system information into a linked context. The linked contexts can be further processed using intelligent assessment approach which is embedded into the agent. Eventually a maintenance action can be made and delivered automatically through agent communication and coordination. This improves the current reliability assessment of large scale belt conveyor system by delivering an automated system where different categories of data are mined and linked together to enable integrated decision-making. Furthermore, such a framework can be extended to account for other equipment reliability assessments within a PL system

To conclude, this thesis investigates the information integration and intelligent control of PL systems by introducing new ICT technologies. The results of this thesis indicate the potential for applying the designed ICT framework and technologies to improve the performance of the PL system at tactical and operational decision-making levels. Utilising the case studies, the feasibility and the applicability of the integrated ICT framework and the selected ICT technologies are demonstrated.

Curriculum Vitae

Fan Feng was born in June 7th, 1987 in TaiYuan, ShanXi, China. He obtained his B.Sc degree in Automation in China. Since 2011, he continue his master study in the Netherlands. He obtained his M.Sc degree in Embedded System from Eindhoven University of Technology in 2013.

Starting from 2013, Fan became a PhD candidate at the Department of Maritime and Transport Technology, Delft University of Technology, the Netherlands. The research is sponsored by the Chinese Scholarship Council. His coauthored papers got the Best Student Paper in the IEEE SOLI conference in 2014 and the Best Application Paper in the IEEE CSCWD conference in 2015. His research focused on proposing and implementing novel ICTs to improve decision-making performance in port logistics systems. His research interests include multi-agent system, artificial intelligence, semantic technology, operations research, and decision-making problems in port logistics systems.

List of publications

Feng, F., Pang, Y., Lodewijks, G. Integrate multi-agent planning in hinterland transport: design, implementation and evaluation. *Advanced Engineering Informatics*, vol 29(4), pp 1055-1071, 2015, https://doi.org/10.1016/j.aei.2015.08.001

Feng, F., Pang, Y., Lodewijks, G., Li, W. Collaborative framework of an intelligent agent system for efficient logistics transport planning. *Computers & Industrial Engineering*,vol 112, pp 551-567, 2017, https://doi.org/10.1016/j.cie.2016.12.044

Feng, F., Pang, Y., Lodewijks, G. Towards Context-Aware Supervision for Logistics Asset Management: Concept Design and System Implementation. In *Information Technology for Management: New Ideas and Real Solutions*, pp. 3-19, Springer, Cham.

Feng, F., Pang, Y., Lodewijks, G. An intelligent agent-based information integrated platform for hinterland container transport. In *Proceedings of the IEEE International*

Conference on Service Operations and Logistics, and Informatics (SOLI), QingDao, China, pp. 84-89, October,2014. (**Winner of the best student paper award**)

Feng, F., Pang, Y., Lodewijks, G., Li, W. Agent-based negotiation and decision-making for efficient hinterland transport plan. In *Proceedings of the IEEE 19th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2015, Calabria, Italy, pp. 395-400, May 2015. (**Winner of the best application award**)

Feng, F., Pang, Y., Lodewijks, G. Application of hybrid algorithm to joint decision making in hinterland barge transport planning. In *Proceedings of the 45th International Conference on Computers and Industrial Engineering (CIE)*. October, Metz, France, 2015.

Feng, F., Pang, Y., Lodewijks, G. An intelligent context-aware system for logistics asset supervision service. In *Proceedings of the IEEE Federated Conference on Computer Science and Information Systems (FedCSIS)*, Gdansk, Porland , pp. 1147-1152, Spetember 2016.

Feng, F., Pang, Y., Lodewijks, G., Cloud & IOT -enabled Prognosis for Asset Management in Transportation and Logistics: A Paradigm for Intelligent Control of Belt Conveyor System. Invited talks in *International Forum of IoT and Applications*, WuHan, China. November, 2015

Feng, F., Pang, Y., Lodewijks, G., Context-aware supervision framework for reliability control of large-scale material handling systems: an ontology-agent integrated approach. Submitted to a journal, 2018.

TRAIL Thesis Series

The following list contains the most recent dissertations in the TRAIL Thesis Series. For a complete overview of more than 200 titles see the TRAIL website: www.rsTRAIL.nl.

The TRAIL Thesis Series is a series of the Netherlands TRAIL Research School on transport, infrastructure and logistics.

Feng, Fan, *Information Integration and Intelligent Control of Port Logistics System*, T2019/1, January 2019, TRAIL Thesis Series, the Netherlands

Beinum, A.S. van, *Turbulence in Traffic at Motorway Ramps and its Impact on Traffic Operations and Safety*, T2018/12, December 2018, TRAIL Thesis Series, the Netherlands

Bellsol Olba, X., *Assessment of Capacity and Risk: A Framework for Vessel Traffic in Ports*, T2018/11, December 2018, TRAIL Thesis Series, the Netherlands

Knapper, A.S., *The Effects of using Mobile Phones and Navigation Systems during Driving*, T2018/10, December 2018, TRAIL Thesis Series, the Netherlands

Varotto, S.F., *Driver Behaviour during Control Transitions between Adaptive Cruise Control and Manual Driving: empirics and models*, T2018/9, December 2018, TRAIL Thesis Series, the Netherlands

Stelling-Koczak, A., *Cycling Safe and Sound*, T2018/8, November 2018, TRAIL Thesis Series, the Netherlands

Essen, van M.A., *The Potential of Social Routing Advice*, T2018/7, October 2018, TRAIL Thesis Series, the Netherlands

Su, Zhou, *Maintenance Optimization for Railway Infrastructure Networks*, T2018/6, September 2018, TRAIL Thesis Series, the Netherlands

Cai, J., *Residual Ultimate Strength of Seamless Metallic Pipelines with Structural Damage*, T2018/5, September 2018, TRAIL Thesis Series, the Netherlands

Ghaemi, N., *Short-turning Trains during Full Blockages in Railway Disruption Management*, T2018/4, July 2018, TRAIL Thesis Series, the Netherlands

Gun, van der J.P.T., *Multimodal Transportation Simulation for Emergencies using the Link Transmission Model*, T2018/3, May 2018, TRAIL Thesis Series, the Netherlands

Van Riessen, B., *Optimal Transportation Plans and Portfolios for Synchromodal Container Networks*, T2018/2, March 2018, TRAIL Thesis Series, the Netherlands

Saeedi, H., *Network-Level Analysis of the Market and Performance of Intermodal Freight Transport*, T2018/1, March 2018, TRAIL Thesis Series, the Netherlands

Ypsilantis, P., *The Design, Planning and Execution of Sustainable Intermodal Port-hinterland Transport Networks*, T2017/14, December 2017, TRAIL Thesis Series, the Netherlands

Han, Y, *Fast Model Predictive Control Approaches for Road Traffic Control*, T2017/13, December 2017, TRAIL Thesis Series, the Netherlands

Wang, P., *Train Trajectory Optimization Methods for Energy-Efficient Railway Operations*, T2017/12, December 2017, TRAIL Thesis Series, the Netherlands

Weg, G.S. van de, *Efficient Algorithms for Network-wide Road Traffic Control*, T2017/11, October 2017, TRAIL Thesis Series, the Netherlands

He, D., *Energy Saving for Belt Conveyors by Speed Control*, T2017/10, July 2017, TRAIL Thesis Series, the Netherlands

Beinovi, N., *Integrated Capacity Assessment and Timetabling Models for Dense Railway Networks*, T2017/9, July 2017, TRAIL Thesis Series, the Netherlands

Chen, G., *Surface Wear Reduction of Bulk Solids Handling Equipment Using Bionic Design*, T2017/8, June 2017, TRAIL Thesis Series, the Netherlands

Kurapati, S., *Situation Awareness for Socio Technical Systems: A simulation gaming study in intermodal transport operations*, T2017/7, June 2017, TRAIL Thesis Series, the Netherlands

Jamshidnejad, A., *Efficient Predictive Model-Based and Fuzzy Control for Green Urban Mobility*, T2017/6, June 2017, TRAIL Thesis Series, the Netherlands

Araghi, Y., *Consumer Heterogeneity, Transport and the Environment*, T2017/5, May 2017, TRAIL Thesis Series, the Netherlands

Kasraian Moghaddam, D., *Transport Networks, Land Use and Travel Behaviour: A long term investigation*, T2017/4, May 2017, TRAIL Thesis Series, the Netherlands

Smits, E.-S., *Strategic Network Modelling for Passenger Transport Pricing*, T2017/3, May 2017, TRAIL Thesis Series, the Netherlands

Tasseron, G., *Bottom-Up Information Provision in Urban Parking: An in-depth analysis of impacts on parking dynamics*, T2017/2, March 2017, TRAIL Thesis Series, the Netherlands

Halim, R.A., *Strategic Modeling of Global Container Transport Networks: Exploring the future of port-hinterland and maritime container transport networks*, T2017/1, March 2017, TRAIL Thesis Series, the Netherlands