2012

An Agent-based Approach for Improving the Performance of Distributed Business Processes in Maritime Port Community

Abdul-Mageed, Loay

http://hdl.handle.net/10026.1/1239

http://dx.doi.org/10.24382/4256

University of Plymouth

COPYRIGHT STATEMENT

An Agent-based Approach for Improving the Performance of Distributed Business Processes in Maritime Port Community

by

Loay Abdul-Mageed El-Sayed

B.Sc., M.Sc.

A thesis submitted to the University of Plymouth
in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

School of Computing, Communications and Electronics

Faculty of Science and Technology

December 2012

Abstract

An Agent-based Approach for Improving the Performance of Distributed Business Processes in Maritime Port Community

Loay Abdul-Mageed El-Sayed
B.Sc., M.Sc.

In the recent years, the concept of *"port community"* has been adopted by the maritime transport industry in order to achieve a higher degree of coordination and cooperation amongst organizations involved in the transfer of goods through the port area. The business processes of the port community supply chain form a complicated process which involves several process steps, multiple actors, and numerous information exchanges. One of the widely used applications of ICT in ports is the Port Community System (PCS) which is implemented in ports in order to reduce paperwork and to facilitate the information flow related to port operations and cargo clearance. However, existing PCSs are limited in functionalities that facilitate the management and coordination of material, financial, and information flows within the port community supply chain.

This research programme addresses the use of agent technology to introduce business process management functionalities, which are vital for port communities, aiming to the enhancement of the performance of the port community supply chain.

The investigation begins with an examination of the current state in view of the business perspective and the technical perspective. The business perspective focuses on understanding the nature of the port community, its main characteristics, and its problems. Accordingly, a number of requirements are identified as essential amendments to information systems in seaports. On the other hand, the technical perspective focuses on technologies that are convenient for solving problems in business process management within port communities. The research focuses on three technologies; the workflow technology, agent technology, and service orientation.

An analysis of information systems across port communities enables an examination of the current PCSs with regard to their coordination and workflow management capabilities. The most important finding of this analysis is that the performance of the business processes, and in particular the performance of the port community supply chain, is not in the scope of the examined PCSs.

Accordingly, the Agent-Based Middleware for Port Community Management (ABMPCM) is proposed as an approach for providing essential functionalities that would facilitate collaborative planning and business process management. As a core component of the ABMPCM, the Collaborative Planning Facility (CPF) is described in further details. A CPF prototype has been developed as an agent-based system for the domain of inland transport of containers to demonstrate its practical effectiveness.

To evaluate the practical application of the CPF, a simulation environment is introduced in order to facilitate the evaluation process. The research started with the definition of a multi-agent simulation framework for port community supply chain. Then, a prototype has been implemented and employed for the evaluation of the CPF. The results of the simulation experiments demonstrate that our agent-based approach effectively enhances the performance of business process in the port community.

Table of Contents

List of Figures

List of Tables

Acknowledgements

The work, and indeed this Ph. D. thesis, would not have been possible without the help and support of my Director of Studies, Dr. Chris Johnson. Thanks go to him for his tireless effort in steering me through the Ph. D. process. His professionalism and experience has been invaluable, and I owe much of my success to his guidance.

Thanks must also go to my other supervisors, Dr. Andy Phippen and Dr. Jonathan Moizer, who have provided helpful guidance throughout my studies.

Finally, I wish to thank my friends and family for their continuous support. In particular, I wish to thank my sweet wife and my beloved daughter, my brothers, and my sister.

AUTHOR'S DECLARATION

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award.

Relevant scientific seminars and conferences were regularly attended and external institutions were visited for consultation purposes.

Word count of main body of thesis: 55832 words.

Signed: ....

Date: ....

Chapter 1

Introduction and Overview

1 Introduction and Overview

1.1 Introduction

Traditionally, seaports[1] are considered to act as an interface between sea and land through which goods and passengers are transferred. Recently, this role has been changed due to globalization and the rise of *"global supply chains"* where various supply chain activities are performed in different parts of the world.

Global supply chains are actually complex networks, which consist of many different stakeholders, including shippers, deep-sea carriers, port operators, and customs organizations. Seaports constitute the links in global supply chains at geographical, political, and commercial boundaries (Robinson, 2002). The developments in global supply chains have compelled seaports to address the need for managing complex business processes and information flows. To address these challenges, seaports need special Information and Communications Technology (ICT) capabilities for information sharing, planning and executing cargo transport in a collaborative way, and establishing inter-organizational information architectures.

Consequently, contemporary seaports have a new role as a more highly developed logistics centre; within which all activities relating to transport, logistics and the distribution of goods, both for national and international transit, are carried out by various operators on a commercial basis. In that sense, seaports become a part of a

[1] For simplicity, the term *"port"* will be plainly used to refer to *"seaport"* throughout this thesis.

cluster of organizations in which different logistics and transport operators are involved in bringing value to the final consumers. In order to be successful, such channels need to achieve a higher degree of *coordination* and *cooperation* (De Souza *et al.*, 2003; Song and Panayides, 2008). Therefore, the concept of *"port community"* has been adopted by the maritime transport industry.

The port community is a concept that goes far beyond the mere traditional port boundaries. It includes all activities in the port area or relevant to port production, whether they are public or private, commercial or administrative. Usually, the port community is the formal association of all organizations and persons involved in the transfer of goods and passengers through the port area. This implies that the port community is a *complex system* with many internal and external actors involved in the transfer of goods and passengers through the port area.

The business processes of the port community form a complicated process; involving several process steps, multiple actors, and numerous information exchanges. In such a process, each actor requires specific information and in turn generates new information; each actor has specific responsibilities and interests; and actors differ in their structures and motivations. For example, shipping and forwarding agents are private profit-making bodies, while others, such as the Port Authority, Customs, or the health inspection organizations are non-profit-making.

Due to process complexity, several bottlenecks may arise in the business processes of the port community supply chain. We perceive that the *lack of coordination* and the *lack of disruption management* are two bottlenecks that adversely affect the efficiency of the port community. *It is this need for coordination and disruption management which motivates our research programme for improving the performance of distributed business processes in maritime port community.*

The need for coordination in port communities arises from the role of contemporary seaports as links in the global supply chain. Consequently, the role of seaports has been dramatically changed into the management and coordination of material, financial, and information flows within global supply chains (Carbone and De Martino, 2003). *It is this need for coordination which motivates our current research using an agent-based approach.*

Operational disruptions may occur in daily business due to unexpected events, such as administrative errors and delays. In the complicated business process of the maritime transport of containers, disruptions may occur both in the physical and in the information flow. These disruptions presumably have impacts on the safety and security of the port community supply chain. They introduce a potential risk for smuggling or stealing of containers or their content; therefore, they make the supply chain more vulnerable. In addition, these disruptions are a source of inefficient container handling, and this leads to delays and high waiting times. Disruption handling should be integrated into everyday operations; and in particular, information systems should be adjusted in such a way that they support the handling of disruptions as well as procedures for business as usual.

The main characteristics of the port community supply chain can be categorised into two groups; characteristics regarding the environment itself (i.e. the port community) and characteristics of the members of the community.

The port community environment is highly dynamic where the business process activities are independent and distributed among players of the port community. It is Ill-structured - loosely coupled network where the business relations between players do not show a clear hierarchy. The port community environment is also characterized by frequent disturbances (disruptions) that may occur during execution and which may affect the operations.

On the other hand, members of the port community are autonomous entities with poor contractual agreements; and therefore, a member cannot contractually force other members to deliver a certain service or charge for poor services. In addition, the port community members have conflict of interest and they are reluctant to share information since they believe that this might degrade there competitive advantages.

These characteristics reflect the suitability of the application of Multi-Agent Systems (MAS) which is an intelligent distributed approach. The MASs are appropriate for applications that are modular, decentralised, changeable, ill-structured, and complex (Jennings, 1996). *The adoption of agent-based approach in this research work is motivated by the resemblance of the port community characteristics with those of the Multi-Agent Systems (MAS).*

Due to the paradigm shift of the role of seaports as a node in the global supply chain, as described above, seaports no longer compete as solely autonomous entities but rather as supply chains. Therefore, in order to achieve competitive advantage, seaports are investing in infrastructure for improving the way they perform their operations. These investments include construction of new deep berths, the use of advanced technologies for container handling and terminal automation to speed up the operations, and the application of Information and Communication Technology (ICT) to improve the efficiency of the container handling process.

One of the widely used applications of ICT in ports is the *"Port Community System (PCS)"*. It is implemented in ports in order to reduce paperwork and to facilitate the information flow related to port operations and cargo clearance. Generally, the PCS is the Industry Inter-Organisational Information System (IOIS) that has the capability to bind together multiple information systems operated by competing and cooperating firms that make up the port community. Existing PCSs are limited in functionalities that facilitate the management and coordination of material, financial, and information flows within the port community supply chain.

In this perspective, it is necessary that the existing PCSs should be augmented by functionalities to satisfy the needs of such a distributed, dynamic, collaborative, and competitive business community. *It is this need for the adjustment of information systems which motivates our proposed use of a middleware as an amendment to the existing port community systems.*

Agent technology seems to have significant contribution to the advancement of the domain of transport logistics since it has been applied to a number of various problem areas within transport logistics (Lee *et al.*, 2002; Thurston and Hu, 2002; Franz *et al.*, 2007). MASs are based on information exchange, communication flows, and negotiations between different agents. As such, MASs could be a mechanism to establish future port community systems that focus on solving problems through synchronization of activities (by means of communication and negotiation). However, agent-based systems have not been widely adopted for port community management yet. We perceive that there is a gap between research and practise concerning inter-organisational multi-agent system application in port community supply chains. *It is this "gap" which motivates our research to introduce a middleware as an improvement to the existing port community systems.*

1.2 Aims and objectives of the research

The research programme described in this thesis attempts to address the utilization of agent technology in combination with workflow concepts aimed at the enhancements of the performance of port community supply chains.

The aim of this research is to define, design and evaluate an agent-based solution suited to improve the performance of business processes in maritime port community. Our general aim in this research programme is to investigate the appropriateness of agent technology to business needs of the port community environment, and then the creation of a prototype is a way of achieving our aim.

Accordingly, the objectives for the research programme can be summarised as follows:

1. To understand the port community and the need for coordination and workflow management (port community problems).
2. To investigate the application of current computer systems technologies for business process management within the port community context.
3. To assess the information systems that currently exist in port communities with regard to the coordination and workflow management capabilities.
4. To design an agent-based middleware to provide effective business process management of a port community supply chain.
5. To implement and test a prototype of the agent-based middleware to evaluate its practical effectiveness.

We consider the research objective no.3, which is primarily based upon the research objectives 1 and 2, a fundamental contribution to knowledge. It has not previously been an analysis of information systems technologies across port communities with regard to the coordination and workflow management capabilities. Therefore, we consider it as a fundamental contribution to knowledge since it would provide further insights into we know ever.

The objectives outlined above relate to the sequence of material presented in the following chapters in the thesis, the outline structure for which is presented in Section 1.4.

1.3 Research Approach

Generally, researchers gain knowledge, concrete understanding and solve problems by employing tools and techniques provided by research methodologies. *"A research method is a strategy of inquiry which moves from the underlying philosophical assumptions to research design and data collection"* (Myers, 1997).

In this research, we follow a design science approach that takes the perspective of new artefact and theory development by means of explorative methodologies. Since more than one methodology approach may be used for answering a research question (Yin, 1994); the explorative research in this work uses a mixed method approach. We conducted desk research and interviews with practitioners, researchers, and potential users. The use of these methods in this research work is described below.

Design science research involves the design of novel or innovative artefacts and the analysis of the use and/or performance of such artefacts to improve and understand the behaviour of aspects of Information Systems. A series of scholars perceive design research as essential in making research more relevant to real interest in practise (March and Smith, 1995; Hevner *et al.*, 2004; Verschuren and Hartog, 2005; Cross, 2007).

March and Smith (1995) explain that "design science consists of two basic activities, build and evaluate. Building is the process of constructing an artefact for a specific purpose; evaluation is the process of determining how well the artefact performs. Hevner *et al.* (2004) provide a framework and a set of seven guidelines which help information systems researchers to conduct, evaluate and present design science research. The seven guidelines address: design as an artefact, problem relevance, design evaluation, research contributions, research rigor, design as a search process, and research communication (Hevner *et al.*, 2004). We follow these guidelines to conduct the research programme introduced in this thesis.

The process of a typical design science research effort proceeds following these common steps: problem identification and motivation, definition of the objectives for a solution, design and development, demonstration, evaluation, and communication (Peffers et al., 2008). A short description of these steps, compiled from Peffers et al., (2008), is introduced in Table 1-1 and accompanied with the corresponding chapters of this thesis and the research method(s) that is used in this research programme for each step.

The work in this thesis is based on various fields of science and scientific theories. The main area of research is computing which has been supported by various fields such as maritime economics, transportation management, and logistics and supply chain management. All these disciplines have their own methodological approaches and preferences, and they also have their own perspectives on design research and evaluation.

Process Step	Description	Chapter	Research Method
Problem identification and motivation	Define the specific research problem and justify the value of a solution. Justifying the value of a solution accomplishes two things: it motivates the researcher and the audience of the research to pursue the solution and to accept the results and it helps to understand the reasoning associated with the researcher's understanding of the problem. Resources required for this activity include knowledge of the state of the problem and the importance of its solution.	2 and 3	Literature review. Interview.
Definition of the objectives for a solution	Infer the objectives of a solution from the problem definition and knowledge of what is possible and feasible. Resources required for this include knowledge of the state of problems and current solutions, if any, and their efficacy.	4	Literature review. Interview. Explorative and industry evaluation
Design and development	Create the artefact. Such artefacts are potentially constructs, models, methods, or instantiations or new properties of technical, social, and/or informational resources. This activity includes determining the artefact's desired functionality and its architecture and then creating the actual artefact. Resources required moving from objectives to design and development include knowledge of theory that can be brought to bear in a solution.	5 and 6	Design and feedback cycles
Demonstration	Demonstrate the use of the artefact to solve one or more instances of the problem. This could involve its use in experimentation, simulation, case study, proof, or other appropriate activity. Resources required for the demonstration include effective knowledge of how to use the artefact to solve the problem.	6 and 7	Prototype
Evaluation	Observe and measure how well the artefact supports a solution to the problem. Depending on the nature of the problem venue and the artefact, evaluation could take many forms. It could include such items as a comparison of the artefact's functionality with the solution objectives from the second activity above, objective quantitative performance measures (e.g. budgets or items produced), the results of satisfaction surveys, client feedback, or simulations.	7	Simulation
Communication	Communicate the problem and its importance, the artefact, its utility and novelty, the rigor of its design, and its effectiveness to researchers and other relevant audiences, such as practicing professionals, when appropriate. Communication requires knowledge of the disciplinary culture.	6 and 7	Interview

Table 1-1: Design Science Methodology Steps.

The premise for adopting a multidisciplinary approach is the need for a better understanding of the domain and possible solutions. This approach is quite time consuming and can be difficult for an individual to grasp such a broad view of the subject. However, compared with a single perspective approach, the advantage of a multidisciplinary approach for conducting research is that a thorough and more complete picture of the characteristics of the domain are realized and captured.

The choice of research methods in this work was driven by the research problem, the context to be investigated, and the existing research. In this research programme, three methods have been used; namely, they are literature review, interviews, and simulation. The use of these methods in this research programme is described in the following.

Literature review: During the initial phase of the research, a review of journals, periodicals, projects reports, and other research publications related to the subject area has been carried out. Periodically, the review was updated throughout the research programme. The purpose was to obtain a concrete understanding of what a port community is, the impact of its performance on country and global economy, problems that the port community may encounter, and what has been done in the domain of port community management and coordination for improving its performance.

An overview of relevant literature is provided in Chapter 2. A second literature review has been carried out and presented in Chapter 3. The second literature review focuses on the use of workflow management, agent technology, and Web Services as enabler technologies for the management and coordination of port community and supply chain. The purpose of that survey was to enhance our understanding of agents, Multi-Agent Systems (MAS), and Web Services (WS) and their application to solve problems related to those studied in this research programme.

Interviews: The primary data collection method was semi-structured[2] interviews that took place during port community field visits. The interviews were carried out using a set of open-ended questions. Interviews were conducted with personnel working for major players in the port community of the Port of Alexandria; such as port authority, container terminal, shipping lines, shipping agents, freight forwarder, customs, and inland transporters.

The primary objective of the interviews is to get more insights and concrete understanding of the business network formed by the firms participating in the port community. This objective has been split into several levels: 1) the role of each individual firm; 2) the vertical and horizontal transaction relationships between firms in the business network; 3) the impact of other industries (i.e. external environment); 4) the nature of the Information technologies employed for doing their business; and 5) to identify areas considered to be bottlenecks.

The gathered data were mostly qualitative and unstructured. The interviews typically lasted between 1 hour and 2 hour. Interviews were conducted with authorized personnel such as CEOs, operations managers, IT managers and developers, and

[2] Semi-structured interviews are described as containing predetermined questions, the order of which can be changed by the interviewer. Also, the interviewer can change wordings, give explanations, and add or omit questions (Robson, 2002).

consultants. Interviews were conducted in Arabic and took place in a personal meeting (face-to-face interviews) or administered over the phone whichever is possible. Meeting minutes were prepared for each individual interview to summarize the discussions. Throughout this research programme, more focused interviews were conducted with selected managers and users to validate the work presented in this thesis.

Simulation: Since its inception, simulation has been applied to various sectors, such as manufacturing, services, defence, healthcare, and public services. It is recognised as one of the most widely used techniques in the field of operations management. The simulation method can be considered as a good alternative strategy for implementing a case study. A distinctive characteristic of simulation as an experimental approach is to evaluate and test one or more solutions. Accordingly, we propose an agent-based port community simulation framework in Chapter 7, and then we used a prototype implementation of such framework to evaluate the effectiveness of a new agent-based collaborative planning facility (described in Chapter 5 and Chapter 6).

1.4 Thesis Structure

The thesis has been structured so that most of the background information (mainly the state of the art survey work) is confined to the initial chapters. The thesis addresses the aforementioned objectives in order and the remainder of the thesis is comprised of the following chapters.

Chapter 2 –Business View: The Port Community – This chapter primarily contributes to the first research objective of the list introduced in Section 1.2. The chapter introduces an outline of the maritime transport business with emphasis on seaports and their role in the global supply chain. In particular, it discusses the role of ports and how it went through evolutionary phases in order to cope with the changes that business environment undergoes. This chapter also discusses the concept of port community emphasising the complex interrelationships of the major players in such community. Consequently, the characteristics and problems of the port community are discussed, and then the chapter ends with a number of requirements that we perceive essential for information systems in seaports to confront the contemporary business environment.

Chapter 3 – Technical View: Process, Agents, and Web Services – This chapter primarily contributes to research objective no. 2. This chapter introduces three concepts that are accepted as a proper mix for solving problems in business process management. These concepts are *process* represented by workflow management technology, *intelligence* represented by agent technology, and *dynamic interoperability* represented by service oriented technology. The chapter investigates and discusses the need for integrating different technologies and how the three aforementioned technologies can be combined in order to realize an acceptable effective management of the business process; especially in a port community.

Chapter 4 – **Port Community Systems: An Analytical Perspective** – This chapter contributes to research objective no.3. In this chapter, we describe and assess the utilization of information technology in seaports. This chapter gives more insight into definition, structure, functionalities and services provided by Port Community Systems (PCSs) in practice. In this chapter, we introduce a survey that has been carried out in order to assess PCSs in a number of top ports of the world. The objective of the survey is to assess whether PCSs provide functionalities to facilitate the management and coordination of port community business processes.

Chapter 5 – **The Agent-Based Middleware for Port Community Management (ABMPCM)** – This chapter contributes to research objective no.4. In this chapter, we introduce our ABMPCM framework with a case study to demonstrate the functionality of different components of the framework. In this chapter, we first describe the characteristics of the port community supply chain and the need for multi-agent systems for managing this chain. We then propose our approach to provide business process management tasks using the ABMPCM. A business scenario to explain the usability of the ABMPCM is also presented.

Chapter 6 – **Implementation of the Proposed Agent-Based Middleware** – This chapter contributes to research objective no.5. In this Chapter, we propose the Collaborative Planning Facility (CPF) framework for bilateral coordination between the members of the port community. The CPF is one of the main components of the ABMPCM framework. It addresses the problems of specifications, automation with optional consumer feedback, and business process tasks coordination in a flexible intelligent agent-based framework. The effectiveness of the proposed CPF is described through its practical application in the domain of inland transport of containers which is an inter-organisational process.

Chapter 7 – **Agent-Based Port Community Simulation Model** – This chapter contributes to research objective no.5. In this chapter, we introduce a proposed multi-agent simulation framework that can be used to study the port community supply chain and the effect of various business scenarios on its performance. An implementation of a prototype of the proposed agent-based simulation framework is also introduced. This prototype is used for the evaluation of the practical application of the CPF (Chapter 6). The description of these experiments as well as the results is introduced in this chapter.

Chapter 8 – **Conclusion and Future Work** – The final chapter assesses the research and whether the objectives were successfully met. It presents the main conclusions arising from the research, highlighting the key achievements and limitations. It discusses some of the future work directions.

A number of appendices are also included, which provide a range of supporting materials and code listings.

Chapter 2

Business View: The Port Community

2. Business View: The Port Community

2.1 Introduction

The main objective of this chapter is to provide a background that motivates this research. In addition, it will familiarize the reader with the domain of maritime transport, and in particular the seaports and the concept of port community.

This chapter starts with an overview on seaports and the evolution of its role, and then we introduce a detailed description of the maritime port community in Section 2.3. This includes a discussion on the actors of the port community and their relationships. In Section 2.4, the characteristics of the port community are identified; these are the bases of the design of the proposed middleware (see Chapter 5).

The bottlenecks that arise from the basic business processes in the port community are introduced in Section 2.5. Essentially, it is these problems which motivate our proposed use of the agent-based middleware. Accordingly, we propose a number of requirements to be amended to information systems in order to enhance the efficiency of the port community business process. These requirements are considered in our proposed agent-based middleware.

2.2 Seaports: A General Overview

Generally, a seaport is a location on a coast or shore containing one or more harbours where ships can dock and transfer people or cargo to or from land. The relevance of seaports in the efficient working of an economy cannot be understated since all goods and passengers transported by sea require the use of, at least, two ports. In the majority of countries, most international trade (export/import), and in some cases also a large share of domestic trade, is done through maritime transport. For simplicity, the term *"port"* will be plainly used to refer to *"seaport"* throughout this thesis.

Seaborne traffic depends on seaports for all its operations, since ports act as interfaces between maritime and inland modes of transport (railways, road or inland navigation). Therefore, in order to have an efficient maritime transport system, seaports must be guaranteed to work efficiently. The basic objective of a seaport is to provide a fast and safe transit of goods and passengers through its facilities, so that generalized costs for passengers and for shippers are minimized.

Traditionally, ports are considered to act as an interface between sea and land. In particular, it is an interface between ships and shore by providing shelter and berthing space, temporary storage and the provision of superstructure and infrastructure for cargo operation and movement within the port. As gateways, ports have always been sensitive to changes in socio-economic trends. Several market-related and societal trends; such as globalization, containerization, and environmental and ecological concerns have created a highly uncertain and complex environment for ports and fundamentally changed the port concept.

The evolution of the port concept has been categorized into four successive *"generations"*, as shown in Table 2-1 (UNCTAD, 1992a; UNCTAD, 1992b). Although this is outside the scope of this study, it is worth mentioning that this approach has been criticized since ports evolve continuously rather than in discrete steps and because the composite reality of ports whereby several streams of evolution can be observed simultaneously in one and the same port would be ignored (Beresford *et al.*, 2004; Bichou and Gray, 2005). However, whilst the generations approach cannot pretend to provide more than a series of snapshots, it does provide some useful insights in the evolution of larger multi-purpose gateway ports.

First generation	Prior to 1950	Sea approach, transfer of goods, temporary storage, delivery
Second generation		Includes first plus industrial and commercial activities which add value to the goods. (i.e. the port is a handling and services centre).
Third generation	Since 1980	Includes first and second generations plus structuring of the port community, empowering links between town and port and among port users, extension of the services provided beyond port boundary and integrated system of data collection a processing. The port becomes a logistics platform for trade.
Fourth generation	Since 2000	Network of physically separated ports (or terminals) linked through common operation or through common administration.

Table 2-1: Generations of Seaports (UNCTAD 1992a; UNCTAD, 1992b)

This categorization points out the evolution of the role of ports. The work in this thesis is primarily inspired by the evolution characteristics of the third and fourth generations.

In the third generation, the port is an integrated transport centre and logistics platform for international trade. It incorporates logistic functions, linked with the distribution of goods, with information processing and telecommunications contributing to generate added value. The realization of the third generation ports principally depends on the presence of an efficient port community. A detailed description of the port community concept is presented in the Section 2.3 below.

In the fourth generation, the *"Network Port"*, the role of port community becomes more crucial. The fourth generation port is characterized by collaboration between port communities and by the internationalization and diversification of activities. A "network port" is integrated into the international transport logistics chain, with "door-to-door" services, along with other logistics operators that operate in various ports. This integration is illustrated in Figure 2-1 that shows how seaports are embedded in the global supply chain (van Baalen *et al.*, 2008).

Accordingly, the flows of information increase rapidly in terms of volume and complexity. These inter-organizational flows of information are increasingly supported by advanced *"port community systems"*. A detailed discussion of port community systems is introduced in Chapter 4 as well as an analytical survey of the existing port community systems.

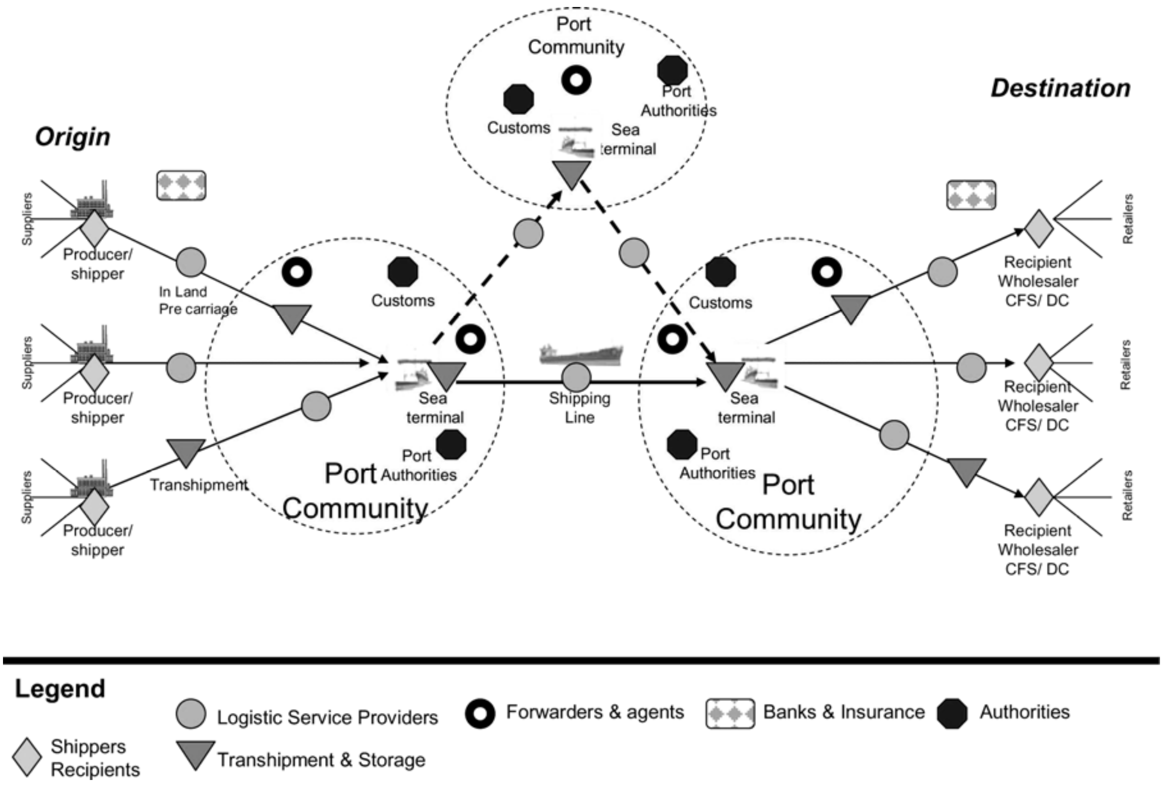

Figure 2-1: Seaports Embedded in the Global Supply Chain (van Baalen *et al.*, 2008).

This changing role of ports, and as a result of the globalization of production and distribution, several studies have been carried out pointing to the importance of supply chain orientation for ports and have also recognizing that ports are increasingly integrated in supply chains (Robinson, 2002; Paixão and Marlow, 2003; Marlow and Paixão, 2003; Carbone and De Martino, 2003; Bichou and Gray, 2004; Notteboom and Rodrigue, 2005; Panayides and Song, 2006; Panayides, 2006). In addition, the role of seaports has been dramatically changed into the management and coordination of material, financial, and information flows within global supply chains (Carbone and De Martino, 2003).

The role of a modern seaport can be summarized in the following UNCTAD[1] definition:

> *"Seaports are interfaces between several modes of transport, and thus they are centres for combined transport. Furthermore, they are multi-functional markets and industrial areas where goods are not only in transit, but they are also sorted, manufactured and distributed. As a matter of fact, seaports are multi-dimensional systems, which must be integrated within logistic chains to fulfil properly their functions. An efficient seaport requires, besides infrastructure, superstructure and equipment, adequate connections to other transport modes, a motivated management, and sufficiently qualified employees."*

This definition indicates that a seaport is not merely an organization that provides a single service, but instead, seaports provide multiple activities. These activities are performed by the actors in the port community; this will be described in the Section 2.3 below. Therefore, it is interesting to study in detail all those tasks, in order to evaluate the most efficient provision of these activities. Moreover, it is also relevant to study how these various activities are coordinated.

2.3 The Port Community

As the role of port has progressed from that of being a mere physical point of transit at the sea/land interface, to being a more highly developed logistics centre, the concept of *"port community"* has been adopted (UNCTAD, 1992a; UNCTAD, 1992b) .

The port community includes all activities in the port area or relevant to port production, whether they are public or private, commercial or administrative. The port community is usually the formal association of all organizations and persons involved

[1] The United Nations Conference on Trade and Development.

in the transfer of goods and passengers through the port area which is a concept that goes far beyond the mere traditional port boundaries.

A port community is generally based on two pillars, the port authority and the association of all the port activities and users. In major ports, usually there are groups which represent major activities in the ports, such as associations of freight forwarders, insurers, cargo handling companies, ship owners, shipping agents, etc. The port community is in fact the alliance of these groups in the port area, each group defending their interests and promoting and coordinating their activities.

There exist various forms and structures of port community in different ports; however, their main objective is almost always the same, the well-being of the port. An essential task of the port community is the coordination within the port area as well as with organizations outside the port area.

The port system is a complex system with many internal and external actors that are considered to be stakeholders[2] in a port community, each with their own interests, objectives, and priorities. Before further discussion of the complex relationships among actors of the port community, it is useful to highlight the general view of the maritime transport process.

[2] In the *broad view* stakeholders are described as any individual or group having interest or being affected by the corporation. The *narrow view* only recognises stakeholders whose relationship is primarily of an economic/contractual kind (Shankman, 1999).

2.3.1 The Maritime Transport Chain

For better understanding of the linkages between major actors of the port community, it would be useful to know first how the process of maritime transport of containers is carried out.

Generally, the logistic chain for maritime transport of goods from the country of origin to the country of destination can be portrayed by the sequence of events as illustrated in Figure 2-2.

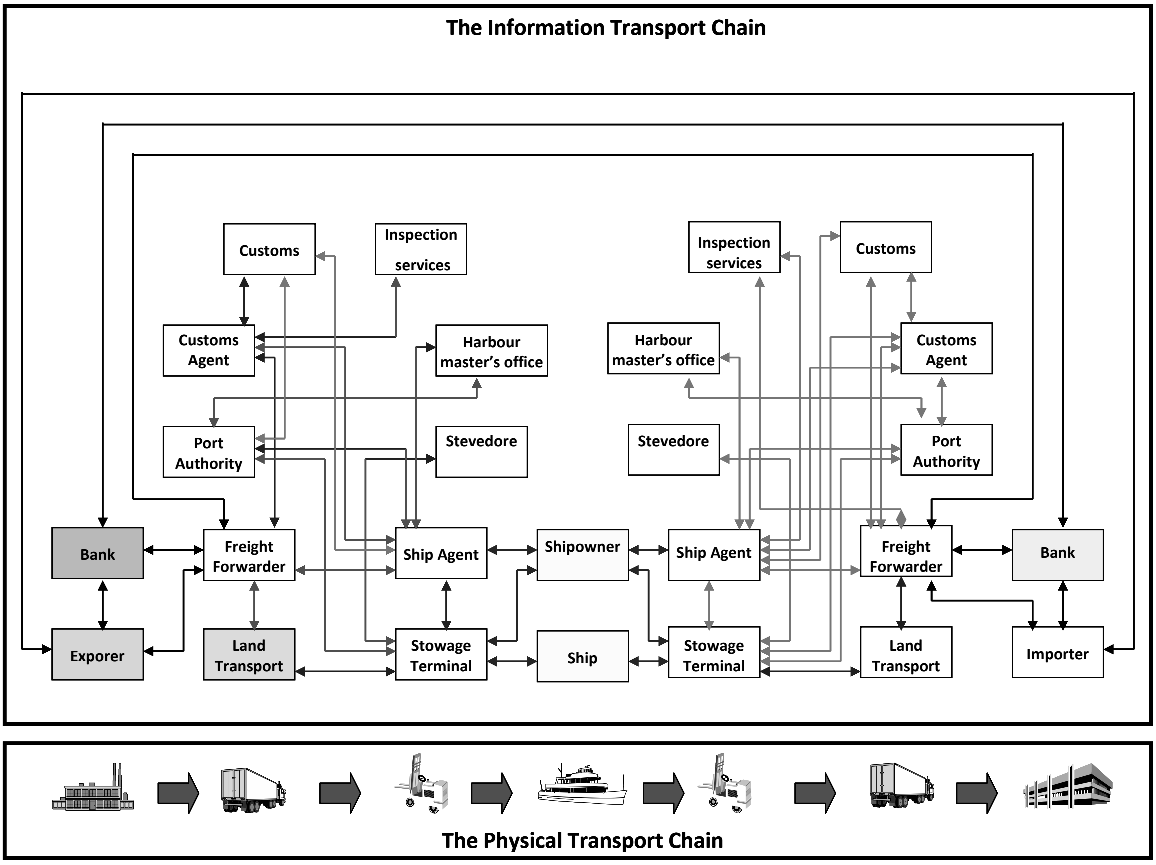

Figure 2-2: The Logistic Chain for Maritime Transport of Goods

This process is carried out in the following steps:

a. At the country of origin, goods are transferred from the location of the sender (i.e. shipper or consigner) to the port of origin using an inland mode of transport.

b. At the port of origin, goods are unloaded from the inland mode of transport and loaded onto the vessel that will transport it to the port of destination.

c. Finally, when the vessel arrives at the port of destination, it docks in the loading terminal where goods are unloaded and transported to the receiver of the goods (i.e. consignee).

Within the context of this sequence of activities, a port is an area in which the goods change their mode of transport. They enter by sea and leave by land (import), or they arrive by land and leave by sea (export). Accordingly, a seaport is an interface between sea transport system on one side, and the land transport network on the other side.

There are basically two forms of interactions in the container transport network: 1) operational interactions related with the physical transfer of cargo (physical transport chain in Figure 2-2), and 2) administrative interactions related with the supervisory and information based exchanges (information transport chain in Figure 2-2). Each member in the transport network operates as a supplier as well as a customer, and generates information that is transferred along the network (van Baalen *et al.*, 2000).

In the landside transport network, companies operate in different roles (e.g. port authority, shipping agents, terminal operators, etc.). These entities form a transport network that allows them to interact and cooperate in order to perform the physical tasks for moving the goods. In particular, this transport network constitutes the port community on which this thesis focus. However, the port community, in a broad sense, incorporates a multitude of actors other than those of the transport network. This issue is discussed in the next section where the major players of the port community are introduced.

2.3.2 Actors of the Maritime Transport Chain

As noted in sub-section 2.3.1 above, a number of players are involved in the maritime transport chain. These actors can be categorized into five distinctive groups based on their role and responsibility (Wagenaar, 1992). Namely, these five groups are customer group, organizing group, physical group, authorizing group, and financial group. Example organizations of these groups are represented in Table 2-2. (For a description of these actors, see Appendix A)

Group	Examples of organizations
1. Customer group	Shipper, Consignee
2. Organizing group	Forwarder; Shipping line agent ; Logistics service provider (4PL)
3. Physical group	Sea terminal operator, Shipping line/sea carrier, Pre- or On-carrier, inland transport, i.e., barge operator, rail operator, road carrier, Inland terminal operator, Logistics service provider (3PL), Empty container depot operator
4. Authorizing group	Customs, Port authorities, Seaport police, River police, Inspection authorities
5. Financial group	Bank, Insurance company

Table 2-2: Organizations Involved in Container Transport; adapted from (Wagenaar, 1992).

Members of the *customer group* (shipper or consignee) are the final customers in container transport where the shipper is the initiator of the container. The actors of the *organizing group* are responsible for the overall planning and control of the whole (or parts of) the container transport chain.

The *physical group* includes organizations that perform the actual transportation and handling of the container from the consigner to the consignee. Organizations in the physical group are operationally responsible for the container and the contents of the container. The *authorizing group* is composed of parties responsible for public infrastructure as well as regulatory authorities that monitor the compliance of the companies in the transport chain to the relevant rules and regulations. The authorizing group is mainly responsible for public safety and security, as well as the inspection of compliance with national and international laws and regulations. Finally, the *financial group* supports financial transactions between organizations in the transport chain.

It is thus obvious that the business process of the port community form a complex system, in which each actor requires specific information and in turn generates new information, and moreover each one has specific responsibilities and interests. These entities widely differ in their structures and motivations. For example, shipping and forwarding agents are private profit-making bodies, while others, such as the Port Authority, Customs, or the Health inspection organizations dependent on the state authorities, are non-profit-making and base their organizations on the legislative provisions of the central government.

In order to give a comprehensive view of the port community, it is worth mentioning its broad perspective. As indicated in Section 2.1, ports act as an interface between sea and land. Therefore, a port both technologically and economically is in fact a node for contacts and contracts, whereby every stakeholder is driven by its own interests and priorities. Ports are associations where a multitude of individuals and interests should collaborate for the creation and distribution of wealth. Hence, the value creation process in ports is dependent upon the support of the different stakeholders groups.

In a broad sense, four main stakeholder groups in a port community have been identified (Notteboom and Winkelmans, 2002):

The internal stakeholders: They are part of the comprehensive port authority organization. For example, stakeholders of this group include port managers, employees, board members, unions and shareholders.

The external stakeholders (economic/contractual): This group includes economic players that are working inside (*in situ*) and outside (*ex situ*) the port boundaries. Different port companies and supporting industries that create added value by directly investing in the port area are considered to belong to the *in situ* group. On the other hand, the *ex situ* group consists of industries located in the foreland and hinterland. A port also is a cluster of strongly intertwined economic activities with linkages to economic activities outside the port perimeter.

Each economic/contractual stakeholder in the port community can be brought into relation with one or more entities/functions within this economic cluster. Some of these companies are mainly involved in physical transport operations linked to cargo flows (e.g. terminal operators and stevedoring companies - including the carrier/terminal operator in case of dedicated terminals). Others solely offer logistical organization services (e.g. forwarding agencies, shipping agencies, etc.).

Industrial companies in the port area (e.g. power plants, chemical companies, assembly plants,), supporting industries (e.g. ship repair, inspection services, etc.) and port labour pools also belong to the group of the first order economic stakeholders. Other economic stakeholder groups include port customers, trading companies and importers/exporters. They are less directly involved than the *in situ* economic groups, as they normally do not invest directly in the port. Nevertheless, they follow the port evolution carefully, because port activity can influence their business results. Moreover, they exert strong demand-pull forces on port service suppliers and as such direct the market requirements to which the port community has to respond.

Legislation and public policy stakeholders: This group includes government departments responsible for transport and economic affairs on a local, regional, and national level. In addition, this group includes environmental departments and spatial planning authorities on the various geographical decision levels.

Community stakeholders: This includes community groups or civil society organizations, the general public, the press, and other non-market players. They are concerned about the port's evolution, i.e. mainly about its expansion programmes, for reasons of the community's well-being. They may experience actual or potential harm or benefits as a result of port action. The various stakeholders of the port community are depicted in Figure 2-3 (Notteboom and Winkelmans, 2002) in which the port community as perceived by many external entities is delineated by a dashed line..

The large number of port stakeholders made port management a complex task. Port managers should acknowledge and whenever possible actively monitor the concerns of all legitimate stakeholders. In decision-making and operations, port managers should appropriately take into account the interests of certain stakeholders; in particular those stakeholders that are most intimately and critically involved. Therefore, port managers need a better understanding of the relationships between and within different stakeholders groups as well as of the objectives and concerns among stakeholders. They seldom have a good overall picture of the underlying dynamics that shape stakeholders relations. In this respect, we are going to introduce an outline on stakeholders' relationships and interactions in the next section.

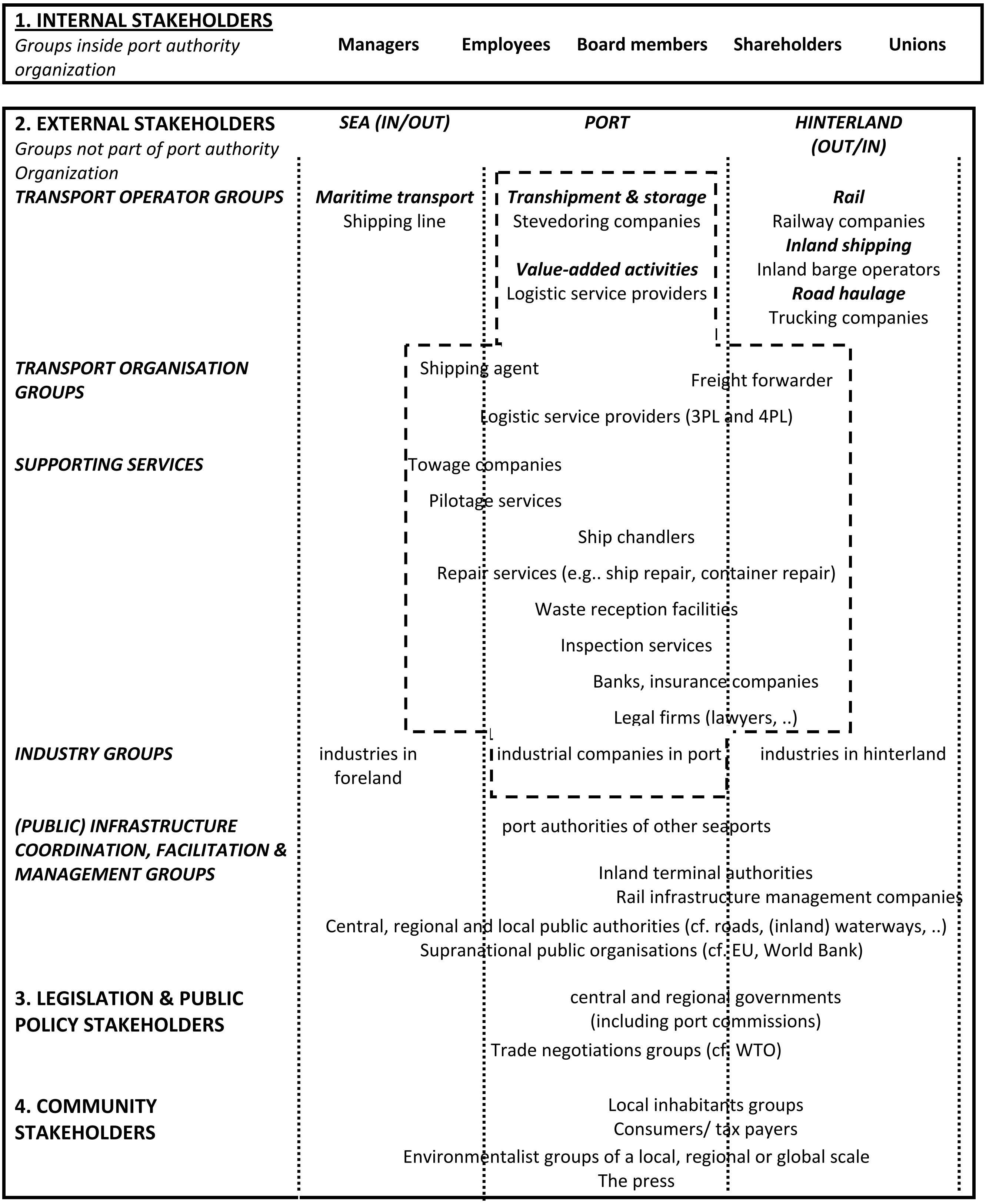

Figure 2-3: The Port Community Stakeholders (Adapted from Notteboom & Winkelmans, 2002).

2.3.3 Relationships among Port Community Actors

Unlike the two forms of interactions in the container transport network indicated in sub-section 2.3.1 (i.e. operational interactions and administrative interactions), The inter-organizational relationships among stakeholders in the broad port community are characterized by two forms of interaction: physical (i.e. related to the physical transfer of cargo) and incorporeal (Martin and Thomas, 2001). The latter type of interactions consists of contractual, supervisory or information based exchanges.

The network metaphor has been widely used in scientific literature to describe the processes, activities, and relationships in ports and terminals (Gambardella *et al.*, 1998; Frankel, 1987; Kia *et al.*, 2000; Notteboom and Winkelmans, 2002). The society or community view of a port or terminal may yield a rich description of the inner workings of the structure and provide insight to the relationships between actors or stakeholders.

The port community is regarded as a networked organization, or as a Networked Enterprise. The Networked Organization has been defined by as (Lipnack and Stamps, 1994):

> *".. where independent people and groups act as independent nodes, link across boundaries, to work together for a common purpose; it has multiple leaders, lots of voluntary links and interacting levels."*

Networked organizations are characterized by distributed control, inter-organizational business processes (i.e., business processes that cross enterprise boundaries and therefore do not belong to one enterprise), various producer–consumer supply chains, and shared information and knowledge. The main challenge is to optimize operations through co-decision, co-ordination, and even negotiation mechanisms.

In that sense, business processes in the port community are a typical Networked Enterprise, for example:

- Control is passed from one actor to the other triggered by events or information flow; for instance after berth allocation, control is passed from vessel to agent and terminal operator.
- The processes do not belong to one single enterprise. The three enterprises (vessel, agent, terminal operator), in the above mentioned example, are autonomous organizational entities.
- Producer-consumer chains; resource owners (trucks, stevedoring, etc) will typically represent such chains vs. the vessel.

As a Networked Enterprise, the port community is dynamic in nature, i.e. nodes or entities can be added or removed with minimum disturbance to the processes being active within the enterprise. This indicates the necessity of agility to sustain changing economic conditions and/or to implement new strategies.

Considering the stakeholders groups described in sub-section 2.3.2, there are several concerns that shape the relationships between and within these groups (Notteboom and Winkelmans, 2002):

- **Distributional concerns**, i.e. issues related to the distribution of costs and benefits among stakeholders, the trade-offs (e.g. between economic, ecological and the social value of ports) and the creation of win-win situations.
- **Efficiency concerns**, i.e. maximum output generation with a minimum input.
- **Behavioural concerns** e.g. related to cheating behaviour, opportunism and bounded rationality. For instance, local pressure groups often defend their local interests in such a fierce way that the individual well being of a few people is becoming an even bigger driving force than the well being of the greater community.

These aspects do not only play a role in formal contracting among stakeholders, but also in less formal situations of stakeholder interaction.

2.4 Characteristics of the Port Community Supply Chain

In this section, main characteristics of the port community supply chain are introduced. The objective of introducing such characteristics is to highlight the complexity of the port community supply chain, and to consider these characteristics in the design of the proposed agent-based middleware.

The characteristics that are introduced in this section are based on earlier studies as indicated in this Chapter and on the knowledge that we have acquired from the interviews that were carried out with maritime transport experts during this study.

The characteristics of the port community supply chain that are considered in this study are divided into two main groups. The first group includes those characteristics regarding the environment itself (i.e. the port community) while the other group includes characteristics of the members of the community. The main port community characteristics are as follow:

Group I: Characteristics of the environment itself:

- **Highly dynamic environment.** The dynamicity of the port community is reflected by the unavailability of information. The information becomes available over time. Therefore, planning of activities of the business process to transport a container always needs to be done without full knowledge of the future.
- **Interdependency of activities.** The maritime transport of containers is carried out through a distributed business process. Activities of the business process are distributed among players of the port community. Therefore, these activities become interdependent. Delays at one player easily propagate through the business process and affect the operations of other players. This interdependency could adversely affect the plans of other players especially when they do not have enough rescores (time, manpower, cost, etc) for rescheduling their activities in real-time. In addition, a very limited or absence of information sharing would add complexity to real time rescheduling.

- **Ill-structured - loosely coupled network**. Business relations between players in the port community do not show a clear hierarchy. The port community can be considered as a market in which each player is somehow free to join and to leave. Consequently, players involved in business at a certain port community might change over time.
- **Disturbances during execution.** During execution, events such as crane breakdowns which affect the operations of terminals. Another cause of disturbance is, for example, when a truck(s) allocated to pick up a container fails to arrive at the container terminal on time.

Group II: Characteristics of the members of the port community:

- **Poor Contractual Relationships.** Usually, there is no contractual relationship between members of the port community. The absence of contractual relationship means that a member cannot contractually force other members to deliver a certain service or charge for poor services.
- **Autonomy**. Each member of the port community prefers to have own control on its operations.
- **Conflict of Interest.** As discussed in Chapter 2, many different players are involved in the port community. Each member has its own objective(s) that might differ from or even conflict with objective(s) of other members.
- **Limited information sharing**. Some port community members, such as shipping agents and inland transporters, are competing with each other

rather than collaborating. Therefore, they are reluctant to share information since they believe that this might degrade there competitive advantages.

2.5 Problems in the Port community

The previous sections were dedicated to give an outline about seaports and the concept of port community. It was intended to make the reader familiar to that particular domain.

In this section, we are going to identify bottlenecks that arise from the basic business processes in the port community. We focus on two issues; coordination and disruptions, which we perceive they adversely affect the efficiency of the port community. In the following sub-sections, we discuss these two aspects, and then we identify the information requirements to deal with these issues.

In the following sub-sections, we use the terms “port community” and “port hinterland” interchangeably. This is because we perceive the port community from a supply chain perspective, as indicated in the previous sections. Therefore, the “port community” and the “hinterland” are interleaved.

2.5.1 Coordination in the Port Community

This section sheds some lights on the need for coordination in the port community. It is this need for coordination which motivates our proposed use of the agent-based middleware.

Researchers in many fields including computer science, economy, and psychology have studied the area of coordination, and they perceived coordination as *"managing the interdependencies among activities"* (Malone and Crowston, 1994). In that sense, coordination of the port community is challenging, given the number of involved actors and the complexity of interactions among them (as discussed in sub-section 2.3.2 and sub-section 2.3.3.

Generally, the hinterland transport costs are higher than the maritime-transport costs, in addition, most bottlenecks in the door-to-door chain occur in the hinterland due to, for example, congestion, inadequate rail infrastructure, and problems with the handling of barges at deep-sea terminals. Consequently, coordination in hinterland transport networks has increasingly become subject of study among scholars in port economics and management.

The need for coordination in port communities arises from the role of contemporary ports as links in the global supply chain, as indicated previously in Section 2.1. It is this role that made port competition moves from competition between ports to competition between transport chains. Accordingly, port competitiveness no longer merely considers the physical attributes of the port, such as, geographical location, physical and technical infrastructure, port performance, cost, and security.

The contemporary supply chain orientation of ports has become the relevant approach for analyzing port competitiveness. This implies that a port's competitiveness becomes increasingly dependent on external coordination and control of outside actors (i.e.

members of the port community). Therefore, the success of a port will depend on the ability to integrate the port effectively into networks of business relationships that shape supply chains. This is more determined by the ability of the ports to fully exploit synergies with other transport nodes and other players within the port community of which they are part. The synergies that can be envisaged relate to efficient capacity utilization and an efficient operational synchronization and integration (Notteboom, 2008).

Coordination in hinterland networks has been studied from an operational and technical perspective (Bontekoning, 2006). It has also been approached from a supply chain management perspective, focusing on chain configuration and integration (Panayides, 2002). The first step towards analyzing cooperation in hinterland chains was the introduction of the concept of Hinterland Access Regimes (HARs) that was defined as (de Langen and Chouly, 2004):

> *"The set of collaborative initiatives, taken by the relevant actors in the port cluster with the aim to improve the quality of the hinterland access".*

However, the concept did not provide a basis for identifying where coordination is required or what mechanisms could be introduced to enhance coordination.

Van Der Horst and De Langen (van Der Horst and De Langen, 2008) use insights from Institutional Economics, with Transaction Costs Economics playing a central role, to study coordination in hinterland networks. They come up with an analytical framework that can be used to further analyze coordination problems and evaluate coordination arrangements that can be introduced in hinterland chains.

A set of coordination problems has been developed for the container transport hinterland of the Port of Rotterdam (van Der Horst and De Langen, 2008). These problems were categorized into specific and generic coordination problems. The specific coordination problems are those that affect specific hinterland transport mode such as road, rail, and inland waterway (through barges), whilst the generic coordination problems are those that affect all hinterland transport modes. The generic problems are illustrated in Table 2-3; in particular, these are the kind of problems that we argue that our proposed agent-based middleware will facilitate.

Coordination problem	Actors involved
Insufficient information exchange of container data causes inadequate planning.	Container shipping line, container terminal operating company, forwarder, truck company, barge operator, rail operator
Insufficient planning on transporting and storing empty containers.	Container terminal operating company, rail terminal operator in hinterland, barge terminal operator in hinterland , container shipping line
Limited planning for physical and administrative inspection between Customs and Inspection authorities causes delay.	Customs, Inspection services
Insufficient information about Customs clearance of a container.	Forwarder, Customs, shipper

Table 2-3: Coordination Problems in the Container Transport Hinterland of the Port of Rotterdam; adapted from (van Der Horst and De Langen, 2008)

These problems revealed that information sharing is one of the most basic activities to realize coordination. Full information sharing, accompanied with coordinated decision-making, can result in significant performance improvements (Straub *et al.*, 2004). Merely full information sharing is not a sufficient condition for supply chain integration. Performance can be sub-optimal when each decision-maker optimizes his individual objective function. Therefore, both information sharing and coordination are deemed to be essential for the integration of the port community.

2.5.2 Disruption Management in Container Transport

In this section, we focus on operational disruptions that can be handled in a routine fashion. These disruptions presumably have impacts on the safety and security of the supply chain, as explained in the paper by Kleindorfer and Saad (Kleindorfer and Saad, 2005).

Generally, it is not surprising that business as usual is disrupted by unexpected events, such as administrative errors, and delays. Usually, a disruption results in human intervention in the information flow and administrative system to solve problems created by the disruptions. As noted previously, the maritime transport of containers is a complicated process; involving several process steps, multiple organizations, and numerous information exchanges. In that environment, disruptions may occur both in the physical and in the information flow.

An example of disruptions in maritime transport of containers is that containers that are planned for loading to the vessel are not available in the yard. Such a situation can be attributed to one of two reasons. The first reason, which is a disruption in the physical flow, is that the container might be routed to a wrong location or is delayed. The second reason, which is a disruption in the information flow, is that the container is mistakenly listed or that the container number has been mistyped. To solve such problem in practice, a lot of communication between numerous parties in the logistic chain is usually needed to find out what is wrong.

Disruption handling is usually aimed at mitigating the negative effects, but could be taken one step further in order to prevent the occurrence of disruptions. This can be realised by introducing more flexible processes through which disruptions could be handled without excessive costs.

The above mentioned disruptions have impacts on the safety and security of the port community supply chain. They introduce a potential risk for smuggling or stealing of containers or their content, therefore, they made the supply chain more vulnerable. In addition, these disruptions are a source of inefficient container handling, and this leads to delays and high waiting times.

In order to address this issue, disruption handling should be integrated into everyday operations, i.e., standard business procedures could be augmented by scenarios of how to deal with disruptions. In particular, information systems could be adjusted in such a way that they support the handling of disruptions as well as procedures for business as usual. It is this need for information systems adjustment which motivates our proposed use of a middleware as an amendment to the existing port community systems. In the next sub-section, we propose information systems requirements that we perceive essential for the port community supply chain.

2.5.3 Information and Coordination Requirements

Based on issues in containerized transport supply chains caused by the lack of information sharing, information quality, and lack of coordination; we introduce the corresponding required capabilities of information systems to address these issues.

Requirement 1: Collaborative planning

Some container information, such as pickup time, final destination, and hinterland transport mode, may change at several time points. In addition, it is not known far in advance when the container will be released from the terminal. In order to plan container stacking and transport onwards in a coordinated fashion, new arrivals of information need to be captured and incorporated into the planning system, preferably in a dynamic way. However, this requires information sharing and collaborative planning between the container terminal and the (intermodal) transport operators. The shipper may even be involved to ensure timely notification of (changes in) the final destination and preferred transport mode and route, when applicable. In order to speed up the release of the container from the terminal, also the shipping line needs to be involved. Physical, transactional, and informational flows need to be coupled. Customs could be involved to speed up the administrative process, but some uncertainty in the container selection process for further inspection needs to be accepted for security reasons.

Requirement 2: Disruption alerts

Due to inherent uncertainties in transport, e.g., congestion, delays may occur in the arrival of transport modes at the terminal. This requires capacity planning to be flexible in order to avoid any cascade of mismatches between planning and execution. As a consequence, updates of the execution status could be beneficial for several organizations in the transport chain. While providing execution data, a balance needs to be found between overloading organizations with execution data and confronting planners with sudden facts. An alerting system, adapted to the needs of the organizations involved, could support an effective communication of disruptions in planning. Otherwise, the uncertain environment would be accepted to a certain extent, and re-planning would be considered as a routine step in planning and execution of processes.

A prerequisite for the above mentioned requirements is the supply chain visibility. This visibility is realized by providing quality information (accurate, timely, and secure) about physical flows, information flows, and financial flows. The supply chain security can be enhanced by increasing the visibility of the supply chain. It will also be the basis for various collateral benefits like increased logistics efficiency, and these benefits will trigger supply chain parties to connect to inter-organizational systems such as our proposed middleware.

2.6 Summary

In this chapter we give an outline about the maritime transport domain through introducing an overview on seaports and the evolution of its role, the maritime transport process, and the concept of port community with emphases on the complexity of the interactions among the actors of the port community supply chain.

Accordingly, and with the aid of the knowledge that we acquired from interviews that were carried out with maritime transport experts during this study, we identified some characteristics of the port community supply chain. These characteristics constitute the foundations upon which the design of our proposed middleware is based.

The main issue that motivates our proposed use of agent-based approach is bottlenecks that arise from the basic business processes in the port community. Therefore, we identified two issues; coordination and disruptions. We perceive these issues adversely affect the efficiency of the port community. In this chapter, we point out that performance in the port community supply chain can be enhanced by coordination at the tactical level, and disruption management at the operational level.

In order to realize this objective, we have identified information and coordination requirements to deal with these issues, this include: collaborative planning, disruption alerts, and supply chain visibility.

In the next Chapter, we introduce the convenient information technologies that can be employed to avail the above mentioned requirements for the actors of the port community.

Chapter 3

Technical View: Processes, Agents, and Web Services

3. Technical View: Process, Agents, and Web Services

3.1 Introduction

As noted in Chapter 1, the wider aim of this thesis is to employ agent technology for effective management of port communities so that the performance of their business processes would hopefully be improved. Therefore, in this chapter we introduce technologies that have been recognized convenient to realize the aim of the thesis.

In addition, in Chapter 2, we discussed the concept of “Port Community” which we perceive as a distributed business network, dynamic, collaborative (both horizontally and vertically), and competitive (since the entities comprising the community are independent). Therefore, we infer that the management of business processes of such an environment is not straight forward. This raises a question about the role of computing, and in particular agent technology, in this regard.

In both academia and industry, three concepts are accepted as a proper mix for solving problems in business process management, and accordingly enhancing its performance. These concepts are *process* which is addressed by workflow management technology, *intelligence* which is addressed by agent technology, and *dynamic interoperability* which is addressed by service oriented technology (Grefen, 2006).

These concepts are considered during the design of our proposed agent-based framework which is introduced in Chapter 5. Therefore, we perceived that it is worth giving more insights into these concepts and technologies. In the following sections, we discuss the three aforementioned technologies, and how they are used in order to realize an acceptable effective management of business processes. In addition, their application in port communities and maritime transport in general is also introduced, if any.

Although Web Services are outside the scope of this thesis, however, it is worth to include it because of the possibility of providing the functionalities of a Port Community System (PCS) by deploying a Service Oriented Architecture (SOA); a detailed description of Port Community Systems is introduced in Chapter 4. In such architecture, the system would be based on collections of web services that allow any authorized party to make use of a transaction, a group of transactions, or a subset of transactions, extend them and integrate them into their internal systems.

3.2 Workflow Management Technology

3.2.1 Background

Workflow management technology has been around since the early nineties. Much of this technology, however, focuses on intra-organizational workflows, assuming a possibly distributed, but homogeneous workflow management system in a trusted environment. When workflow management is extended across organizational boundaries, the complexity is increased.

Due to its ability in modelling, executing, and monitoring processes; workflow management technology is one of the areas that had attracted the attention of many researchers, developers and users since the early 1990's. Workflow technology includes a set of technological solutions intended to automate or support business or work processes that are described in an explicit process model.

The term *"workflow"* is often used casually for referring to a business process, specification of a process, software that implement and automates a process, or software that simply supports the coordination and collaboration of people that implement a process. One widely accepted definition of the term *"workflow"* comes from the Workflow Management Coalition[1] (WfMC, 1999):

> *"The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules."*

The evolution of workflow as a technology has encompassed a number of different product areas, such as: Image Processing, Document Management, Electronic Mail, Groupware Applications, Transaction-based Applications, and Project Support Software (Hollingsworth, 1995).

[1] The Workflow Management Coalition (WfMC), founded in August 1993, is a non-profit, international organization of workflow vendors, users, analysts and research groups. The coalition's mission is to promote and develop the use of workflow through the establishment of standards for software terminology, interoperability and connectivity between workflow products. (http://www.wfmc.org/)

Workflow management involves everything from modelling processes up to synchronizing the activities of information systems and humans that perform the processes. One approach for realizing workflow is the implementation of Workflow Management System (WfMS). The role of the WFMS is to provide the services required for automating workflow processes. In general, a workflow is a representation of the business process in a machine-readable format. Hence, a workflow management system is defined as (WfMC, 1999):

> *"A system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications."*

Workflow Management Systems (WFMS) could be **centralized** or **distributed**; according to the way they execute and manage business processes (Khare and Rifkin, 1999). In the **centralized approach**, a single workflow engine is responsible for the control and execution of all business process instances. The external applications can be located either on the same system or on different systems. In the **distributed approach**, several workflow engines cooperate in the execution of a process instance and each one is responsible for managing part of the overall execution. One process instance can start in one engine and parts of it can be executed in another one. This is a more flexible and distributed solution. It is obvious that centralized systems have several drawbacks like load balancing and scalability problems in comparison to the distributed approach (Grefen *et al.*, 1999). However, due to the distributed execution of business processes from different workflow engines, some synchronization problems might be created.

3.2.2 Approaches for Workflow Management Systems Implementation

A number of alternative approaches for implementing WFMS have been emerged. In these approaches, different mechanisms have been employed to support the communication between the process engine and the external applications. Namely, these approaches are the **client/server**, the **message-based**, the **event-driven**, the **object-oriented**, and the **agent-based** approach.

A. The agent-based approach

The **agent-based approach** is rather a recent category of WFMS. In these systems, the execution and management of business processes is performed by a set of autonomous, intelligent agents that cooperate, in a distributed manner, to execute the process and thus, to reach a business goal (Borghoff *et al.*, 1997; Bellifernine *et al.*, 1999). This approach is described in more detail in Section 3.4.

B. The client/server approach (i.e. 2-tier systems)

This is the first widely adopted approach, and the WFMSs that are implemented using this approach are considered the first generation of the workflow management systems (Filos and Ouzounis, 2000). In these systems, the process engine and the external applications and instances use conventional client/server concepts such as TCP/IP and relational databases. The interfaces among the entities are static, programming language dependent, and network protocol specific. These systems are convenient to be deployed for business process execution and management in one organization (i.e. intra-organization), where the interfaces of the external applications are well known, centrally specified, and static. These systems are in general closed, not adaptable, and inflexible. One of the major drawbacks of these systems is the low degree of distribution and autonomy (Miller, 1996).

C. The message-based approach

In the message-based approach (or n-tier message clients) (Eder and Liebhart, 1996), the workflow engine and the external entities co-operate through exchanging messages that have a well-defined format, syntax, and semantic. Each activity within the workflow system is considered as a message client that can send and receive messages. The underlying communication system is a messaging system that is responsible for forwarding messages to the corresponding receivers (Alonso *et al.*, 1995). The communication model can be synchronous or asynchronous, and point-to-point and/or multi-point.

The message-based workflow systems have been basically deployed for intra-organizational purposes; therefore, a proprietary ASCII format is usually used to specify the message content description languages. Consequently, these systems can be characterized as closed and having limited degree of autonomy and flexibility and low level of adaptability. These drawbacks resemble those of the WFMS implemented using the client/server approach. The flexibility and openness of these systems could be improved when XML (the eXtensible Mark-up Language) is used to describe the syntax and semantic of the exchanged messages among the entities of the workflow system.

D. The event-driven approach

This approach is considered as a special type of the message-based WFMS. In these systems, different entities of the workflow system coordinate their activities through exchanging events that have specific format (Grefen, 1998). These systems depend on the underlying event management system and the syntax and semantics of event description languages. The event-driven systems cannot explicitly be used for data passing among workflow entities. They are mostly used for notification of process status, i.e. events can be used as a coordination mechanism (Grefen *et al.*, 1998).

E. The object-oriented approach

In the object-oriented approach (or n-tier systems) there is more than one workflow engine that cooperate with the external entities by deploying distributed object-oriented platforms, like Common Object Request Broker Architecture (CORBA), Remote Method Invocation (RMI), or Distributed Component Object Model (DCOM). The communication approach among the involved entities is remote procedure calls. These systems are characterized by adequate degree of flexibility and distribution. There are static and well-defined interfaces for the integration of legacy applications and systems. However, these systems do not provide the appropriate degree of autonomy, intelligence, and dynamic behaviour (Alonso *et al.*, 1995). The business process execution and management is static, pre-defined and no alterations during process execution can easily be done. Additionally, external applications and services need to implement special, static, and pre-defined interfaces, something that it is not always feasible especially in the case of different administrative domains (Bolcer and

Kaiser, 1999). Finally, as in the other approaches, most of the OO-workflow systems are deployed for intra-organizational business processes, the semantic of activities and sub-processes is well known, pre-defined, and centrally specified (Georgakopoulos, 1998).

The above discussions of various WFMS approaches reveal that all these approaches have significant benefits and drawbacks regarding their deployment in the context of inter-organizational business process execution and management. Therefore, these approaches can not be adopted for implementation in the port community which is perceived as a complex networked enterprise as discussed in Chapter 2.

In this regard, a clear need for distributed, inter-organizational workflow management approach has been emerged; therefore, this approach is discussed in sub-section 3.2.4. We first give a background about the inter-organizational information Systems (IOIS) in sub-section 3.2.3. We see it worthwhile since the IOIS is relevant to the port community systems which are considered as an industry IOIS as will be discussed in Chapter 4.

3.2.3 The Inter-Organizational Information Systems (IOIS)

Generally, Inter-Organizational Systems are automated information and communication technology-based systems that surpass legal enterprise boundaries (Kumar and van Dissel, 1996). IOIS has emerged in the 1980s as a strategic and productivity tool for companies implemented for efficient exchange of business transactions (Barrett and Konsynski, 1982; Cash and Konsynski, 1985; Johnston and

Carrico, 1988). IOISs facilitate the exchange of information electronically across organizational boundaries and provide both processing capabilities and communication links.

The first attempts to realize IOIS was in the form of using the traditional Electronic Data Interchange (EDI). EDI is defined as (Premkumar *et al.*, 1997):

> *"EDI is the direct computer-to-computer communication between an organization and its trading partners of business documents and information in a machine-readable, structured format that permits data to be processed by the receiver without re-keying."*

In such set up, the IOIS is controlled by individual companies, who establish direct links with their partners (Bakos, 1991). These systems had a point-to-point nature and were rather inflexible. Though EDI was a significant progress towards the direction of inter-organizational business automation, getting these systems to work in the larger supply chain context was virtually impossible. Problems regarding the standard format of EDI messages, the insecure open transport networks, and the rather restricted context of EDI (only focus on electronic commerce) made EDI not an attractive solution for inter-organizational workflow management (Bolcer and Kaiser, 1999; Gibon *et al.*, 1999).

In an alternative scenario, a group of firms of a business sector or industry collaborate to setup up an industry IOIS (Damsgaard and Lyytinen, 2001; Hock-Hai *et al.*, 1997; Kambil and van Heck, 2002; van Baalen *et al.*, 2000), which will be used by these firms and maybe others in the industry. In these industry IOISs, firms do not have to implement direct links with each of their partners; rather, they establish a unique link with the IOIS. Port Community Systems, which are the focus of this thesis, are a real life example of an industry IOIS.

Industry IOISs qualify as business sector information infrastructures (Ciborra, 2000; Hanseth *et al.*, 1996; Monteiro and Hanseth, 1995; Star and Ruhlender, 1996); which is defined as an *"evolving, shared, open and heterogeneous installed base"* (Hanseth, 2000). Information infrastructures may be classified into: corporate (i.e. ERP systems), business sector (i.e. EDI networks) and universal (i.e. the Internet) (Hanseth and Lyytinen, 2006).

An industry IOIS is shared in the sense that it is set up, organized and used by firms in the same industry. It evolves as new companies integrate with it or as new types of exchanges and functionalities become available through the IOIS. It is heterogeneous as it encompasses multiple technologies as well as non-technological elements (social, organizational, institutional, etc.) that are necessary to sustain and operate the IOIS (Hanseth and Lyytinen, 2006). Finally, an IOIS usually embeds and supports a data and process standard, which defines the syntax, semantics and pragmatics of their transactions, and that is built by the same industry actors (Markus *et al.*, 2006).

Although a second generation of IOS technology provided new concepts, such as central planning hubs, it remained hard to establish working systems due to the different systems and procedures of all the supply chain actors that had to connect to the hub. Two other drawbacks of the centralized system appeared to be: Firstly, it is difficult to encapsulate all the required information that should be ideally needed for decision making – see for example (Sridharan *et al.*, 2005), and secondly optimization of such complex models needs a large amount of processing time – hence the fact that an optimization run of a deployed ERP system within a large company could easily take an entire night.

Establishing a centralized planning system that coordinates all the activities of the different (and independent) parties involved is not easy to achieve (For example in the maritime community, the coordination of barges with container terminals). The reasons for this are, amongst others, players feel a lack of control about their own operations and the central party probably makes decisions beneficial for one player, but are costly for another, which is hard to justify. Also, sharing gains obtained by the system is a difficult issue that may hinder smooth implementation and use. A number of researches on the critical success factors for inter-organizational system implementation derive to similar conclusions (Lee and Lin, 2005): some of the major influencing factors are strong management beliefs and achieved high-levels of trust between partners.

3.2.4 Inter-Organization Workflow Management Systems

AS noted in sub-section 3.2.2, traditional WFMS systems are able to support workflows inside one organization; however, they do have certain problems dealing with workflow crossing organizational boundaries (Wil and Aalst, 1999). One of the main issues is the limited autonomy and flexibility that they have. Traditional WFMS execute upon well-defined business processes specified in a specification language that the system could understand. In addition, remote invocation of business processes, provided by different organizations, should follow access control, authorization and contract checks. Current workflow systems do not provide such mechanisms (Klingemann and Aberer, 1999). Finally, in traditional workflow systems, shared business processes are being specified statically in relation to remote processes, i.e. the organization partners that would provide them are specified statically (Geppert and Tombros, 1998).

This approach is not suitable for situations where organizations that can provide parts of the shared business process are not known in advance. On the contrary, the remote organizations are being selected dynamically after negotiation and during the business process execution. This means that for the same business process specification different instances might exist. For every instance a set of different partners might be selected according to the needs and requirements of the process (Ouzounis and Tschammer, 1999).

Several developments in the direction of inter-organizational workflow management have been carried out; some are summarized in the following.

The Workflow on Intelligent Distributed database Environment (**WIDE**)[2] project (Grefen *et al.*, 1999) had a main objective which is to extend the technology of distributed and active databases in order to provide added value to advanced application-oriented software products implementing workflow techniques.

[2] http://wwwhome.cs.utwente.nl/~wide/wide.html

Specifically, the main goals of WIDE are to define an advanced conceptual model for describing both the flow of activities, and the organizational environment in which these activities are performed. Particular emphasis has been put on specifying exceptions in the normal flow of activities, and on supporting different types of exceptions and abnormal situations. Additionally, special mechanisms have been developed to provide flexible workflow management through advanced database systems including active database technology and advanced transaction management in a distributed environment with long running transactions (Grefen, 1998). WIDE provide tightly integrated features concerned with advanced transactions, by supporting distributed and asynchronous processing in the context of long-running and co-operative activities, and with reactive processing, by supporting a rich event language, as well as, enhanced, flexible coupling to transactions (Grefen *et al.*, 1999). The WIDE project has a very clear focus towards distributed, intra-domain workflow management systems and deployment and integration of conventional distributed components.

The **WISE** project (Alonso *et al.*, 1999) (Workflow based Internet SErvices), aims at designing, building, and testing a viable infrastructure for distributed workflow based applications over the Internet. Such infrastructure include an Internet based workflow engine, acting as the underlying distributed operating system, that controls the execution of distributed applications, a set of brokers enabling the interaction with already existing systems, that are to be used as building blocks, and tools for programming in the large to allow final users to configure and develop distributed applications. The project aims to solve the limitations of workflow systems and to extend their applicability to the Internet by providing a broker based platform for interacting with heterogeneous, stand-alone applications and implementing transactional mechanisms as a way to provide execution guarantees (Alonso *et al.*, 1999). These solutions will be integrated into a robust, reliable, and scalable execution engine able to provide a distributed control over the execution of distributed applications over the Internet. The WISE project follows a centralized approach where the Internet based workflow engine plays the role of the coordinator and manager of the shared business processes.

3.3 Agent Technology

3.3.1 Agents and their Characteristics

There is no universally accepted definition of the term agent, and indeed there is much ongoing debate and controversy on this subject (Jennings, 2000; Wooldridge, 2005). In general, there is an agreement upon what agents are capable to, but it has proven more difficult to find a commonly accepted specific definition. In the following, we introduce various definitions of the term agent that we encounter in the literature.

An agent can be defined as a physical or abstract entity that can act in its environment (Ferber, 1999). This entity possesses a partial representation of the environment and can communicate with other agents. Its behaviour is determined by its knowledge, its perception, and its interaction with other agents.

According to Maes (Maes, 1995), autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realise a set of goals or tasks for which they are designed.

A widely accepted working definition has defined the agent as (Wooldridge and Jennings, 1995): is a computer system that is situated in some environment, and that is capable of flexible and autonomous action in this environment in order to meet its design objectives. By flexible we mean that the system must be responsive, proactive, and social.

A minimal common definition of an agent regards it as a physical or virtual entity which (Ferber, 1999):

- is capable of acting in an environment,
- can communicate directly with other agents,
- is driven by a set of tendencies or goals (in the form of individual objectives or of a satisfaction/survival function which it tries to optimise),
- possesses resources of its own,
- is capable of perceiving its environment (but to a limited extent),
- has only a partial representation of this environment (and perhaps none at all),
- possesses skills and can offer services,

- may be able to reproduce itself,
- Whose behaviour tends towards satisfying its objectives, taking account of the resources and skills available to it and depending on its perception, its representation, and the communication it receives.

The various definitions presented in the literature identified the key properties that characterise an intelligent agent (Wooldridge and Jennings, 1995; Franklin and Gasser, 1997; Wooldridge, 1998; Jennings *et al.*, 1998; Ferber, 1999; Wooldridge, 2005):

- Autonomy: agents operate without the direct intervention of humans or others, but have some kind of control over their actions and internal state using a set of tendencies. Tendencies are individual goals to be achieved by the agent.
- Social ability: agents cooperate, negotiate, and communicate with other agents.
- Reactivity: agents perceive their environment, and respond in a timely fashion to changes that occur in it in order to satisfy their design objectives.
- Pro-activity: agents do not simply act in response to their environment; they are able to exhibit goal-directed behaviour by taking the initiative. Agents are capable of handling complex and high-level tasks. The decision as to how such a task is best split up into smaller sub-tasks, and in which order, and action, the sub-tasks are best performed, should be made by the agent itself.
- Temporal continuity: agents are continuously running processes.

- Mobility: an agent has the ability to transport itself from one computer to another, retaining its current state.
- Learning: agents are able to learn and adapt themselves to fit their environment.

Two classes of agents have been defined in the literature: reactive agents and cognitive agents (Wooldridge and Jennings, 1994; Wooldridge and Jennings, 1995; Nwana, 1996; Jennings *et al.*, 1998; Wooldridge, 1998; Ferber, 1999; Wooldridge, 2005).

Cognitive agents possess internal representation models of the world and expertise, have goals and plans, are capable of reasoning, and can cooperate, coordinate, negotiate, and communicate with other agents. Reactive agents in contrast do not have any internal symbolic models of their environment, and they act using stimulus/response type behaviour by responding to the present state of the environment in which they are embedded. Broadly speaking a reactive architecture is a collection of task accomplishing behaviours (Jennings *et al.*, 1998). Each behaviour is a finite state machine that continually maps perceptual input to action output. This mapping is achieved via (situation action) rules, which simply determine an action to perform on the basis of the agent's current state.

Several theoretical architectures have been defined in literature to describe the basic structure of cognitive agents. The most common modules of the internal architecture of an agent are: perception, execution, self-knowledge, acquaintance knowledge or social knowledge, domain knowledge, reasoning, learning, cooperation, and communication (Wooldridge and Jennings, 1994; Jennings *et al.*, 1998; Ferber, 1999; Weiss, 1999; Wooldridge, 2005). These are described in more detail bellow.

- Perception: the perception module is one of an agent's interfaces to its environment. Commonly the perception module obtains signals from the agent's sensors. But in most architectures this module is integrated into the communication interface.
- Self-knowledge: An agent's self-knowledge contains the agent's knowledge about itself, including its physical state, location and skills, etc.
- Domain knowledge: this knowledge concerns the problem-solving domain and environment. Usually this module contains the description of the problems to be solved.
- Social knowledge (acquaintance knowledge): this knowledge, also called beliefs, is the knowledge used by the agent to interact with its acquaintances. It describes the skills and identity of acquaintance agents. The knowledge is used by the agents to identify other agents with whom it is useful to interact, and wish to determine which agents have the skills necessary to perform a particular task. This knowledge must indeed model the role, competence, localisation (address of an agent), goals, plans, and the resources of these dealings to be able to interact with them.
- Learning: an agent working in a dynamic environment needs to adapt to changes in that environment. It needs to learn in order to update its

knowledge about its environment, other agents, and the problems to be solved.

- Reasoning: This is the decision making process which decides to act on the basis of the information it receives, and in accordance with its own objectives to achieve its goals.

- Communication: This is the interface used by the agent to communicate with its environment and with other agents.

- Cooperation: defines the models of coordination and cooperation to interact with other agents in order to perform tasks for other agents.

3.3.2 Multi-Agent Systems

Various definitions have been proposed for the term multi-agent system (MAS). Durfee *et al.* (Durfee *et al.*, 1989) defined a MAS as a loosely coupled network of problem solvers that work together to solve problems that are beyond the individual capabilities or knowledge of each problem solver. Another definition regards MAS as a system composed of a population of autonomous agents, which interact with each other to reach common objectives, while simultaneously each agent pursues individual objectives (Ferber, 1999). According to Oliveira *et al.* (Oliveira *et al.*, 1999) a MAS is defined as a collection of possibly heterogeneous, computational entities, having their own problem-solving capabilities and which are able to interact in order to reach an overall goal.

The main characteristics of MAS are (Jennings *et al.*, 1998):

- Each agent has incomplete information, or capabilities for solving the problem, thus each agent has a limited viewpoint;
- There is no global system control;
- Data is decentralised;
- Computation is asynchronous.

In order for MAS to solve common problems coherently, the agents must cooperate, coordinate, and communicate amongst themselves. Cooperation, coordination, and communication are central to MAS. Agents need to interact with other agents to achieve their objectives either because they do not have sufficient capabilities or resources to complete their problem solving alone, or because there are interdependencies between the agents that follow from being situated in a common environment (Faratin *et al.*, 1998). These interactions can vary from simple information interchanges to requests for particular actions to be performed, cooperation (working together to achieve a common objective), coordination (arranging for related activities to be performed in a coherent manner), and negotiation (a process by which a group of agents come to a mutually acceptable agreement on some matter) (Jennings *et al.*, 2001).

3.3.3 Agent Technology Applications in Maritime Transport

In this section, we introduce an overview of the research efforts made in the agent technology domain focusing on a logistical perspective (i.e. focusing on transportation of freight). Agent technology approaches that are employed to supply-chain management are not considered in this survey.

Scientific researches in the agent technology domain provide techniques, methods, and tools that have potential applications in the area of transportation management at large. However, very few researches have been carried out focusing on seaports or port community management. In a similar survey (Davidsson *et al.*, 2005), they declared that agent technology applications, and in particular in sea transportation are scarce and most of the publications have focused on the alignment of activities in a container terminal; for example (Itmi *et al.*, 1995; Rebollo *et al.*, 1999; Henesey, 2006).

These researches deal with applications of agents aiming at the increase of the efficiency of the container terminal operations. Specifically, they tend to focus on the marine-side interface whilst disregarding the other processes in the terminal that determine overall terminal performance. The main reasons mentioned in the papers for adopting an agent-based approach are: facilitates distributed control and ability to model complex problems.

Among the literature on agent-based approaches in transport logistics, we found only one paper (Yi *et al.*, 2002) that deals with a similar domain to that which we consider in this thesis (i.e. the port community supply chain management). They focus on presenting a modelling and simulation methodology in the harbour supply chain that satisfies the overall operations and logistics.

Yi et al. (2002) employed agent technology for modelling and analysis of the dynamics of business processes and interaction between business entities in a harbour supply chain. They developed and implemented a multi-agent system and simulation model to (1) describe a harbour supply chain network regarding its components, behaviours, and interaction, (2) represent business entities as agents and the involved information and material flows with proposed coordination method for collaborating among agents during simulation, and (3) simulate a harbour supply chain for the choice of the most effective strategic and operational policies in smoothing the variations in the supply chain.

They developed the simulation model for berth allocation and crane assignment policies. The berth allocation policies simulate the movement of the ship to the berth and assignment of the ship to the berth with certain rules. The crane assignment policies simulate the assignment of the cranes to a ship at the berth based on a number of rules. To measure the supply chain performance, they used order fulfilment rate and on-time delivery as an output performance measure, volume and delivery flexibility as a flexibility performance measure, and resource utilization level as a resource performance measure.

Though they deal with a similar domain to that which we consider in this thesis; our work differs in a number of ways. Firstly, they focus on the operations in the harbour supply chain (inside the container terminal) while we focus on a wider scope that covers the business process of the port community as discussed in Chapter 2 of this thesis. Secondly, our main focus is to overcome the limitations and shortcomings in the existing PCSs (Port Community Systems) as discussed in Chapter 4 of this thesis; an aspect that is not covered at all in the work of Yi *et al.* (2002). Thirdly, they focus on employing agent technology for modelling and simulation in the harbour supply chain while we focus on employing agent technology to provide a solution for facilitating coordination in a port community supply chain. The framework of the proposed agent-based solution is introduced in Chapter 5 of this thesis. Nevertheless, we adopt the use of MAS (Multi-Agent Systems) to develop a simulation environment to evaluate the effectiveness of the proposed solution as discussed Chapter 7 of this thesis.

Furthermore, the literature survey shows that agent technology has been applied to a number of various problem areas within transport logistics. These agent approaches deal with problems such as planning and scheduling, fleet management, transport scheduling, traffic management, and traffic control; therefore, these approaches were distributed and complex. Contributions in this field are, for example, (Bürckert *et al.*, 2000; Lee *et al.*, 2002; Thurston and Hu, 2002; Franz *et al.*, 2007). Unfortunately, the majority of the papers reviewed do not state clearly the type of agent approaches used or how their agents are able to communicate and make decisions.

The majority of the surveyed publications introduce merely a conceptual level of an agent model, and in some cases, they describe the expected performance of the model without presenting experimental results. Most of the surveyed agent-based approaches in the maritime industry are not evaluated properly and comparisons with existing techniques are rare. Examples of publications that apply a quantitative evaluation of their multi-agent model are (Henesey *et al.*, 2009; Mes *et al.*, 2007; Lokuge and Alahakoon, 2007; Mes *et al.*, 2008).

An example for the application of agent-technology in practice is the Approach project. They employ agent-technology for the coordination of loading and unloading of containers by barges in the Port of Rotterdam (Douma *et al.*, 2006; Douma *et al.*, 2009). The planning and coordination of the rotation of the barges in the Port of Rotterdam is a complex task. Barge operators have to make reservations with eight different terminal operators on average; they, in turn, need to plan the loading and unloading processes of the ocean going vessels. Normally, this planning process relies completely on telephone and fax communication. During the execution, delays and changes in the number of containers to be loaded and unloaded will affect the schedules of the rotation. Consequently, this creates inefficient rotations in the port and causes much frustration with each of the parties.

Several attempts to improve the coordination process by means of a central planning unit or central information systems failed. To solve this problem, Dauma *et al.* (2006, 2009) used a multi-agent system in which each barge and each terminal is represented by an agent. The agents negotiate to define the terminal visits and rotation schedules that are acceptable for all parties. The system uses all the available information and supports the autonomy of each barge and terminal operator. The main advantage of this approach is that the process of coordination become much faster and many more negotiation rounds can take place before and during the execution of barge rotations.

In conclusion, although we find very few researches that focus on seaports, and a single paper related to port supply chain management; we have identified a number of useful features of the current state of agent-based approaches in maritime transport logistics as follow:

- A number of various approaches have been suggested and investigated.
- Several logistics problems that have been studied possess characteristics that are closely match those of an ideal agent technology application.
- Agent technology seems to have significant contribution to the advancement of the domain of logistics.
- Few practical experiments have been performed to assess the proposed agent-based approaches.
- The deployed agent-based systems are rarely found.
- The suggested agent-based approaches are often not evaluated properly and comparisons with existing techniques and systems are rare. Accordingly, not only qualitative approaches but also quantitative are required for the assessment of agent-based approaches. Qualitative approaches are useful to explain the benefits and limitations of agent technology compared to the

existing solutions, whilst quantitative approaches evaluate these solutions based on experiments.

3.4 Agent-based workflow management systems

Using agents to perform workflow management functionalities is an emerging field where researchers are exploring the ability of agents to improve process integration, interoperability, reusability and adaptability (Huang *et al.*, 2006; Trappey *et al.*, 2006). Many researchers believe that agent technology can provide coordinative architectures for integrating multiple heterogeneous workflow system (Kuo, 2007).

An agent-based WfMS is a set of software agents that manage and coordinate the flow of work defined by a business process (Odgers *et al.*, 1999). From this viewpoint, an agent-based system is a distributed and collaborative system rather than a stand-alone knowledge-based system (Liu, 1998). In these systems, the autonomous agents execute and manage either parts of processes or specific tasks of processes. In any case, the autonomous agents communicate through special message exchanges utilizing special coordination protocols, like FIPA-ACL (FIPA Agent Communication Language) or KQML (Knowledge Query and Manipulation Language). The content and semantic of the messages are being described in open, globally specified ontologies that the agents can understand.

Many researchers investigated the effectiveness of agent technology to provide flexible, distributed, and intelligent solutions to WfMS. An overview of some research efforts made in this domain is illustrated in Table 3-1 .

Contribution
Combining developments in autonomous agent technology and distributed computing platforms to develop an agent-based process management system (O'Brien and Wiegand, 1998).
Application of agent technology in WfMS to reduce business collaborative cost and to solve the coordination problems between WfMS (Oliveira *et al.*, 1999).
Integration multi-agent technology to WfMS to facilitate teamwork in collaborative product design (Huang *et al.*, 2000).
Convert user-defined UML process to workflow by using component-based agent architecture as a middleware (Blake, 2001).
Use of intelligent multi-agents to monitor the dynamic nature of transaction processes in B2B ecommerce workflow (Xu, 2002).
Defining an agent-based mechanism to model, control and manage the cognitive flow process to improve the problem solving ability of a team (Zhuge, 2003).
Proposing a novel workflow monitoring approach based on intelligent agents to perform flexible monitoring tasks in an autonomous and collaborative way in securities trading (Wang *et al.*, 2005).
Embedding of autonomous agents in a workflow-based distributed systems infrastructure to support design activities in an industrial context with a case study (Madhusudan, 2005).
Development of a prototype of agent-based intelligent workflow system for product design collaboration in a distributed network environment (Huang *et al.*, 2006).
Development of an agent-based workflow model to serve inter-enterprise collaboration by applying web service-based technology for messaging, service description, and workflow enactment (Wang *et al.*, 2006).

Table 3-1: Research Efforts in Agent-based Workflow Management Systems.

3.5 Web Service Technology

Although the Web Service (WS) technology is outside the scope of this thesis, however, it is worth to include because of the possibility of providing the functionalities of a Port Community System (PCS) (See Chapter 5) by deploying a Service Oriented Architecture (SOA). In the following sub-sections, we give an overview on Web Services and their potential use in ports.

3.5.1 Overview

The trend in software development was moving from object oriented models and then distributed component object models, such as Common Object Request Broker Architecture (CORBA)[3] and Component Object Model (COM)[4], which finally led to Service Oriented Computing (SOC) (Curbera *et al.*, 2003). On the other hand, several advancements in communication systems had direct implications on the underling network of the software applications which finally emerged as Web-based distributed systems. Web Services (WSs) stem from these two evolutions by combining SOC with Web accessibility.

Web Services (WS) are considered as the leading Internet based technology as a perfect implementation of SOC (Casati *et al.*, 2003; Curbera *et al.*, 2003). Web Services represent the most desired solution to increase the interoperability and leverage

[3] CORBA is an architectural specification by the Object Management Group (OMG) (http://www.omg.org)

[4] COM is a component standard developed by Microsoft.

collaboration in Business-to-Customer (B2C) or Business-to-Business (B2B) applications (Hogg *et al.*, 2004).

The main idea behind service-based systems is that applications can be assembled from a collection of network-enabled services. According to the application domain, the functionality of services can be quite diverse, from very simple, e.g. the functionality for currency conversion, or very complex, e.g. invoking a complex business applications. These services are basically modular software applications that publish one or more methods that can be called by client applications through standard interfaces. In that sense, a Web Service (WS) is a self-contained encapsulated piece of software, with a well-defined interface, that provides a set of functionalities to businesses or individuals and is made available on the Web. A formal definition of WSs introduced by the Web Services Architecture Working Group of the World Wide Web Consortium (W3C) (W3C, 2002) states that:

"A Web Service is a software application identified by a URI, whose interfaces and binding are capable of being defined, described and discovered by XML artefacts and supports direct interactions with other software applications using XML-based messages via Internet-based protocols".

What makes Web Services attractive is the ability to discover a Web Service that fulfil users' needs, negotiate service contracts and have the services delivered where and when the users need them. By letting applications access data across different devices, operating systems and locations, Web Services bring the real-time enterprise a step nearer.

The conceptual architecture of Web Services is illustrated in Figure 3-1 (Kreger, 2001). It is based upon the interaction of three roles; namely, Service Provider, Service Registry and Service Requestor. These roles interact through operations that involve the Publish, Find and Bind operations. A complete service description is published in standard-based searchable repository (services registry). The service description contains the details of the interface and implementation of the service. The service description comprises information about the operations provided by a service, input and output parameters, as well as binding information and network location. Users or Client applications search for required services in the services registry. If a desired service is found, then the service can be invoked.

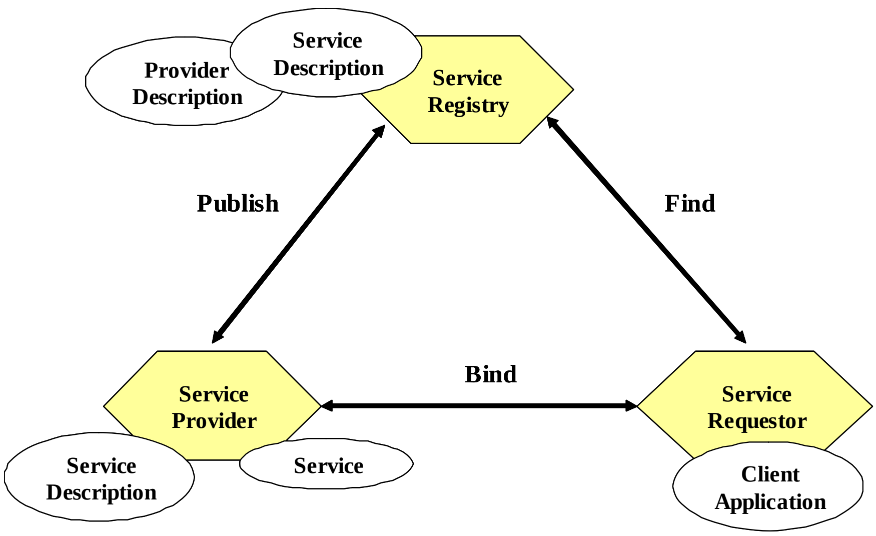

Figure 3-1: The conceptual architecture of Web Services (Kreger, 2001).

Web services operations (i.e. publish, describe, and bind) are based on specific standards in order to ensure interoperability. A number of standards committees; for example, the World Wide Web Consortium (W3C)[5] and the Organization for the Advancement of Structured Information Standards (OASIS)[6]; are involved in publishing and updating standards to meet the growing demands of Web services. Figure 3-2 depicts some of the current WSs standards as organized in a stack according to their use. The key standards for WSs include:

- Web Services Definition Language (WSDL)
- Simple Object Access Protocol (SOAP)
- Business Process Execution Language (BEPL)
- Universal Description Discovery and Integration (UDDI)
- Web Service Choreography Interface (WSCI)
- Blocks Extensible Exchange Protocol (BEEP)
- Web Services Addressing (WS-Addressing)
- Security Assertion Mark-up Language (SAML)
- Web Services Distributed Management (WSDM)
- Web Services Trust Language (WS-Trust)

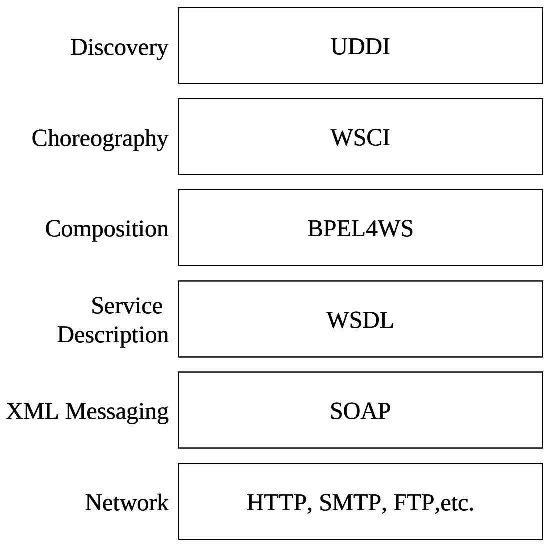

Figure 3-2: The Web Services Standards Stack.

[5] www.w3.org
[6] www.oasis-open.org

Without going into much detail of the component of the stack, the intention here is to show that the Web Services architecture is based on different de facto standards (XML, SOAP, WSDL, and UDDI). It is not essential that these standards depend on each other; however, they complement each other to form a complete Web Services concept.

To this end, we have established a generic background about the concept of Web Services. In the next sub-section, we shed some lights on the research efforts towards employing WSs in maritime transport port communities.

3.5.2 Web Services Deployment in Port Communities

Progressive port systems are increasingly exploring the Internet as the key medium for use with WSs and SOA including the XML based ebXML framework and standards proposed by the W3C.

Web Services have been introduced as one of the advanced technologies that will facilitate provisioning of widespread, sophisticated and interoperable port e-service (electronic services), and this will give ports the opportunity to offer similar services (Marianos *et al.*, 2007). In particular, they introduced the Semantic Web Services Framework as technological framework that matches the description of port e-services since it adds semantics and improve the e-port services expandability and interoperability.

Based on an exploratory research covering 20 ports worldwide, they introduced a categorization of port e-services that are provided by these modern port organizations. Accordingly, they devised taxonomy for systematic identification, assessment and selection of individual port e-services that should be implemented by contemporary and emergent port organizations.

On the other hand, the port community is considered as target enterprise for the deployment of SOA (Service Oriented Architecture (Baltzersen *et al.*, 2009). This does not mean that each system within that community must be based on SOA. However, all parties involved in the community must develop interfaces to expose business logic as services within their legacy systems, and they must also provide mechanisms to consume these services. Accordingly, the EFFORTS[7] methodology has been described which supports a *process model based approach* to develop interoperable ICT systems in Port Communities (Baltzersen *et al.*, 2009).

The EFFORTS methodology has been developed within the integrated European project EFFORTS (Effective Operations in Ports)[8]. It has been proposed as a solution for the challenges that exist in port communities. The main challenge in port communities is its extremely heterogeneous environments which include several actors, stakeholders, and systems to handle information exchange and interoperability. To add complexity to this, changes in national and international legislations, changes in customer requirements on tracking and tracing of goods, and changes in port organization put

[7] www.efforts-project.org

[8] http://www.efforts-project.org/

new requirements on the development and maintenance of interoperable ICT systems in port communities. The objective of obtaining enhanced interoperability within a port community is to provide an efficient way of exchanging information.

The EFFORTS methodology was developed in terms of a common port process map and a methodology for specification, implementation, and validation of interoperable port systems. In addition, a tool has been developed in Enterprise Architect[9] that implements the methodology, and an ICT handbook describing how to use the EFFORTS methodology and the tool has been developed.

An advantage of the EFFORTS Methodology is that a large set of standard business cases and port processes (e.g. manifest distribution and berth allocation) is included in the EFFORTS Toolbox as actors, use cases, activities, and classes, messages, and data elements. These elements can be used directly as-is when specifying the ICT system , or they can be augmented and further changed to suit specific requirements for the actual port community.

3.6 Summary

Port communities are heterogeneous environment with numerous different actors and systems. Supporting such dynamic business process networks with a qualified ICT solution requires a combination of technologies.

[9] http://www.sparxsystems.com.au/

In this Chapter, we introduced three of these technologies that are widely accepted as appropriate combination for solving problems in business process management. These technologies are the workflow management technology, agent technology, and service oriented technology. We gave an overview on each technology emphasizing its applications in port communities or in maritime transport at large.

We introduced the Web Services technology though it is out side the scope of this thesis, and this because of its potential use for providing the functionalities of a Port Community System (PCS). This is obvious from interest of the European project EFFORTS (Effective Operations in Ports)[10] as described in this Chapter. However, we did not find any evidence of using WSs in the existing port community system that we have examined; the issue of port community systems will be discussed in Chapter 4.

In this Chapter, we introduced the workflow management technology which is essential for business process management. We give emphasis to different approaches to inter-organisational workflow management since it is appropriate for the port community environment. In order to improve process integration, interoperability, reusability and adaptability; agent technology is incorporated to perform workflow management functionalities. Accordingly, agent-based management systems have emerged.

[10] http://www.efforts-project.org/

Regarding agent technology, we give an overview covering definitions, capabilities, and characteristics of agents and the multi agent systems. The use of agent technology in the maritime transport is introduced through an overview of the research efforts made in the agent technology domain focusing on a logistical perspective (i.e. focusing on transportation of freight).

In the literature, very few researches have been carried out focusing on seaports or port community management most of the publications have focused on the alignment of activities in a container terminal. These researches deal with applications of agents aiming at the increase the efficiency of the container terminal operations. Among the literature on agent-based approaches in transport logistics, we have found a single paper that focus on the same domain that we consider in this thesis (i.e. the port community supply chain management). However, we perceived that agent technology seems to have significant contribution to the advancement of the domain of logistics.

To this end, in Chapter 2 and in this Chapter, we have taken a look at the business and technical perspectives of port community as a dynamic business process networks. In the next Chapter, we will discuss how port communities employ Information Technology to manage the business activities. This step is essential in this research programme in order to identify how to employ the technologies described above to realize the main objective of the thesis (i.e. to improve the business process in port communities).

Chapter 4

Port Community Systems: An Analytical Perspective

4. Port Community Systems: An Analytical Perspective

In this chapter, we give more insight into the functionalities and services provided by Port Community Systems (PCSs) in practice. We assess the capabilities of the existing PCSs with a particular focus on their functionalities and benefits related to the port community supply chain. Specifically, the aim of this chapter is to answer a fundamental research question about the capabilities of existing PCSs to manage and improve the overall business process of the port community supply chain; such as capabilities for collaborative planning, disruption management, and supply chain visibility that we have suggested in Chapter 2.

This chapter start with an overview on the use of IT in ports and in particular the port community systems. Then, various definitions of the “port community system” and its evolution are discussed. To give more insight on how researchers and practitioners deal with issues related to PCSs, we introduce a literature review on PCSs. Finally, the chapter end with the analysis of the existing PCSs in a number of advanced ports.

4.1 Introduction

Recently, the international port logistics environment has changed tremendously due to the increase in the volume of international trade, the changing customer demands, and the developments in information technology and new handling equipments. Advanced ports around the world are eager to ensure their strategic position in the global supply chain. Therefore, ports are investing in infrastructure and improving the way they perform the operations. These investments include construction of new deep berths or deepening the existing ones, the use of advanced technologies for container handling and terminal automation to speed up the operations, and the application of ICT to improve the efficiency of the container handling process.

The construction of new terminals or new berths would require a long period of time and a huge amount of investment, therefore, the intelligent use of ICT is a critical area for port planners to improve port operations (Bagchi and Paik, 2001). Currently, ports extensively employ ICT in their operations, and information systems are used in almost every port with various levels of advancement. The term *"port information system"* is used for every kind of information technology – hardware or software – that is used in port operations. However, these information systems can be categorised as port management systems, container terminal operating systems, and port community systems.

The port management information systems are the systems that are specifically satisfy the requirements of port authorities to monitor and control the overall port activities, and to receive official declaration of vessel arrivals or application for port facilities utilization. Those systems are also referred to as port administration support systems.

On the other hand, the container terminal operating systems are defined as *"Computer systems available for organizing the container terminal itself"* (Jeffrey, 1999). Typically, it should provide the functions to manage the flow of containers through the terminal, to plan loading/unloading schedules and yard transfer operations, to process the containers transported into the terminal by rail or road, and to notify shipping companies and trucking companies about the locations of containers (Choi *et al.*, 2003). Modern terminal operating systems applications use artificial intelligence to coordinate the operations within the terminal in order to increase its overall efficiency.

The port community systems (PCSs), which are the focus of this study, are being implemented in ports in order to reduce paperwork and to facilitate the information flow related to port operations and cargo clearance. The PCSs are considered as the industry Inter-Organisational Information System (IOIS) that has the capability to bind together multiple information systems operated by competing and cooperating firms that make up the port community. It links the activities of the firms operating in the port community. Various definitions of PCSs are introduced in Section 4.2 below. In the following, we introduce the rationale that raises the need for PCSs.

As discussed in Chapter 2, the primary consumers of port services are the firms and organizations that are involved in the movement and storage of goods. This collection of private and public entities creates a complex web of interdependent relationships to conduct a great deal of intra-community business. Accordingly, it is obvious that the information assets of ports are heterogeneously distributed across individual organizations.

The main feature of that environment is the exchange of information related to consignments, payments, and business transactions. The bottleneck in ports is that these flows of information increase rapidly in terms of volume and complexity, and it take place using a wide variety of disparate business systems and processes, databases and methods of exchange. A further problem is the fact that the documents often travel physically together with the cargo, due to legal requirements. This means that these documents cannot be processed until the cargo arrives.

In this regard, PCSs promise to overcome the problem of extensive information exchange, and to integrate operational data flow among the different entities so that partners will have the access to relevant data and systems to process business transactions. Using PCSs, the flow of data and information is expected, theoretically, to stay ahead of the flow of goods but in practice this is the key problem associated with port management (Hakim *et al.*, 2005).

From another perspective, ports are no longer handling just cargo, but they are becoming *"information handlers"* (Henesey, 2002). The embedding of ports in the global supply chain, as discussed in Chapter 2, obliged ports to act as an *"information hub"* in the global chain. Information hubs can be supported by IT infrastructures that act as intermediaries supporting inter-firm relationships (Christiaanse and Rodon, 2005). Modern ports play a crucial role in the global supply chain integration in that they serve as *"electronic hubs"* to facilitate and coordinate information sharing throughout the chain.

The Port Community Systems (PCS), acting as an electronic hub, have the potential advantage of achieving collective benefits that go beyond the firm level. New generations of PCSs should even go beyond the function of information sharing as they can serve a variety of supply chain processes with different application modules. However, the success of port community systems in ports depends to a large extent on the IT capabilities of ports to design, develop, and implement these inter-organizational systems (van Baalen *et al.*, 2008).

To this point, the inevitability to have a PCS is obvious. However, as noted in Chapter 2, the essential information and coordination requirements that arise from the basic processes in container transport include collaborative planning, disruption alerts, and supply chain visibility. Therefore, the question is to what extent the currently existing PCSs meet these information and coordination requirements. This issue is discussed in Section 4.5 below through analysing the PCSs in a number of ports. Before analyzing current PCSs further, it is useful to examine the definition of PCS with respect to these information and coordination requirements.

4.2 What is a Port Community System (PCS)?

The form and characteristics of PCS vary in each and every port depending upon the port function, location, and hinterland. Furthermore, the functionality of a PCS depends on the initiators of the system as they have a large influence on the development of the system. Therefore, it is hard to put a general definition of PCS because it differs for each port. Various definitions of PCS appear in the literature. In this section, these definitions are presented and discussed.

A relatively old definition of the PCS states that (Grizell, 2001):

> *"Port Community Systems are centrally operated systems for transferring data and providing other services with the help of this data, which can be used by any party who is interested in information concerning sea born transport. A Port Community System avoids bilateral data transfer".*

From this definition, we can infer that the main objective of the PCS is to avoid bilateral transfer of data. This is certainly valid for the objective of the early PCSs where bi-lateral communications were a common pattern for exchanging transactions. Nevertheless, new generations of PCSs are developed to realize more sophisticated objectives. The issue of PCS evolution is introduced in Section 4.3 below. In the early PCS systems, an industry Inter-Organisational Information System (IOIS) was set up and controlled by individual companies, who establish direct links with their partners (Bakos, 1991). This is the case of most of traditional Electronic Data Interchange (EDI) networks. EDI is defined as (Premkumar *et al.*, 1997):

> *"EDI is the direct computer-to-computer communication between an organization and its trading partners of business documents and information in a machine-readable, structured format that permits data to be processed by the receiver without re-keying."*

In an alternative scenario, which is the focus of this thesis, a group of firms of a business sector or industry collaborate to setup up an industry IOIS (Damsgaard and Lyytinen, 2001; Hock-Hai *et al.*, 1997; Henesey, 2002; Kambil and van Heck, 2002; van Baalen *et al.*, 2008). This system is used by these firms and maybe others in the industry. With these industry IOISs, firms do not have to implement direct links with each of their partners; rather, they establish a unique link with the IOIS.

Regarding system functionalities, this definition focuses merely on *"transferring data"* as the main function of the system. It does not provide any identification of the *"other services"* that could be provided by the data.

Another definition has been introduced states that (Smit, 2004):

> *"A Port Community System can be defined as an entity delivering information to supply chains operating in the port. The PCS is responsible for: data supply, data control, data distribution, and data conversion".*

The concept of supply chain appears in this definition. However, it focuses only on information delivery and data responsibilities as functions of the PCS.

Rodon and Ramis-Pujol (Rodon and Ramis-Pujol, 2006) define PCSs as:

> *"An electronic platform that connects the multiple systems operated by a variety of organizations that make up a seaport community".*

They adopt the viewpoint that a PCS is not designed from scratch; it is rather an incorporation of the pre-existing systems (installed base[1]). Therefore, this definition focuses on the integration of pre-existing systems. In this perspective, they indicated that a PCS qualifies as a business sector information infrastructure (BSII) (Hanseth and Lyytinen, 2006). Accordingly, a PCS is a *shared*, *evolving* and *heterogeneous* installed base of ICT capabilities using standardized interfaces (Hanseth and Lyytinen, 2006). It is *shared* in the sense that it is set up, organized and used by firms in the same sector, in this case a port community. A PCS is *evolving* as new companies integrate with it or as

[1] All the IT artefacts including hardware and software.

new types of exchanges or functionalities become available through the PCS. A PCS is *heterogeneous* in the sense that it links multiple types of technologies, processes, people and standards.

A recent definition of PCSs states that (Srour *et al.*, 2008):

> *"Port Community Systems (PCSs) can be defined as holistic, geographically bounded information hubs in global supply chains that primarily serve the interest of a heterogeneous collective of port related companies".*

This definition reflects the new role of ports as an information hub in the global supply chain. In addition, the definition considers the broad view of the port community as well as the diversity and heterogeneity of its members of the port community. These heterogeneous companies often include terminal operators, carriers (ocean, road, and rail), freight forwarders, enforcement agencies (i.e. customs), port authorities, and various lobby groups (including workers' unions, environmentalists, and other policy makers). When such a diverse user community is considered, ports emerge as playing a pivotal role in facilitating effective supply chain operations. This implies that PCSs are supposed to manage and promote general, private and public objectives that are often divergent, if not contradictory.

This definition does not specify any functionality of the PCS; however, it sets the goal of the PCS as a common framework to all players who use maritime transport in a local supply chain. Port community systems bring the diverse actors of the port community together in a transaction recordkeeping and information sharing platform. Therefore, these systems enhance the physical flow of goods since they avail the transaction data in advance of the cargo rather than with the arrival of the cargo.

Although it is not easy to define a port electronic data processing system because these systems differ from port to port; one study indicated that, in principle, the different types of the Single Windows for the maritime community are (PORTEL, 2009):

- **Port Single Window (PSW):** System which provides local level information about the vessel to the authorities on a port level, that has B2G (Business to Government) character.

- **Port Community System (PCS):** A tool to exchange messages in port environment, having a commercial and logistic nature, which has B2B (Business to Business) character.

The study introduced a general definition that can be applied to a port electronic data processing system which is: "*an entity delivering information to supply chains operating in the port"*, and that definition could be given to both systems (PSW and PCS). The main difference between these systems is that some ports use PCS for commercial/merchant transactions and the PSW are used for administrative transactions. However, some ports use both systems indistinctly for commercial and administrative transactions.

Port Community Systems are complex by their nature due to a diverse base of involved stakeholders that operate in the same environment (i.e. the port community). From the operational point of view, port community systems are supposed to manage and promote general, private and public objectives that are often divergent, if not contradictory. A successful implementation of a port community system should be able not only to incorporate business processes of all involved parties, but also to further promote their autonomy. These two apparently opposing principles need to be reflected in the creation of an operational framework of the PCSs.

Port community systems that bring these diverse parties together in transaction recordkeeping and information sharing can also serve to improve the flow of goods. Specifically, with the advent of modern information and communication technology, the transaction data that once travelled with the cargo can now travel in advance of the cargo. This phenomenon coupled with the challenge of landside access at many of the world's ports (Chatterjee *et al.*, 1997; Lipinski and Clarke., 2008) clearly motivates the desire (and need) to exploit the information flows for more than mere recordkeeping purposes. These information flows can simultaneously serve to enhance the physical flow of goods; thereby overcome the bottlenecks that so often occur at ports.

4.3 Evolution of Port Community Systems

Traditionally, the parties in a port community were exchanging cargo related documents and forms for port service requests through paper-based methods, such as fax or handing documents directly. Sending the documents via E-mail also became a common practice due to the diffusion of the internet. The delivered information must be typed again into the port's information systems. Such typing work is time consuming and data is vulnerable to typing errors.

After the appearance of EDI technology, in the context of container transport, parties of the port community started to exchange cargo related documents using EDI messages in a bi-lateral communication pattern. EDI technology enables data transfer between organizations' databases without human intervention. The data is converted into a standard format through a data converter, and then transferred to the trading partner via E-mail, internet or a dedicated line. At the receiver side, the EDI message is decoded and then data is inserted directly into the receiver's database. The key to EDI is use of the same standard data format on both sides of the transfer.

Using EDI can lead to important benefits for an organization. EDI benefits can be categorized into direct, indirect, and strategic benefits (Jimenez-Martinez and Polo-Redondo, 2004). The direct benefits include paper savings, avoidance of repetitive administrative procedures, and reduction in administrative personnel. Indirect benefits include avoiding errors and faster payments which improve cash flow. Finally, strategic benefits include increasing business relationships with companies using EDI and improving customer loyalty. Despite the promising benefits of EDI, a lot of companies were reluctant to implement EDI (van Heck and Ribbers, 1999). The study of Suzuki and Williams (Suzuki and Williams, 1998) addresses this resistance behaviour and states that the resistance is due to uncertainty, lack of standards, and low perceived benefits.

In the traditional EDI scenario, every party sends business transaction data, which could be the same, to every party according to their trading partnership agreements. Data is exchanged using Electronic Data Interchange (EDI) messages instead of hard copy documents. Figure 4-1 depicts the traditional bi-lateral communication pattern in the port community. The bilateral network architecture worked well for establishing connections between large parties that support many information exchange transactions. In container transport, EDI emerged at the seaside, between the large shipping lines (agents) and sea terminal operators. Due to different formats and the extension to the much more fragmented hinterland, there was a need for a central messaging infrastructure, where EDI messages could be routed to different parties and which supported the translation from one format into another. This has led to the emergence of the first PCS in the 1980s.

On the use of port community systems, data exchange is done through a central system. Every party sends business transaction data to a central system which in turn sends the data to the concerned party. This PCS improves the efficiency and effectiveness of communication in the port community. Figure 4-2 shows the communication pattern when a PCS is used.

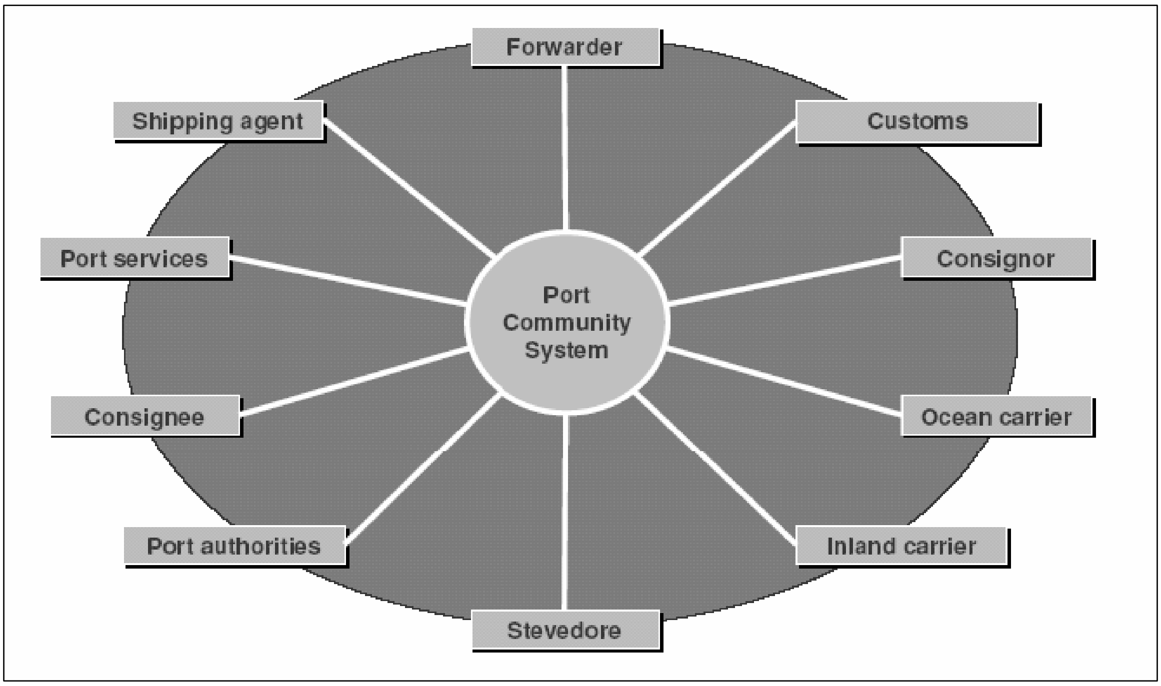

Figure 4-1: The traditional communication pattern in the port community.

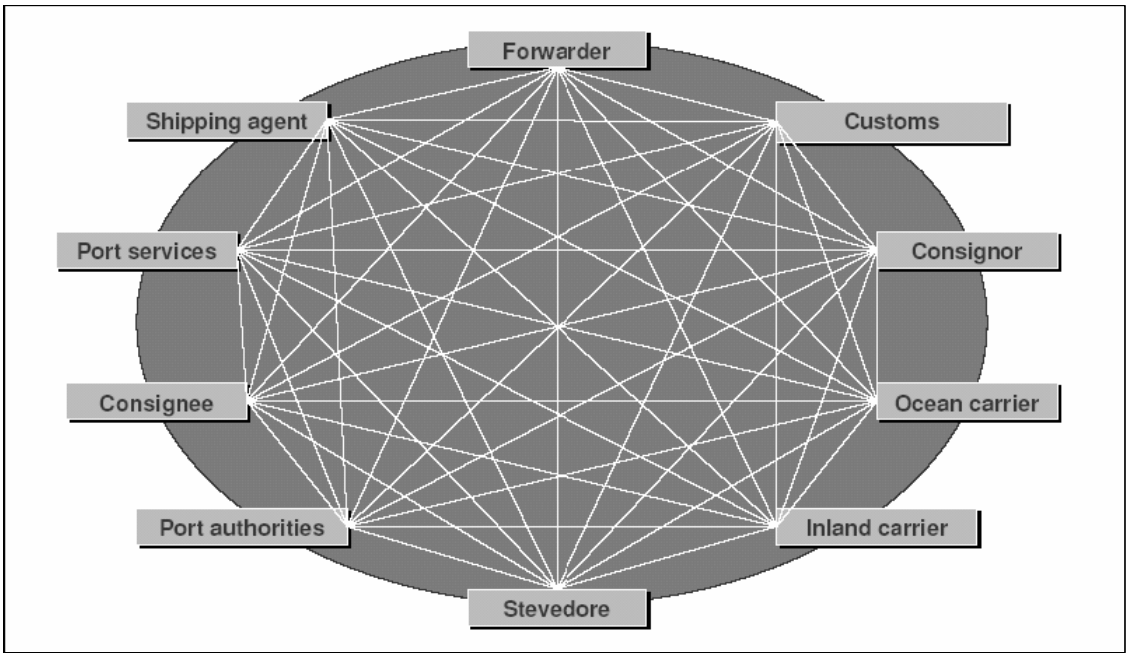

Figure 4-2: The communication pattern when a PCS is used.

Port Community Systems (PCSs) are used to standardize message exchange among stakeholders and centralize all community information as much as possible. By capturing the information produced in any exchange within the community, the need to retype data can be avoided. Electronic message exchange substitutes the exchange of physical documents (postal mail and faxes), thereby reducing errors and processing costs. In the end, PCS provide transparency and possibly real-time information, to facilitate the tracking and tracing of goods and reveal inefficiencies.

Throughout the 1980s and 1990s, several port communities worldwide set up industry IOISs to support the EDI, cargo tracking, electronic documentation, etc, for the organizations that operated in their logistic communities. These industry IOISs, which aimed at speeding up the passage of the goods through the port area, were based on the development of transactional platforms enabling the companies operating in them to exchange data electronically instead of doing so using manual mechanisms (fax, telephone, letter, etc.). These industry IOISs support the information flows between the landside and seaside transport chains (see Figure 4-3).

Two generations of PCSs can be identified in ports across the world as depicted in Figure 4-4 (Grizell, 2001).

In the first generation, various different bilateral applications are linked together to create a central information system. In this case, the PCS acts as a post box which allows actors to exchange messages. A PCS with a Generation I architecture is common among the older systems that started in the 1980s.

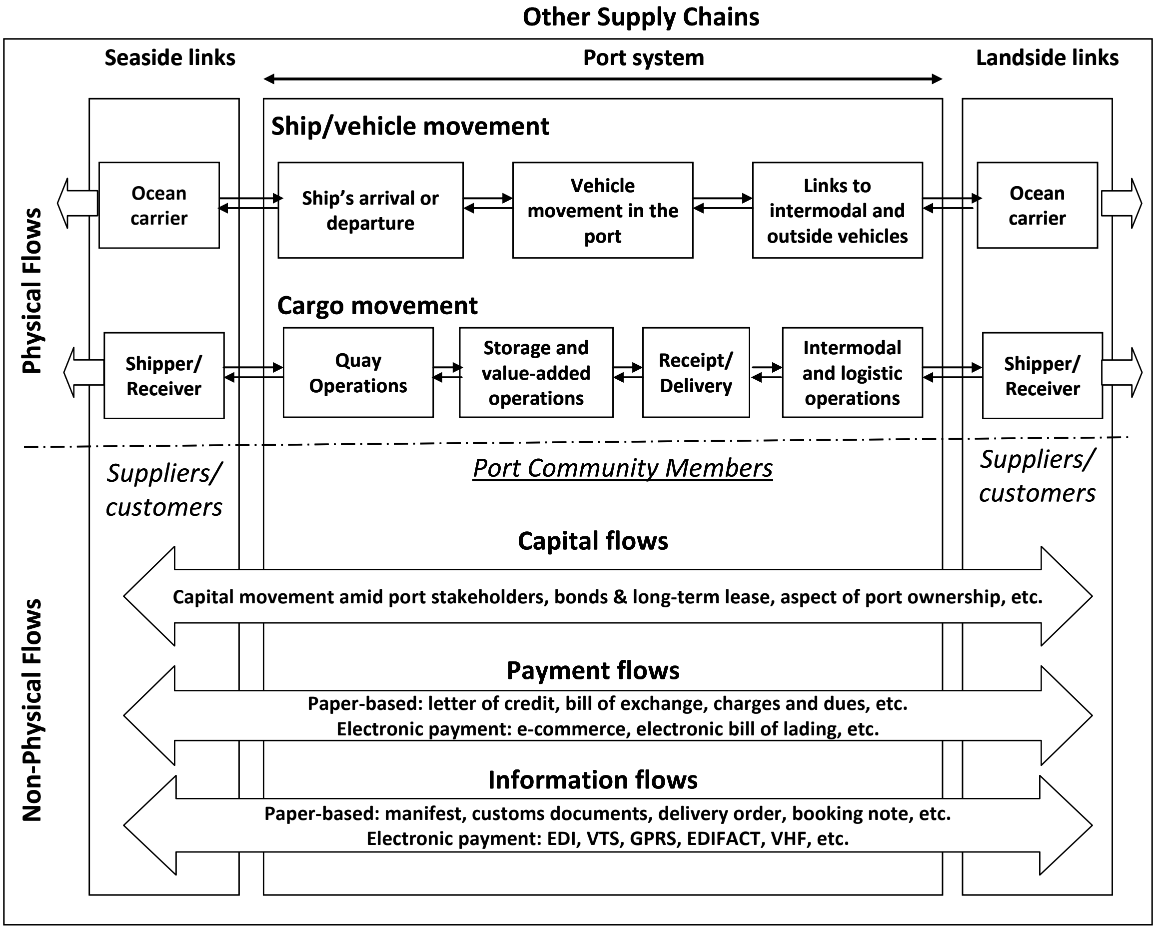

Figure 4-3: The information flows between the landside and seaside transport chains (Beresford *et al.*, 2004).

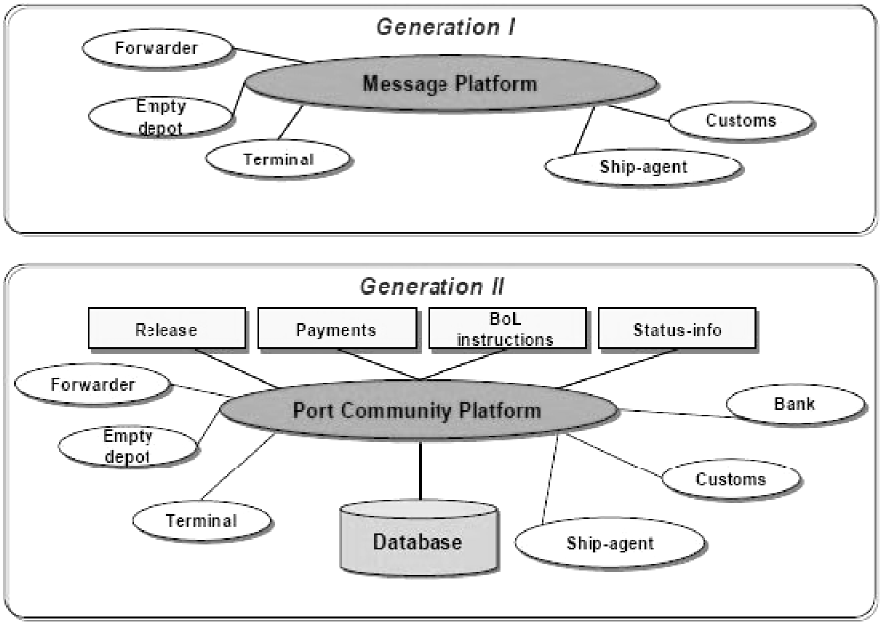

Figure 4-4: The structures of the two Generations of Port Community Systems (Grizell, 2001).

In a PCS with a Generation I architecture, an application with an underlying database (message platform) is created for the exchange of information between two parties in the port. Some parties exchange information in their application that is also needed in other bilateral applications. In such a case, an application will be developed between two bilateral applications linking the two bilateral systems together. Since information in ports is used by numerous parties for various purposes, many applications have emerged that link bilateral systems, thereby creating an (unmethodical) Web of applications which together comprises the PCS. Moreover, Generation I PCS have a more complex structure, which leads to higher maintenance costs and difficulties in extending and expanding with new services and communication channels.

The Generation II PCS architecture has been developed in order to overcome the above mentioned shortcomings of the first generation. The architecture of Generation II PCS consists of one central database for message storage. Port community users can send or extract data from the central database according to certain user privileges. In addition, the central database is used for providing additional services. On top of the central database, a modular and intelligent middleware (port community platform) is built representing different processes in the port.

The modules are the points where information can be entered or adjusted. The intelligent middleware controls how information is connected between modules and will manage data manipulations accordingly, thereby supporting unique data sourcing. This architecture can be defined as a mix of a centralized system (the database and the middleware) combined with local flexibility in the modules.

The two generations of port community systems focus on the capability to retrieve data from other information systems and distribute to authorized users, either by connecting bilateral information exchanges (Generation I) or by providing modular information exchange services to which organizations can subscribe via the port community platform (Generation II). However, Generation II port community platforms have the potential to provide services such as collaborative planning, disruption alerts, and visibility. The analytical survey presented in this chapter demonstrates that ports are in different stages of making the transition from Generation I to Generation II architecture.

Principally, PCS was developed to ease the exchange of information within a port community. Three e-collaboration models have been reported for information exchange in port information systems and port community systems as follows (Boertien *et al.*, 2002):

Bilateral Information Model (BIM): In this model, information is exchanged directly between the different actors on bilateral basis (see Figure 4-5). When basic communication channels (e.g. fax, telephone or e-mail) are used, BIM type of information exchange is easy and relatively cheap to implement. However, the bilateral information model suffers from problems of scalability, and therefore, it is best suitable for the situations where the number of parties involved in the information network is relatively small (Srour *et al.*, 2008).

Centralized Information Model (CIM): In this model, an independent operator provides centralized information services and stores, forwards and retrieves all information (see Figure 4-6). The service provider can also offer value-added services that may improve logistic processes of the companies. In the CIM model, information is not pushed from one actor to another as in the BIM model but it is retrieved on demand. The centralized information model is well-suited for small and medium sized companies in communication with big parties who have their own internal system with EDI access (Boertien *et al.*, 2002).

Decentralized Information Model (DIM): In this model data is stored and controlled by each individual party (see Figure 4-7). The central broker is not responsible for the control of the actual information. A broker service can help in retrieving the information from the right source. It is only aware who knows what information, where information is stored, how the information is retrieved and when the information is updated. In the DIM model, information is exchanged when it is needed (Boertien *et al.*, 2002).

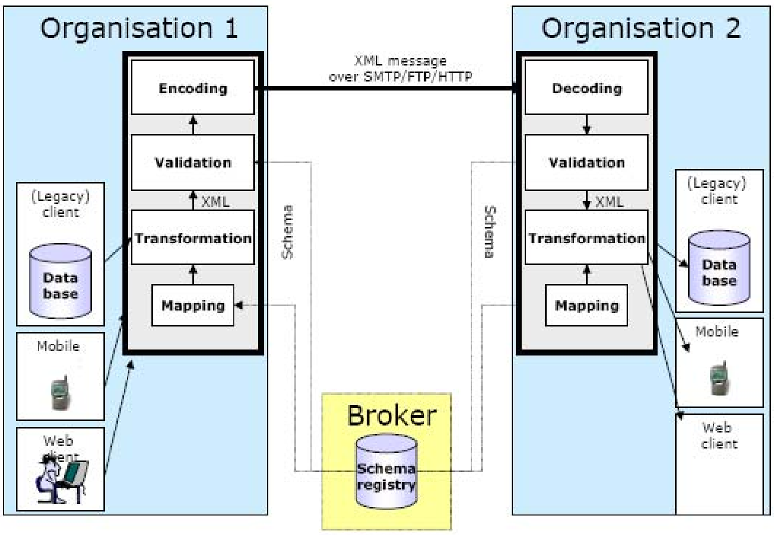

Figure 4-5: Bilateral communication model with help of a standards registry (Boertien *et al.*, 2002).

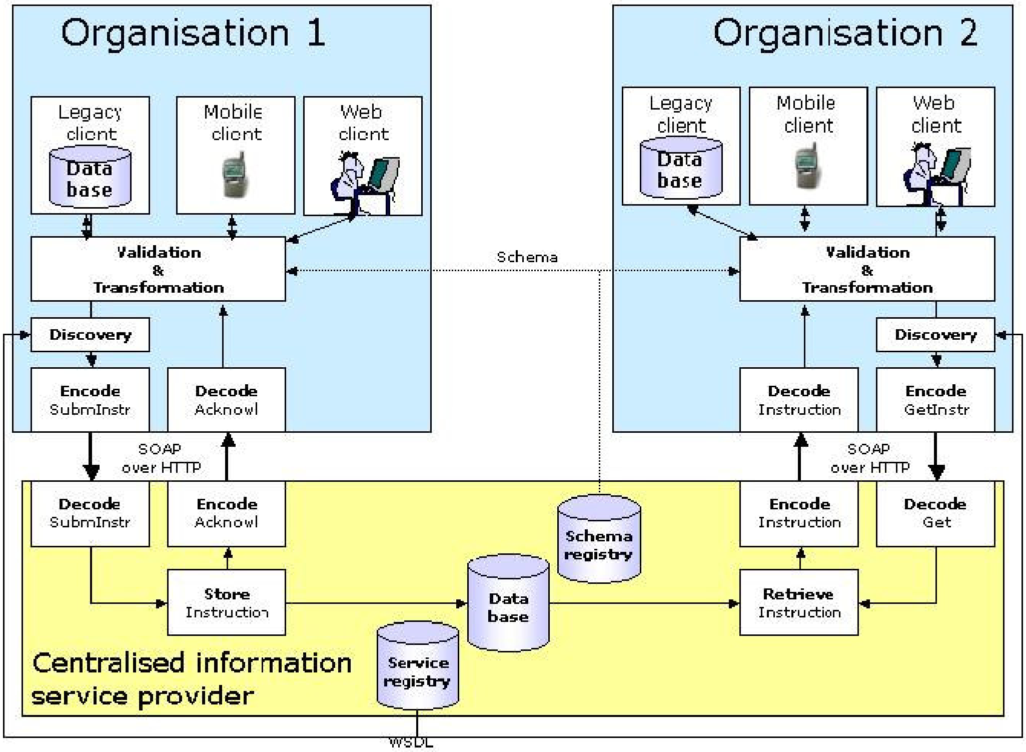

Figure 4-6: Centralised information service provider (Boertien *et al.*, 2002).

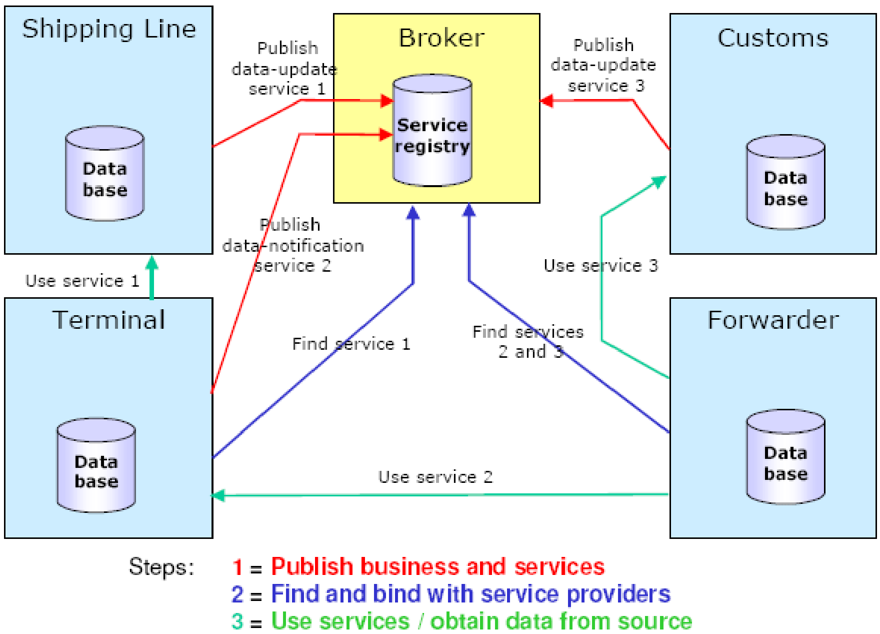

Figure 4-7: Decentralised information provider (Boertien *et al.*, 2002).

4.4 PCS in the literature

In spite of the wide spread and the recognized benefits of PCSs in ports all over the world; there are few studies in the field of information systems that focus on investigating port community systems. Some of these studies are descriptive, for example the case studies on PortNet in the Port of Singapore (Applegate *et al.*, 2004), or TradeLink in Hong Kong (King and Konsynski, 1990). On the other hand, there are a number of analytical studies that are mainly concerned with utilization assessment, adoption, and security issues of the PCSs. To the best of our knowledge, there has been neither academic research assessing the business process capabilities of PCSs nor other any studies on the performance of these PCSs.

In this section, we portray how researchers and practitioners deal with issues related to PCSs. The aim of presenting such an overview is primarily to position this work in the domain of information systems application in port community. In addition, this literature review gives more insights on different perspectives of PCSs, and this presumably reveals new lines of research in this area.

Based on the enterprise architecture, a comparative study has been carried out for three port community systems; namely, Seagha in the Port of Antwerp, DAKOSY in the Port of Hamburg, and Port Infolink in the Port of Rotterdam (Smit, 2004). In that study, the architecture view considers both the business elements and the Information Technology elements and their interrelation. The objective of the study was to identify differences and similarities of the three PCSs. For that purpose, a Port Community Architecture Framework (PCAF) has been developed to analyze each port community system, and accordingly a comparison was drawn. The framework includes various aspects of PCSs and identifies the points of interest to compare PCSs.

The role that PCSs can play in terms of implementing a comprehensive Single Window was highlighted in an article that traces the history of the Felixstowe Port Community System from the early days of planning (Long, 2009). The objective was to provide an insight into the benefits that can be achieved by both Government and the trade, transport and logistics communities.

Regarding the use of information systems in European ports, an inventory of the reporting transactions carried out by European Sea Ports using their electronic data processing systems was developed (PORTEL, 2009). The study has been carried out by PORTEL SERVICIOS TELEMATICOS S.A.[2] within the context of the SKEMA[3] project. It aimed at establishing a Sustainable Knowledge Platform for the use of stakeholders in the maritime transport and logistics industry. The inventory developed in the study includes 12 European ports where several types of information about their information systems have been compiled (for example the transactions achieved by the systems; their function and the documents required by those transactions; the authorities and other stakeholders that interact with these systems and how they do it).

2 http://www.portel.es
3 SKEMA is a three year project funded by the European Commission - DGTREN under the Seventh Framework Programme. (http://www.skematransport.eu)

The IAPH[4] (International Association of Ports and Harbours) and its Committee on Trade Facilitation & Port Community Systems carried out a survey on the main features of PCS running at member ports (Mila, 2007; Mila, 2009). The objective of the survey was to clarify the models of PCS running at member ports and to identify best practices.

The survey results (Mila, 2009) revealed that the port authority, either alone or with other member of the port community, is the initial promoter to the PCS in the majority of the cases under investigation. However, the use of the PCS is mandatory in only 50% of the cases under investigation. In most cases (about 70% of the cases), the PCS does not remain as an isolated system; it is rather connected to other portals or PCSs. Regarding offered services, the PCS offering information services and documents exchange services using EDI and/or XML.

Another survey on existing port community systems has been carried out within a study that originally aimed at the evaluation of the suitability of a port community system to the Finnish port environment (Posti *et al.*, 2011). Their survey revealed that especially the large container ports located in Western Europe and in Southern, Eastern and South-eastern parts of Asia have highly developed PCSs. On the other hand, only few comprehensive port community systems exist in the Baltic Sea region.

[4] http://www.iaphworldports.org/

The study was mainly focusing on the evaluation of a port community system in Finnish ports. The evaluation was carried out through interviews with 35 Finnish logistics/ICT experts in 22 organizations in South and Southeast Finland. Their results revealed that the information sharing in a typical Finnish port supply chain contains several bottlenecks that cause delays in shipments and waste resources (For example, the use of traditional bilateral communication techniques and the large number of diversified documents or messages). The information systems and services used in Finnish port community are actually well developed but from the users' point of view they are rather scattered. They also conclude that Finnish ports lack a comprehensive port community system that would combine G2G (Government to Government), B2G (Business to Government) and B2B (Business to Business) under one Single Window information system.

The issue of **PCS evaluation** from the user's point of view has attracted other researchers. An evaluation of the Korea's national port community system (PORT-MIS) has been carried out from port users' point of view (Keceli *et al.*, 2008b). PORT-MIS was developed in Busan Port in Korea for exchanging required documents between port users and the port authority. The research aims at evaluating the success level of PORT-MIS from users' point of view, and accordingly proposing a direction for further improvement of the system for customer satisfaction.

The evaluation of PORT-MIS was carried out based on four technical factors that influence the use of PORT-MIS; namely, compatibility, complexity, cost, and reliability. In addition, organizational trust towards the operator of PORT-MIS was considered as a factor for evaluation (i.e. the willingness to rely on an exchange partner in whom one has confidence). This factor was employed in the evaluation although it is an organizational factor rather than technical. The evaluation has been carried out by conducting a questionnaire survey targeted to sea and land carriers in the Busan Port region and by interviewing the operators of the system. They conclude that the perception of the users about PORT-MIS is in the medium level for all of the factors, and this was interpreted that the users of PORT-MIS are satisfied with the current level of the system to some extent but the system still needs further improvement in some areas. However, the study did not specify any of these areas.

In the same research vein, a number of studies have been carried out in order to assess the situation of utilization of information systems in Turkey's major public and private container ports (Choi *et al.*, 2008; Keceli *et al.*, 2007; Keceli and Choi, 2008; Keceli *et al.*, 2006). Problems in information systems as well as sever administrative and structural problems in public sector Turkish ports had been identified. Accordingly, and based on a comparison to the world's advanced ports, an improvement directions have been suggested. It was found that the relatively low level of information technology utilization in Turkish public ports is due to severe administrative problems, such as out-of-date business processes, severe bureaucracy and insufficient human resources, rather than any technical incapability. The studies proposed directions for improvement that are essential for the future development of information systems in Turkish ports. These directions include the enhancement and simplification of regulations, standardization of information system specifications in Turkish ports, and establishing a common understanding of information system among the members of the port community.

Another line of research that is perceived from the literature of PCSs is the investigation of the **implementation and adoption of PCSs**. In Srour *et al.* (Srour *et al.*, 2008); they gave insight into PCS implementation success factors in practice and provided a documentation the lessons learned from PCS implementation. The deployment of information systems at ten ports in North America, Europe, and Asia has been studied. The review has been carried out with particular focus on the PCS deployment life-cycle framework which includes four stages: project initiation; the system analysis and design; the implementation and adoption process; and the maintenance and growth of the system. They also identify key success factors associated with each stage where the success of the deployment appears to be correlated to specific practices undertaken within each of these stages.

They reached a general conclusion that the success or failure of the system deployment in the cases under investigation was due to specific issues encountered in distinct stages of system development. They end up with lessons learned for successful practices during the development of PCS as summarized in Table 4-1.

	PCS deployment Stage	Lessons Learned
1	Project Initiation	• Support is most easily gained when the problem(s) and goal(s) are clear and urgent for all parties. • Sponsorship should be carefully considered in terms of both type (i.e. financial or legislative) and timing.
2	System analysis and design	It is important that the system architecture is carefully designed to match the technological capabilities of the participants and mirror the existing port environment and culture. The system architecture should mirror the organizational context.
3	Implementation and adoption	A modular implementation strategy will likely yield a successful “start small and add” deployment model. Each module must have clear objectives and quickly realized tangible benefits for all parties involved.
4	Maintenance and growth	• The ability of the information system to evolve in order to seize emerging opportunities will enable its continuous success. • PCS should be continually maintained and updated in order to meet the requirements of evolving business practices. The system’s ability and agility to grow and evolve are the key issues in this stage.

Table 4-1: Summary of Lessons Learned from PCS implementation [Adapted from Srour (Srour *et al.*, 2008)].

As discussed in Section 4.2, a PCS connects multiple information systems that are operated by members of the port community. Therefore, the issue of integrating pre-existing systems in a group of firms with a PCS had been explored by scholars (Rodon and Ramis-Pujol, 2006). They have investigated the ePortSys system as a case study in their research. The ePortSys is a PCS that was launched in mid-1999 in a Spanish seaport in order to coordinate the activity of firms in the port’s landside transport network and to integrate all the information being exchanged between the various port agents (Rodon and Ramis-Pujol, 2006).The study focused on external integration (i.e. the integration between systems external to an organization with the internal systems of the organization) at two levels; interface integration and network integration.

Interface integration refers to the process during which a firm alters its business practices and applications so that they interface with its EDI applications. **Network integration** refers to the process during which a group of firms agree on the way they are going to interact electronically and to connect their systems.

In particular, the study focuses on the actions taken by diverse actors in order to carry out necessary adaptation during the integration of the different systems. The study tackles two types of adaptation: technical and organizational. Technical adaptation includes the modifications to existing hardware, telecommunications, software, and data models to integrate with the PCS. Organizational adaptation includes the changes in workers’ roles and responsibilities, necessary skills, experience and knowledge, as well as the changes operational processes and structures as a result of integrating with the PCS.

According to their results, the major challenges related to the introduction and development of ePortSys system were: companies’ willingness to retain their autonomy, limits of the standard, difficulties in aligning the interests of different actors, consequences of tighter integration with the system, and unexpected uses and effects of the system.

Regarding the community adoption of PCSs, the factors affecting PCS adoption by the sea and land carrier companies located in the Busan Port region have been examined (Keceli *et al.*, 2008a; Keceli *et al.*, 2008c). They have developed a model to determine the factors affecting the user acceptance of PCS and to assess the influence of these factors on PCS adoption. In that model, the factors that affect the use of PCS are divided into four categories; namely, factors related to PCS service provider, factors related to provided services, factors related to PCS adopter, and factors related to system characteristics.

According to the results of that study, factors related to PCS adopter appeared to have the most important influence on PCS adoption among the four factor groups. In particular, factors related to top manager support, adopters' technical and non-technical readiness, competition and perceived benefits have the most important influence on PCS adoption. On the other hand, additional services, such as logistics solutions or e-Business functionality among the PCS users have the least impact on the system adoption. It may even decrease the intention to adopt the PCS because most of the users consider that the port community system as a tool for submitting documents related to port authority and customs rather than a mean for generating added value

They also conclude that:

- Technical reliability of the system has also direct impact on PCS adoption while cost of the system is seen as a less important factor. Thus, security, encryption and networking issues must be fully considered in PCS development.

- Additional logistics services and e-Business functionality among port community members should be delivered in such a way that users' concerns about information leakage and decreased competitiveness must be addressed. Thus more detailed analyses are needed to develop a business model to overcome these issues.
- The main target of PCS should be small companies. Thus PCS development projects should aim to design the system to fit the requirements of small and medium size companies. In the case of large companies, long term relations in strategic level must be established.

Although the study has reached useful results, it is difficult to generalize the outcomes of this study for other ports around the world. The reason is that the behaviour of PCSs adoption varies significantly in different countries while the study was limited to the users in Busan port as well as its limitation to sea and land carriers.

Another research area that is related to the implementation of PCSs is the **security issues of the PCSs**. In this regard, Aksentijević *et al.* emphasized the role of cooperation between stakeholders involved in a PCS and the internal education for users to achieve a proper organizational security (Aksentijević *et al.*, 2009). In addition, they indicated that the unification of common security language between all port community stakeholders should also be considered. Further, the researchers recommended the standardization/certification of information security practices and risk management of the system in order to ensure continuity of the system. In the same research vein, **Tijan et al.** emphasized the importance of the methods and sets of procedures that can be used to ensure port community business continuity and disaster recovery in case of a catastrophic event caused either by human error or inevitable accident (Tijan *et al.*, 2009). They indicated that every port community system should have a disaster recovery plan that clearly outlines the measures that should be executed within port security systems in order to eliminate threats and to remedy possible disasters. They also recommended that the business continuity plan should be a part of the overall PCS risk assessment.

4.5 Survey on the current Port Community Systems

In order to get more insight into Port Community Systems (PCSs) in practice, we carried out a purposeful exploration of the main features and benefits of PCSs currently exist in ports. The objective of this exploration is to assess the capabilities of the existing PCSs with a particular focus on their functionalities and benefits to the port community as a supply chain. Specifically, the survey focuses on the availability of functions for collaborative planning, disruption management, and supply chain visibility.

The port cases were selected based on its advanced position in their world port ranking and its advanced level of IT utilization. In this regard, we rely on the world port ranking of the American Association of Port Authorities (AAPA)[5] from which we selected 6 port cases out of the top 20 ports. The selected port cases are illustrated in Table 4-2.

[5] http://www.aapa-ports.org

Port	Country	System	Operator
Singapore	Singapore	PortNet	Portnet.com
		TradeXchange	CrimsonLogic Limited
Hong Kong	China	OnePort	OnePort Limited
		Tradelink	Tradelink Electronic Commerce Limited
Busan	South Korea	PORT-MIS	KL-Net
		KTNET	KTNET
Rotterdam	Netherlands	Port Infolink	Port Infolink B.V.
Hamburg Port	Germany	Dakosy	Dakosy AG
		COAST	HHLA
Antwerp	Belgium	Seagha	Seagha

Table 4-2: The Selected Port Cases and their Port Community Systems.

Lack of publications on port community systems limits the number of examined port cases as well as the depth of detail. However, all available literature relevant to each case was reviewed. In addition, Web sites of the case studies were also investigated. Most of the essential information was successfully brought together from publicly available documents and Web sites, however, experts were interviewed to confirm and to fill in any missing information. To track this data collection process, a summary document was prepared for each case, benefits and key features have been identified for each case, and then a focused study was carried out for all cases. An overview on PCS implemented in some leading ports of the world is introduced in the following sub-sections.

In addition to the PCSs, cases of relevant inter-organizational information systems are also consider in the survey. These cases are not fundamentally a PCS due to its limited scope; it rather could be employed as a component in a PCS. Furthermore, inter-organizational systems that are under development are also considered; such as the

Shared Intermodal Container Information System developed in the European project INTEGRITY[6].

4.5.1 PCS in the Port of Singapore

The Port of Singapore (PSA)[7] plays a pivotal role in many global supply chains, as well as a trading, manufacturing, and transport hub. This is for a large part due to its strategic position and its remarkable port facilities for the transhipment of containers. It is the world's largest container port and it is a good example for the best practice of information technology application in ports.

In its ambition to become a global hub for business, services, and transportation; the Singapore government invested heavily in the development of large inter-organizational infrastructures. In 1980s, PSA started with the implementation of two very successful, large EDI-networks: PORTNET[8] and TradeNet[9]. These two EDI networks became the backbone of an integrated solution that provides services over the Internet.

Currently, PSA provides integrated services through its Collaborative Port Community Solution. Their solution is considered as the world's first nation-wide business-to-business (B2B) port community solution which enables cross-community process integration. It was the key factor for ranking Singapore as the World's Number one

[6] http://www.integrity-supplychain.eu
[7] http://www.internationalpsa.com
[8] http://www.portnet.com
[9] http://www.tradexchange.gov.sg/tradexchange/default.portal?_nfpb=true&_pageLabel=main_tn

logistics hub by the World Bank in 2007. The solution offers services for connecting shipping lines, haulers, freight forwarders, shippers and local government agencies operating in Singapore. It serves about 8000 integrated users and generates about 90 million transactions a year (PSA, 2006).

The technologies that are used in this solution include:

- Internet business-to-business integration services – for real time integration of information exchange
- Business Process Engine – to proactively manage business activities and provide alerts to exceptions
- Mobile Technology – to extend access to mobile devices, such as Personal Digital Assistants (PDAs) and mobile phones
- Clustering Technology - to provide high availability and reliability

The solution provided by PSA integrates a number of components as follows:

- CargoD2D™: It is the system trough which a shipper/consignee sends electronic shipping orders to a carrier.
- Haulier Community System: It is the system to which the shipping order is routed as a trucking order to the Haulier. Through the Haulier Fleet Optimisation Engine, the optimal truck and driver for the operation requirement is selected.

- PORTNET®: It handles all electronic vessel and container data passing through PSA Singapore Terminals.
- TradeNet: It is the system through which various parties can exchange structured trade messages and submission of permits declarations online (for example inward, outward or transhipment permits).
- EZShip® and ALLIES™: To provide a facility for shipping lines to ensure the best connecting vessel and allocate vessel space before containers can be loaded.
- TRAVIS™: It is a business reporting and management tool that provides reports on transhipment details, throughput, dwell time, and vessel performance.
- PORTNET® Mobile: It is the system that allows users to access real time information and interface with the community anytime, anywhere, via any mobile device. Real-time information available includes container status and current container position which can be automatically pushed to the customer's mobile device or provided on demand. This enables more efficient planning of resources and provides better customer service in the handling of cargo.

Through these solutions, PSA has empowered the entire port community with a suite of innovative services to add value to their ship operations and cargo handling.

4.5.2 PCS in the Port of Rotterdam

The Port of Rotterdam[10] is the largest container port in Europe. The container throughput in 2010 was over 11 million TEU (see port statistics at www.portofrotterdam.com).

Based on strong demand of the users of the port for a PCS, the Port of Rotterdam decided in 2001 to analyze the scope and potential solutions for a future PCS in the project "Port of Rotterdam Main Information Services (PROMISE)". One of the conclusions was that the PCS could not be bought from another port, and that it should be developed with the latest proven technology. Accordingly, a company named "Port Infolink" has been established in August 2002.

Port Infolink started to build the PCS from scratch based on the analysis of business needs and the elicitation of associated information requirements. The PCS services support the import and export processes, communication between organizations in the market, and communication with governmental organizations. Port Infolink enhances the port of Rotterdam's competitive edge by optimising the information flows and work processes of the public and private sector. It does that by developing a port-wide Port Community System with services geared towards specific target groups, which facilitate the simple and efficient exchange of data. The Port Community System offers 24 different services, with around 4500 users who send more than 20 million electronic messages a year.

[10] http://www.portofrotterdam.com

Generally, Port Infolink provides infrastructure with a goal of reducing the costs of the port users and therefore improve the competitive position of the port of Rotterdam. The users of Port Infolink are shipping agents, shipping lines, terminal operators, the port authority, customs, the veterinary authority, banks, barge operators, rail operators, road hauliers, and forwarders.

The main services that the PCS provides include:

- Track and Tracing (T&T) per product
- Sending reports and declarations between private companies and government (customs)
- Barge and road planning
- Cargo/container/vessel arrivals and departures
- Veterinary inspection process for containers (from line to terminal)
- Declaration transit documents for containers
- Customs scan process for containers

A recent development is the merger of Port Infolink with the Amsterdam PCS PortNet into the common system Portbase[11]. Currently, Portbase offers 28 different services to around 5000 users in all sectors of the ports of Rotterdam and Amsterdam where about 33 million electronic messages are exchanged via the PCS per year.

[11] http://www.portbase.com

4.5.3 PCS in the Port of Hamburg

The Port of Hamburg[12] is the largest German port and it is one of the largest European ports. It is one of the ports in the so-called Hamburg-Le Havre port range, which includes: Hamburg, Bremen, Rotterdam, Antwerp, Zeebrugge, and Le Havre. The competition between these ports is severe. Clients easily switch to other ports because of differences is prices, tariffs, speed of transhipments, etc.

During 1970s, the idea of developing a common EDI system in the port of Hamburg has been initiated by the need of the port-related companies to exchange information. In the mid of 1983, Hamburg's data communications system (DAKOSY)[13] has been launched. The system was named after the independent company "DAKOSY", who developed of the system. The launch of DAKOSY signalled a quiet revolution in maritime and logistic business transactions.

Currently, DAKOSY is the "single window" of the Port of Hamburg which became a "paperless port". All companies and authorities involved in the export, import and transit processes can handle their transport processes rapidly and with electronic assistance by using the B2B services and applications of DAKOSY. The system users are more than 2000 include haulage companies, line agents/ship owners, rail transport companies, trucking companies and feeders as well as all involved authorities (e.g. customs, harbour police, fire service etc.).

[12] http://www.hafen-hamburg.de
[13] http://www.dakosy.de/en

DAKOSY is a messaging platform that facilitates information flow among the companies along the whole transport chain in the Port of Hamburg. The export-related transport (and transit) processes are supported by the “EMP - Export Message Platform", while the "IMP - Import Message Platform" is designated to support the import-related processes.

The EMP groups together all necessary documents that the business partners exchange with one another and with the involved authorities as part of the export or transit process. These documents are exchanged as messages using internationally standardized message formats (e.g. EDIFACT or XML). All messages can be implemented and used by the customer as an EDI service.

The IMP supports the individual sector requirements of the carriers, terminals, haulage and transport companies and all relevant authorities, while optimizing the overall process from the ship's entry to the Port of Hamburg through to delivery of the goods at the customer in the hinterland.

DAKOSY uses a distributed architecture for the PCS. It means that a large number of bilateral applications are linked together to create a central PCS. Whenever a new service is needed; DAKOSY company starts building a new application. Examples of these applications are:

- ZODIAK (for information exchange with customs);
- TRUCKSTATION (exchange between haulers and terminals);

- HABIS (connecting German railways with the shipping and transport industry);
- GEGIS (for hazardous goods electronic notification to the harbour police);
- UNIKAT (connecting rail operators with transport partners); and
- SEEDOS (connecting German ports with forwarders).

The main services that DAKOSY provides are:

- Tracking and tracing of vessels; real-time data updates;
- sending reports and declarations between private companies and government (mainly export to customs and dangerous goods to the port authority);
- statistics (for example counting the number of messages);
- rail, barge, and road planning; and
- cargo/container/vessel arrivals and departures (vessel scheduling and status information).

4.5.4 PCS in the Port of Busan

The Port of Busan14 lays at the south east of the Korean Peninsula. It is the main port in Korea which is considered as the gateway that connects the Pacific Ocean to Asia. PORT-MIS was developed by Busan Regional Maritime Affairs and Fisheries Office and implemented in Busan Port in 1992. Port-MIS is an online port management system that provides port clearance service and real-time access to the operational information of a vessel from its arrival to departure. It is used to manage and control port activities and administrative procedures of vessel and cargo movements inside the port.

Generally, the main functions of PORT-MIS are built on four management fields: vessel management (e.g. vessel registration, vessel movement control, berth control, pilot control), cargo management (e.g. cargo registration and dangerous cargo reports), billing management and statistics management.

The communications and Electronic Data Interchange (EDI) services are provided by KL-Net[15] which is a private network company. KL-Net provides a range of online services through which companies apply for services request, such as booking for port facilities, customs clearance, immigration application for ship crew, railroad transport, request for inspection. In addition, there is a financial module that enables online payment of taxes and port services.

The system connects terminal operators, carriers, fright forwarders, haulers, and inspection companies, and therefore it supports collaborative business. The system provides precise logistics information, automated handling of data comparison, and a standardisation of requests and orders to minimise information duplication.

[14] www.busanpa.com
[15] http://www.klnet.co.kr

4.5.5 Reflections on the Survey of the Existing Port Community Systems

The analytical survey presented in this chapter revealed that the main objective of the PCSs under consideration is to streamline business processes and to control the document flow in the port community. This objective is realised by providing a platform that the port community parties can use to exchange documents in electronic format rather than the traditional paper format (i.e. paperless environment).

The availability of such platform allow for providing useful functions and services to the port community actors. The services that are provided by the examined PCSs can be grouped into three major categories; namely, services related to port management, services related to customs, and electronic commerce services. On the other hand, the main functions of the surveyed PCSs can be summarized as follow:

- To provide the online services.
- To facilitate electronic exchange of business documents.
- To integrate the actors of the port community through seamless flow of information.
- To coordinate the business processes using the readily available information.

The cases examined also highlight the role that ports serve in global supply chains. While the port itself may be geographically bounded, a PCS does not need to be similarly bounded. Port community systems can pass information upstream and downstream through supply chain and around the world. Extending the coverage of port information systems provides the opportunity for chain-wide (visibility) services and overall workflow orchestration.

When the functionalities and services of the examined PCSs were critically analysed, we perceived that some of these systems offer only a portion of the services required by the port community supply chain. The type of the services that are provided by the PCS depends on the major stakeholders of the system. This is reflected by the absence of functionalities related to business process management. For example, we did not find any document indicating the existence of business process monitoring as a service in the PCS.

In addition, the survey revealed that the performance of the business processes and in particular the performance of the port community supply chain is not in the scope of the examined PCSs. We did not find any evidence for measuring the performance of the whole port community supply chain. Furthermore, the concept of the management of exceptions is not clear in any of the examined PCSs. Only the port community system of the Port of Singapore states that it provides a real-time tracking and proactive notification on business exceptions, however, they did not give any further details about how these exceptions are handled.

As indicated in Chapter 2, the information and coordination requirements to deal with issues raised from basic processes in container transport include: collaborative planning, disruption alerts, and supply chain visibility. From this perspective, the survey of the examined PCSs revealed that these requirements are incorporated only to a certain extent. The PCS and their modules all support visibility by providing a platform for information exchange. However, additional features for collaborative planning and disruption alerts are not fully supported.

Collaborative planning modules are provided (e.g. Rotterdam PCS), but they merely support the exchange of planning information as opposed to the support of a joint planning process. The full support of monitor and control activities (see Chapter 2), by providing alert mechanisms, have not yet been developed. The development of alert mechanisms is in the scope of inter-organizational systems that are under development, such as the Shared Intermodal Container Information System developed in the European project INTEGRITY[16].

In conclusion, the analysis of the PCSs under consideration revealed that there are a number of shortcomings in the existing PCSs. We perceived these shortcomings as a strong support to our argument that the port community needs a middleware that provides such services. Therefore, in this thesis we propose our agent-based middleware which is introduced in Chapter 5.

4.6 Summary

Principally, the port community systems (PCSs) have been employed in ports in order to reduce paperwork and facilitate the information flow related to port operations and cargo clearance. However, the structure of these PCSs as well as their features and functionalities vary in each and every port depending upon the port function and location. Therefore, we give we give more insight into Port Community Systems (PCSs) in this chapter through introducing various definitions of PCS as well as their evolution.

The main objective of this chapter is to introduce the findings of an analytical survey that we carried out to evaluate PCS in practice. The evaluation focuses on the business process management capabilities of the existing PCSs. In particular, we focus in the capabilities for collaborative planning, disruption management, and supply chain visibility. The survey includes a number of advanced ports.

This survey presented revealed that the main objective of the PCSs under consideration is to streamline business processes and to control the document flow in the port community. The most important finding is that the performance of the business processes and in particular the performance of the port community supply chain is not in the scope of the examined PCSs. In addition, the collaborative planning functionality is very limited and the concept of the management of exceptions is not clear in any of the examined PCSs.

It is these limitations of the existing PCSs which motivates our proposed agent-based approach. In the next Chapter, we discuss the potential benefits of employing the agent-based approach, and then we give a detailed description of our proposed agent-based middleware.

[16] http://www.integrity-supplychain.eu

Chapter 5

The Agent-Based Middleware for Port Community Management (ABMPCM)

5 The Agent-Based Middleware for Port Community Management (ABMPCM)

5.1 Introduction

The analysis of the current Port Community Systems (PCSs), as noted in Chapter 4, revealed that current PCSs are weak in, or even lack, the capabilities to manage and control the business process of the port supply chain. Consequently, we believe that it is necessary to have a more integrated, flexible, dynamic, and autonomic solution, which can provide business process management requirements to port community actors without affecting any current PCS configurations and infrastructure.

Accordingly, we propose, in this Chapter, the Agent-Based Middleware for Port Community Management (ABMPCM) as an approach for the management of business processes in the port community supply chain. A description of that ABMPCM is fundamentally what this chapter is about.

The ABMPCM promises to enhance the performance of the container terminal community supply chain. As discussed in Chapter 2, the performance of the container terminal community, as a supply chain, can be improved through coordination at the tactical level, and disruption management at the operational level. Therefore, the proposed agent-based middleware provides essential management facilities (functions) to members of the port community. These functions will facilitate collaborative planning, process management and visibility. In addition, ABMPCM will assist a port community member (e.g. consignee) to creating, executing, and monitoring of a business process that will realize his requirements.

In this chapter, we first emphasise the rational for adopting the multi-agent approach as a solution for managing the port community supply chain. We then describe the framework of our proposed ABMPCM and its functionalities. A business scenario to explain the usability of the ABMPCM is also presented.

5.2 The Need for Multi-Agent Systems for Managing Port Community Supply Chain

The idea of employing a multi-agent approach in this thesis was motivated by the attractive characteristics and capabilities that multi-agent systems are known to possess (e.g. autonomy, heterogeneity, reactivity, and flexibility); as noted in Chapter 3. We perceived that such systems are convenient for managing the distributed business processes of the port supply chain. The reasons for that is summarised in the following.

First, as described in Chapter 3, Multi-Agent Systems (MAS) are an intelligent distributed approach that is appropriate for applications that are modular, decentralised, changeable, ill-structured, and complex (Jennings, 1996). These properties of the appropriateness of using agent technology can be reflected by the nature of the port community, as discussed in Chapter 2. The port community can be viewed as possessing these properties since it is:

- **Modular:** Each decision maker in the port community and the resources used for executing tasks can be seen to have its own set of state variables that are distinct from the environment.
- **Decentralized:** A set of members (actors) in the port community can be decomposed into stand-alone processes, each capable of doing useful tasks without continuous direction by the other member(s)/process(es).
- **Changeable:** The structure of the port community is affected by changes in that continuously new ships are entering, whilst others are leaving, with each ship having its own unique demands and configurations. In addition, the cargos that move through the port community are of various types that need different handling and clearance processes.
- **Ill-structured:** Complete information on all the processes for managing the port community is not achievable due to the use of a number of different systems that are probably not connected to each other.
- **Complex:** The port community domain is considered to be a complex system with a large number of interacting entities and uncertainties.

Second, cargo transport through the port supply chain is a multi-stage process. It is logically and geographically distributed, involving a variety of processes. The processes have different constraints, and might use different models and methods for management and scheduling. Therefore, a distributed and autonomous system seems a more convenient solution rather than centralised, complex, and non-autonomous solution.

Third, the multi-agent approach supposes the existence of a set of heterogeneous agents that integrate and optimise a range of scheduling objectives related to different processes, and can adapt to changes in the environment while still achieving overall system goals. Local autonomy allows the agents to take the responsibility for carrying out local dynamic scheduling, taking into account local objectives, real-time information, and information received from other agents. Agents co-operate in order to find a globally good schedule that is able to effectively react to real-time disruptions, and to optimise the original goals whilst minimising disruption caused by unexpected events occurring in real-time.

Fourth, multi-agent systems provide the foundation for the creation of an architecture that possesses capabilities of reduced complexity, reduced software development costs, high modularity, high flexibility, and improved fault tolerance.

In addition, multi-agent systems provide unique capabilities that are necessary for a middleware in distributed environment such as the port community. These capabilities are:

- Robustness: the use of a multi-agent approach in distributed and autonomous process management systems is characterised by improved reactivity to disturbances and have a potential to solve dynamic problems more efficiently. Local autonomy allows the agents to respond locally to local variations, increasing the robustness and the flexibility of the system. The malfunctioning of a process is kept at a local level, which facilitates the detection and recovery from failures and minimisation of their impact on the whole working environment.

- Reliability: multi-agent systems maintain functionality despite individual component failures.
- Inherent flexibility: the inherent modularity and distribution of multi-agent systems allow rapid response to new system requirements through the addition of new modules and units or reconfiguration of existing ones.
- Open and dynamic structures: it is possible to dynamically integrate new agents (software, hardware, resources, tasks, etc.), remove existing agents, or upgrade agents independently as new functions are needed without disrupting the links previously established and re-initialising the work environment.
- Speed-up and efficiency: agents operate asynchronously and in parallel, which results in computational efficiency.

5.3 The ABMPCM Framework

5.3.1 Overview

The ABMPCM is proposed as an agent-based solution to the problems of managing the distributed collaborative business processes in maritime port communities. The ABMPCM framework is based upon a group of agents (an agent community) that collaborate together in order to provide port community management functionalities. In addition to agents, a group of knowledge repositories is employed to provide essential knowledge necessary for agents' decision making. The framework of the proposed ABMPCM is illustrated in Figure 5-1.

The functionalities of the ABMPCM are provided through five modules. Each module comprises agent(s) which execute functional behaviours to provide a specific facility to the port community members or customers. Namely, these modules are: the Request Handling Facility (RHF), the Collaborative Planning Facility (CPF), Process Management Facility (PMF), Monitoring and Visibility Facility (MVF), and Disruption Management Facility (DMF). A description of each module is introduced below in sub-sections 5.3.2, 5.3.3, 5.3.4, 5.3.5, 5.3.6 respectively.

The necessary knowledge for agents' decision making is provided through four knowledge repositories in the ABMPCM framework. Namely, they are: the Business Process Repository, Business Process Execution History Repository, Key Performance Indicators (KPI) Repository, and Negotiation Policy Repository. A description of each repository is introduced below in sub-sections 5.3.7, 5.3.8, 5.3.9, 5.3.10 respectively.

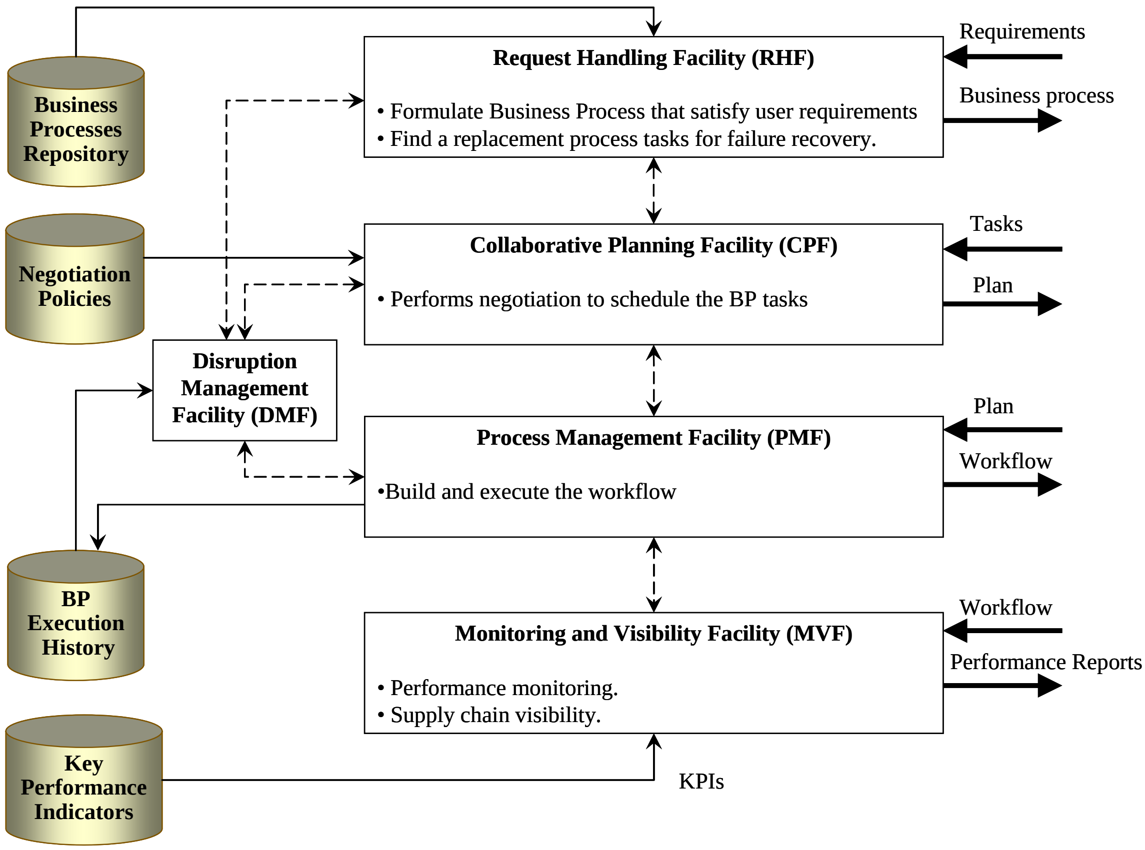

Figure 5-1: The ABMPCM Framework.

The interaction between the modules of the ABMPCM is carried out through FIPA-ACL messages. The FIPA-ACL is a well-specified and much-used agent communication language (ACL) adopted by the Foundation for Intelligent Physical Agents (FIPA)[1]. In each module of the ABMPCM, a Responsible Agent (RA) carries out the interaction through a set of request/response FIPA-ACL messages. The format of a message is a set of key values written in FIPA-ACL (FIPA, 2002a). The content of the message is expressed in a content language, such as the FIPA Semantic Language (FIPA-SL) (FIPA, 2002b), and content expressions can be grounded by referenced ontologies.

[1] The Foundation for Intelligent Physical Agents (FIPA) is an international organization that is dedicated to promoting the industry of intelligent agents by openly developing specifications supporting interoperability among agents and agent-based applications. (See: http://www.fipa.org)

The FIPA-SL content language is a human-readable string-encoded content language (i.e. a content expression in SL is a string), based on an S-Expression syntax. FIPA-SL is a logical language with a prefixed syntax. It consists of a first-order predicate calculus language, extended with additional modal operators to represent cognitive attitudes of agents (beliefs, uncertainties, intentions) and the occurrence of actions (FIPA, 2002b).

The particular format of messages is compliant with that defined by the FIPA-ACL message structure and each message includes the following fields (FIPA, 2002a):

- The *sender* of the message.
- The list of *receivers*.
- The *performative* (i.e. the communicative act) indicating what the sender intends to achieve by sending the message. For instance, if the performative is REQUEST, the sender wants the receiver to perform an action; if it is INFORM the sender wants the receiver to be aware of a fact; and if it is a PROPOSE or a CFP (Call for Proposals), the sender wants to enter into a negotiation.
- The *content* holding the actual information to be exchanged by the message (e.g., the action to be performed in a REQUEST message, or the fact that the sender wants to disclose in an INFORM message, etc.).
- The *content language* indicating the syntax used to express the content.
- The *ontology* indicating the vocabulary of the symbols used in the content.

- Some additional fields, such as conversation-id, reply-with, in-reply-to and reply-by, are used to control several concurrent conversations and to specify timeouts for receiving a reply.

In the ABMPCM, the FIPA-ACL messages are exchanged according to a structured protocol. The FIPA Request Interaction protocol is used to structure a conversation between agents in the ABMPCM. As depicted by the sequence diagram in Figure 5-2, the FIPA Request Interaction Protocol allows one agent, the Initiator, to *request* another, the Participant, to perform an action. The Participant processes the request and makes a decision whether to *accept* or *refuse* the request.

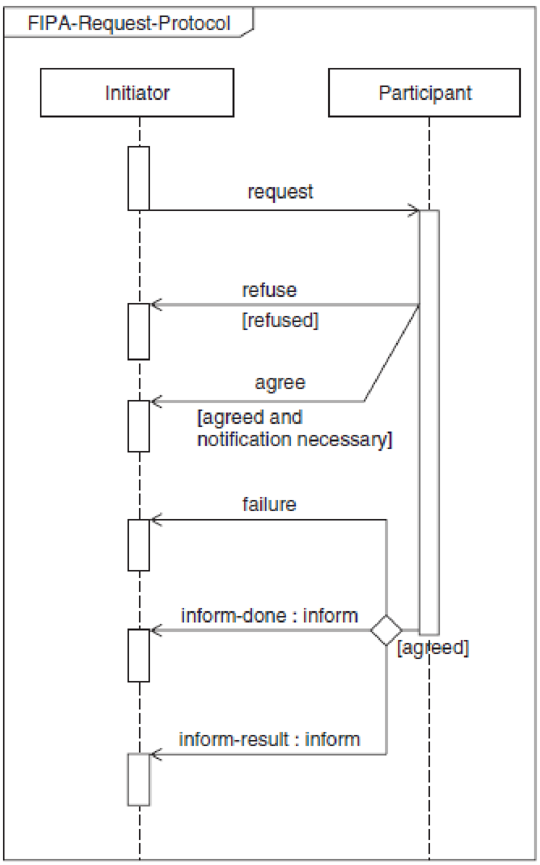

Figure 5-2: The FIPA Request Interaction protocol.

The following example shows the usage of the FIPA-ACL. Suppose that the Responsible Agent at the RHF (RA-RHF) requests a plan (plan1) for business process “bp142” from the CPF, then it could send the following message to RA-CPF:

```
(request
    :sender ra-rhf
    :receiver ra-cpf
    :content (action planning-service
                (plan bp142 (user actor1)))
    :language fipa-sl
    :reply-with plan1
    :protocol fipa-request
)
```

The receiving RA-CPF agent working on behalf of the CPF can confirm the request. The RA-CPF agent may have problems to understand the user “actor1”. In this case, the agent can send a request message back to agent RA-RHF asking to determine the user “actor1”. In the other case the RA-CPF agent knows the user “actor1” because this information is already stored in the business process repository. Then, the RA-CPF agent would send the following agree message:

```
(agree
    :sender ra-cpf
    :receiver ra-rhf
    :content ((plan bp142 (user actor1))
    :language fipa-sl
    :in-reply-to plan1
    :protocol fipa-request
)
```

The ABMPCM will act on behalf of the user (i.e. an actor in the port community) to manage and control a business process that satisfies the user’s requirements. The modular architecture of the ABMPCM allows a port community member or consumer to selectively outsource one or more tasks to the corresponding modules of the ABMPCM, or alternatively, to request a comprehensive service of process management. Figure 5-3 illustrates the interaction of the proposed ABMPCM with port community actors (organizations) and the existing Port community Systems (PCSs).

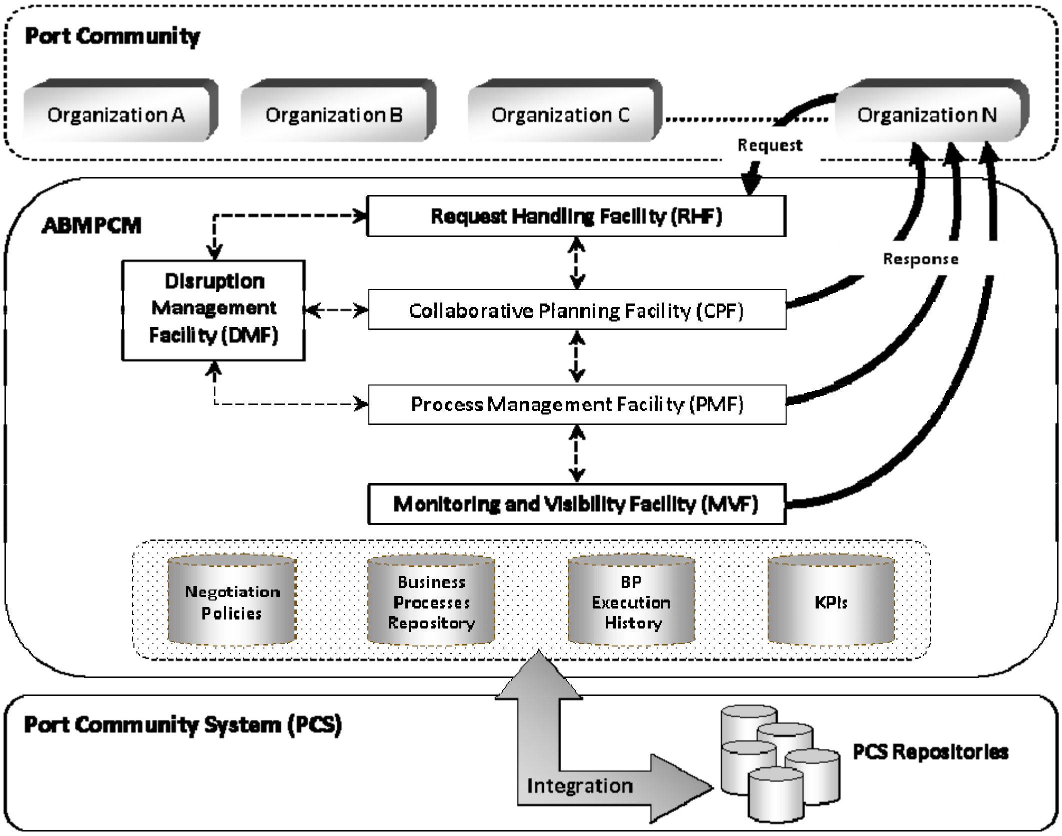

Figure 5-3: The Interaction between ABMPCM, Port Community Actors, and PCSs.

The design of the ABMPCM framework was inspired by concepts of autonomic management, from the autonomic computing community. Autonomic Computing (Ganek and Corbi, 2003) has emerged as a solution for dealing with the increasing complexity of managing and tuning computing environments. Computing systems that feature the following four characteristics are referred to as Autonomic Systems:

- **Self-configuring -** Define themselves on-the fly to adapt to a dynamically changing environment.

- **Self-healing -** Identify and fix the failed components without introducing apparent disruption.
- **Self-optimizing -** Able to find the most efficient way to distribute and perform tasks within the system.
- **Self-protecting -** Protect themselves from attacks by managing user access, detecting intrusions and providing recovery capabilities.

We perceive our proposed ABMPCM as an autonomic manager since it enables the business process to manage itself with minimum human intervention. This aspect in the ABMPCM is described in Section 5.4 below.

In the following sub-sections, we describe functionalities of each component of the ABMPCM in further detail.

5.3.2 The Request Handling Facility

The Request Handling Facility (RHF) finds a suitable business process for the customer of the port community based on some specified selection criteria. It accepts specifications describing requirements (i.e. what does the user need to do?). Then, it formulates a business process that satisfies the user requirements. The business process formulation is carried out through customizing a process template retrieved from the *"Business Process Repository"* (see sub-section 5.3.7 below5.3.7 below). The RHF, then, generates the specification of the customised business process using a formal language (e.g. BPML). This specification is composed of a set of selected activities arranged in the order of execution. Finally, the RHF returns that specification to the Collaborative Planning Facility (CPF).

The proper formulation of the user requirements as well as the business process that satisfies these requirements is a critical success factor because it is the roadmap for other components of the middleware to do their job successfully. Therefore, the requirements should be described using a formal language. The specification of the language for formulating the user requirements is outside the scope of this research work; however, this language should be specified using XML (eXtensible Mark-up Language) and ontology.

In addition to its primary functionality (i.e. formulation of a suitable business process), the RHF is used to find a replacement process task(s) to recover a failure that might arise due to disruptions during the execution of a workflow. The RHF communicates directly with the *"Collaborative Planning Facility"* to initiate the next step for scheduling the process activities for executing.

5.3.3 The Collaborative Planning Facility

The Collaborative Planning Facility (CPF) coordinates the activities of the business process among port community players. The CPF takes an ordered list of activities of the business process and the negotiation policy of the players of the business process in consideration. The CPF employs a group of agents which negotiate through a communication process in order to reach a mutually accepted agreement on an execution plan for the business process. The CPF performs the negotiation locally and generates a schedule (plan) for the execution of the business process activities.

The policies specify the context of the negotiators, their goals, constraints, preferred values and priorities of the negotiable issues that may influence the decision process. Negotiation policies are stored in the *"Negotiation Policy Database"* from which stored policies can be retrieved for negotiations between pairs of port community players.

The control of coordination could be decentralized or centralized. The decentralized control is the coordination of activities among multiple self-interested port community players that make decisions independently. Therefore, the business process performance is a result of the actions and decisions of individual port community players. On the other hand, centralized control is the coordination of activities by a single party that decides on and coordinates the actions of individuals, to maximize the system performance. Decentralized control is in the literature also referred to as distributed rational decision making (Sandholm, 1999) or just Multi-Agent systems (Jennings *et al.*, 1998; Sandholm, 1999; Wooldridge, 2005).

The CPF is designed as an agent-based system for business process activities coordination. The coordination of activities among self-interested agents can be done either via negotiation (direct communication), or indirectly by means of a Contract Net Protocol (Wooldridge, 2005). Direct communication means that agents contact each other directly to negotiate. A contract net protocol is the allocation of tasks by means of an auction (Wooldridge, 2005).

In the port community environment, direct communication is the way that current manual systems work at the moment. Therefore, it is more likely that negotiation is an acceptable approach for coordination among port community players. However, the Contract Net Protocol could be employed in some parts of the business process, for example, in the selection of an inland transporter.

5.3.4 The Process Management Facility

The Process Management Facility (PMF) takes an ordered list of selected tasks with the necessary input parameters for each of them, and generates a Business Process Execution Language (BPEL) (OASIS, 2006) specification for workflow orchestration. The workflow can be returned to the port community member or consumer to be executed locally or can be executed by the PMF.

The PMF provides workflow composition and execution services independently of the other modules in the ABMPCM. The PMF defines workflows with check-points for exception handling and monitoring purposes and executes them locally as the BPEL workflow engine. Centralized execution allows better handling of the exceptions and can be designed to re-invoke services with fewer constraints in the case of failure.

5.3.5 The Monitoring and Visibility Facility

The Monitoring and Visibility Facility (MVF) provides two main functions, namely, process performance monitoring and port supply chain visibility. For process performance monitoring, the MVF takes the workflow specification and the Key

Performance Indicator(s) (KPI) for each process step as input and measures KPIs for each step of the workflow during its execution.

The MVF measure performance data for each activity in a workflow as it is executed. It analyses the performance data, verifies KPIs by comparing target and actual performance, and then creates a performance report that could be sent to the concerned port community customer (e.g. the workflow executors or the process manager).

Regarding port supply chain visibility, the MVF is the supply chain visibility platform (SCVP) that can be used to integrate information from the physical layer, the transaction layer, and the governance layer to enhance supply chain visibility. This information contains data on the times that the cargo should be moving, times that it should be stationary, times that it is being inspected, times that it should have arrived at some location, etc. Matching the business process plan (i.e. what is expected) with workflow execution (i.e. what actually happened) will result in the type of insight that is required for real-time visibility of the chain. Deviations from the expected will raise alerts that need to be followed up by immediate actions.

Furthermore, the MVF plays a role when an exception is detected during workflow execution; it notifies the Disruption Management Facility (DMF) to initiate corrective actions.

Moreover, the MVF helps to build the Key Performance Indicators Repository from the actual performance data. The Key Performance Indicators Repository will enable further analytical studies on the performance of the port community aiming for business process improvement.

5.3.6 The Disruption Management Facility (DMF)

The Disruption Management Facility (DMF) is responsible for handling exceptions (i.e. unexpected conditions) that may occur during workflow execution. It provides the functionality of initiating corrective actions when an exception is detected during workflow execution. Exceptions occur due to disruptions that arise during workflow execution; for example, when the containers that are reported on the planned loading list are not available to be loaded on board of the vessel. In addition, exceptions in the workflow arise when a specific KPI exceeds a predefined (planned) threshold. In such situations, the DMF receives a notification from the MVF indicating that the current workflow is disrupted.

To carry out corrective actions, the DMF first communicates with the PMF in order to gather information about the current workflow instance. Then, it consults the Business Process Execution History repository in order to find out how similar exceptions were compensated in previous workflows. Accordingly, it works in collaboration with the RHF, the CPF, and PMF to recover the workflow. It requests the RHF to find a replacement for the failed activity in the business process, the CPF to negotiate and schedule the modified workflow, and the PMF to build and execute the revised (modified) workflow with the replaced (new) activities.

5.3.7 The Business Process Repository

The Business Process Repository is a specialized repository that contains data about the inter-organizational business processes that take place in the port community. In this section, we give a description of the functionalities that the Business Process Repository provides as well as the sort of the process-specific data that it holds.

We distinguish two process types that can be identified in the Business Process Repository, these are:

- The *port community specific process* which is a process that is designed to describe inter-organizational activities in the port community supply chain.
- The *template process* which is an abstract standard process that can be reused and adapted to develop port community specific processes.

The Business Process Repository provides specialized functionalities that are specific for business process models in addition to the functionality that is commonly provided by repositories of database management systems. In addition, it provides specific services for managing objects as opposed to data; such as, version management and configuration management.

The process-specific functionalities that the Business Process Repository provides include: functionality to assist with lifecycle management of business processes, functionality to help maintain consistency between the private view of business processes (which is the view that organizations have internally on their business processes) and the public view on business processes (which is the view on those parts of business processes that companies want to make visible publicly), and functionality to assist with configuration management of business processes as they are composed of sub-processes and tasks.

In addition to these process-specific functionalities, the Business Process Repository of the ABMPCM supports a series of basic functions to effectively manipulate the processes that it stores. These functionalities include the functions to create, update and delete processes or parts of processes. Furthermore, other functions to retrieve the required process according to some criteria are provided. A process is retrieved by searching the repository to get processes that match criteria that are given as keywords or specified using a process-specific query language, such as, the Integrated Process Management - Process Query Language (IPM-PQL) (Choi and Jang, 2007) or the query extension of the Business Process Model and Notation (BPMN-Q) (Awad, 2007).

Regarding the process-specific data that the Business Process Repository in the ABMPCM holds; it includes data to describe the following:

- The activities that are performed in the context of a process.
- The control-flow relations between activities.
- The information that is used and changed during the execution of a process.

- The physical resources that are required to execute activities in a process, including human resources.
- The description of who is authorized to perform which part of a process.
- The definition of how the performance of a process should be monitored.

5.3.8 The Business Process Execution History Repository

The Business Process Execution History Repository is a specialized repository that contains data about instances of the business processes that take place in the port community. It provides functionalities similar to those provided by the Business Process Repository as described in sub-section 5.3.7 above. The only exception is that the functionalities to manipulate (create, update, and delete) the stored processes are prohibited in this repository.

In the ABMPCM, we distinguish two process types that can be identified in the Business Process Execution History Repository, these are:

- A *process instance* which is an execution of a process for a specific user of the ABMPCM.
- The *process log* which is historical data about previous executions of process instances.

The Business Process Execution History Repository holds process-specific data that describe the following:

- The activities that are performed in the execution of a specific process instance.
- The control-flow relations between activities of a specific process instance.
- The information that is used and changed during the execution of a specific process instance.
- The physical resources that are used to execute activities in a specific process instance.
- The performance measures that are monitored during the execution of a specific process instance.

5.3.9 The Key Performance Indicators (KPI) Repository

The Key Performance Indicators (KPI) Repository stores the values of the performance measures for each activity in the business processes of the port community supply chain. It stores both target (planned) and actual performance measurements.

As indicated in sub-section 5.3.7 above5.3.7 above, the Business Process Repository stores the performance indicators that should be monitored during the execution of a business process. The corresponding set of target (planned) performance values is stored in the KPI Repository. The target performance is used by the Monitoring and Visibility Facility (MVF) of the ABMPCM for monitoring the actual performance of the business process as described in sub-section 5.3.5 above.

Measuring the port community supply chain performance indicators plays an important role to identify the gap between targets and the actual process outcome(s). Therefore, the actual performance measurement for each activity in the business process is recorded by the MVF in the KPI Repository. The performance data in the KPI Repository will enable further analytical studies on the performance of the port community supply chain aiming for business process improvement.

Organizations involved in container transport are progressively concerned with performance with respect to the supply chain as a whole, and with respect to environmental and social impacts of transport. Performance indicators that appear in container transport are, amongst other things, expressed in terms of efficiency, time compression, flexibility, resource utilization, carbon emissions, and security. Important examples of environmental and social performance measures are emissions, noise levels in populated areas, and safety and security of cargo and containers. In this regard, we consider the performance dimensions of container transport in terms of people, profit, and planet. Examples of some performance indicators regarding people, profit, and planet are depicted in Table 5-1 (van Baalen *et al.*, 2008).

Social Performance (People)	**Economic Performance (Profit)**	**Environmental Performance (Planet)**
Legality	Cost	Pollutant emissions
Job Satisfaction	Throughput time	Release of toxic materials
Security and Safety	Productivity	Consumption of energy
Network congestion and use of public infrastructures	Reliability	Consumption of virgin raw material
	Responsiveness and Flexibility.	
Noise and visual Intrusion		Waste Disposal

Table 5-1: Example performance Dimensions (van Baalen *et al.*, 2008).

The issue of identifying the KPIs of the port community supply chain is outside the scope of this research programme. However, it is worth mentioning that traditional performance indicators are necessary as a tool to evaluate organizational performance, but they are too specific and hence not always useful to measure inter-organizational performance or supply chain performance. The various dimensions of performance that can be distinguished in the port community supply chain and the involvement of multiple stakeholders considerably affect the way that performance can be monitored and controlled. Therefore, the issue of firm-specific versus supply chain wide performance indicators should be considered.

5.3.10 The Negotiation Policies Repository

The Negotiation Policies Repository contains business policy specifications for each member of the port community. Business policy information is essential for the coordination process that is carried out by the Collaborative Planning Facility (CPF) as described in sub-section 5.3.3 above. During the coordination process, each concerned port community member provides the CPF with a well-defined policy specification. These policies are processed and stored into the Negotiation Policy Database enabling easy retrieval and updates of policies.

Basically, policies are sets of high level governing rules which define actions to be taken when certain conditions are met. Negotiation policies specify the context of the negotiators; their goals, constraints, preferred values and priorities of the negotiable issues that may influence the decision process. In the CPF, this information is used to initialize the strategy for an autonomous agent which conducts the coordination on behalf of a port community member. A detailed description of the coordination process in the CPF, which is a core component of the ABMPCM, is introduced in Chapter 6.

The CPF accepts policies from the port community members that define negotiation parameters namely: context, goal, issues, preferences, constraints, and other necessary metadata using a domain specific schema. The rules in a policy specification can be defined as blocks of if-then-else clauses that can be grouped together using “And”, “Or”, “Not”. Relational operators (=, >, <, etc.) are generally used to express rules as equations. Each negotiation policy defines parameters for different negotiation contexts and can refer to the context of the other party with regards to the specification of negotiation rules. The negotiation parameters are explained below in further detail.

Context: Three types of context information is referred to in the policy specification; *Negotiation Context* refers to the specific negotiation context (role of the party, a brief service description etc.) to which the policy applies; *Consumer Context* refers to the party’s own context information, and *Other Context* refers to the context of the other negotiating party.

Goals: Each party in the negotiation has a goal. High level goals such as maximize profit, obtaining long or short term contracts, and targeting large reputed companies are based on the strategic business plans of the negotiator. Each party in the negotiation can specify multiple goals. Based on the higher level goal, detailed policy specifications can include further details in the form of rules regarding how to achieve the goal.

Issues and Options: Issues are the negotiable parameters in a service offering and options are the different values that can be taken by the negotiable parameters. The negotiating parties need to specify the best and worst acceptable values for each issue and option, the normalized preference weights to indicate the relative importance of the issues, and optionally a threshold value. If during negotiation the threshold value is exceeded, then the negotiating party is contacted for decision making. For example, to make an offer containing a response time that is beyond the threshold value specified by the service provider, the party is contacted to verify feasibility of the new offer.

Constraints: Constraints are combinations of conditions that define unacceptable values of multiple issues and are declared explicitly in a policy specification to rule out options and narrow down the acceptable set of solutions. For example, a constraint can state a combination of values of different issues is not acceptable. The maximum time for negotiation is optionally set as a constraint.

Preferences: The negotiating parties can define relative priority values for the different goals when more than one goal applies, and for issues and options to facilitate the trade-off between the different issues. Typically numerical normalized weight values are used to indicate the relative importance of a set of issues where the sum of the weights equals to 1.

Metadata: The metadata contains information about the port community actor who invokes the service of the CPF, and the name and date of the policy specification for easy reference to retrieve the policy from the Negotiation Policy Repository if required.

5.4 The Readiness of the ABMPCM for Web Services

As described in Chapter 3 (sub-section 3.5.2), the port community is considered as a target enterprise for the deployment of SOA (Service Oriented Architecture (Baltzersen *et al.*, 2009). This means that all parties involved in the community would develop interfaces to expose business logic as services within their legacy systems; and they would also provide mechanisms to consume these services. Furthermore, Web Services have been introduced as one of the advanced technologies that will facilitate provisioning of widespread, sophisticated and interoperable port e-services (electronic services); this will give ports the opportunity to offer similar services (Marianos *et al.*, 2007).

Therefore, the potential deployment of Web Services in the port community was in mind during the design of the proposed ABMPCM framework. The integration of Web Services is considered in the design ABMPCM framework in several aspects as follows:

1. The UDDI (Universal Description Discovery and Integration) will be used to store semantic mark-up information about port community service offerings, properties, parameters, and return values for service discovery.

2. The facilities of the ABMPCM can be provided as Web Services. In such a configuration, each facility of the ABMPCM would be invoked as a service independent of other facilities. In addition, it would be possible to distribute the facilities, where each facility would be outsourced and provided by a trusted service provider.

3. When the port community members develop interfaces to expose business logic as Web Services; the business process could be, partially or in whole, composed of a collection of Web Services forming a so called "Web process". In that case, the ABMPCM will facilitate the management of Web Services discovery (through the RHF), composition (through the CPF), and execution (through the PMF and MVF).

4. In the **Request Handling Facility (RHF)**, the specification of the language for formulating the user requirements would be specified using XML (eXtensible Mark-up Language) and a Web Service Ontology. It is useful to use the work on Web Ontology Language for Services (OWL-S)[2], which supplies a core set of mark-up language constructs for describing the properties and capabilities of Web Services in an unambiguous and computer-interpretable form. Based on standards like OWL-S, the formal language would be used to match semantic service selection criteria against the information in the UDDI.

5. In the **Process Management Facility (PMF)**, the PMF takes an ordered list of selected Web Services with the necessary input parameters for each of

[2] http://www.w3.org/Submission/OWL-S/

them, and generates a Business Process Execution Language (BPEL) specification for Web process orchestration.

6. In the **Monitoring and Visibility Facility (MVF)**, if the workflow is composed of Web Services (Web process), the MVF will take the Service Level Agreement (SLAs) and performs SLA compliancy checking in order to perform process performance monitoring.

5.5 Autonomic Aspects of the Proposed ABMPCM

As noted in sub-section 5.3.1, we perceive the ABMPCM as an autonomic manager for the business processes of the port community supply chain. The ABMPCM provides autonomic business process management capability to the port community supply chain. In this section we describe this aspect of the proposed ABMPCM.

The ABMPCM provides a framework for the management of the distributed collaborative business process in maritime port community. It enables the process to manage itself without human intervention. Upon providing a valid requirement specification from the port community customer (user), the business process is configured, scheduled, executed, and monitored by the ABMPCM framework with self-healing and implicit self-protecting capabilities.

The self-configuration feature is manifested by Request Handling Facility (RHF), Collaborative Planning Facility (CPF), and the Process Management Facility (PMF) through selecting, scheduling, and thereby, executing the process.

The self-healing feature is enabled jointly by the PMF, the MVF, and the Disruption Management Facility (DMF) by monitoring and analyzing the performance data and deciding about the recovery action. Once the action plan is in place, all the components in the ABMPCM may be involved to execute the action.

The self-optimization feature is enabled by the CPF. It tries to optimize the workflow at scheduling time by allowing independent activities and sub-processes to run in parallel.

The self-protection feature for the process is strongly tied to the self-protection feature of the ABMPCM as each module in the ABMPCM provides a broker service for managing the business process. Therefore, trust and access control schemes are considered for all the components of the ABMPCM.

5.6 Example Business Scenario

In this section, the functionalities of the ABMPCM are described through an example business scenario from the maritime transport domain. The business scenario is about a consignee (i.e. cargo receiver) who is expecting a container and he would like to plan the trip of the container from the container yard to the final destination. Typically, this will require multiple activities such as container arrival, cargo clearance, and truck reservation to be chained and coordinated together to form a business process.

The ABMPCM can assist the consignee in creating, executing and managing these activities. The requirements of the consignee should be specified appropriately since it is the roadmap that guides the functionalities of the ABMPCM. The consignee provides his requirements with the aid of an intelligent user interface within the RHF (Request Handling Facility) of the ABMPCM. This interface has the knowledge about the services provided by the port community as well as the business rules that control provision of these services. For example, Information that the consignee provide to the RHF include the ship ETA (Expected Time of Arrival), number of containers, cargo type, cargo weight and volume, final destination, preferred date and affordable cost.

Accordingly, the RHF determine the required activities for the process. If a satisfactory activity cannot be found**,** the process is reanalyzed to find an alternative break down of tasks. Finally, The RHF returns selected activities in the order of execution either to the consignee if only the RHF service is invoked, or to the Collaborative Planning Facility (CPF) if the complete ABMPCM facilities are invoked.

For each activity, the CPF gather negotiation policies from both the consignee and the port community actor that will accomplish the activity into the negotiation policy repository (knowledge base). Based on the negotiation policy specifications, the CPF selects appropriate negotiation strategy for each party, conducts negotiation locally and returns a plan to both parties upon successful negotiation. If a negotiation fails, it notifies the consignee to modify his requirements or his negotiation policy. In some cases, the CPF asks the RHF to select an alternative activity. When an alternative activity is found, the CPF does the negotiation and the cycle continues until activity is successfully contracted.

When a set of activities are selected with corresponding Key Performance Indicators (KPIs) for the process, the Process Management Facility (PMF) is called with the consignee choices, which are used as input parameter values for the activities, the scheduled (planned) list of the selected activities, and the process information. The PMF builds a workflow with the selected activities in sequence and the parameter values obtained from consignee preferences or previous activities in the workflow.

Before executing the process, the PMF requests the Monitor and Visibility Facility (MVF) for monitoring the activities. The MVF obtains a performance report for each activity as they are invoked during the execution of the workflow. A missing report indicates a failure of an activity. After a threshold period, if a report is not received the MVF notifies the DMF to initiate corrective actions.

5.7 Summary

This chapter presents our proposed Agent-Based Middleware for Port Community Management (ABMPCM) as an approach for the management of business processes in the port community supply chain. First, we discuss the need for multi-agent systems for managing port community supply chain. We then present our ABMPCM framework and its functionalities. It provides essential management facilities through five facilities. The functionalities of each facility have been described. In the end, the functionalities of the ABMPCM are described through an example business scenario from the maritime transport domain.

In the next Chapter, we describe the Collaborative Planning Facility (CPF), which is a component of the ABMPCM, in further detail.

Chapter 6

Implementation of the Proposed Agent-Based Middleware

6. Implementation of the Proposed Agent-Based Middleware

6.1 Introduction

In Chapter 5, we introduced the framework of our proposed Agent-Based Middleware for Port Community Management (ABMPCM). It comprises five components; the Request Handling Facility (RHF), the Collaborative Planning Facility (CPF), the Process Management Facility (PMF), the Monitoring and Visibility Facility (MVF), and the Disruption Management Facility (DMF). These components will fill the gap in functionalities due to the shortcomings of the existing port community systems, as noted in Chapter 4.

In order to focus on the fundamental aim of this research, which is the enhancement of the performance of business processes in port communities, we chose to implement the Collaborative Planning Facility (CPF) component of the ABMPCM. We have made this selection because the CPF component realises a crucial prerequisite for the required capabilities in information systems to address coordination, as noted in Chapter 2. In this Chapter, we first describe the framework of the proposed CPF, and then we implement it through a practical application.

We propose the CPF for bilateral coordination between the members of the port community in a flexible agent-based framework. The CPF addresses the problems of specification and automation of the coordination of business process tasks with possible user feedback.

The effectiveness of the proposed Collaborative Planning Facility (CPF) is described through a practical application in the domain of inland transport of containers. The inland transport of containers is an inter-organisational process since it involves several port community members (consignee, terminal, inland transporter, and freight forwarder). Therefore, coordination is essential to complete the daily operations of the inland container transport. The implementation focuses on the role of the CPF as a real-time agent-based order assignment system that coordinates the assignment of trucks to containers. We validate the framework with an experimental prototype and agent-simulations. The CPF prototype is discussed in this chapter whilst the agent simulation experiments are described in Chapter 7.

In the sections to follow, we first briefly introduce the inland container transport business domain. This description is largely based on our field observations and on the interviews that we have carried out with companies in the port community. Then we present the CPF framework and its functionalities. Next we discuss our partial implementation of a prototype of the model.

6.2 The Inland Container transport

As noted in Chapter 2, the inland container transport is the transport of containers between the port and a place in the hinterland. Many organizations are involved in this process with different interests, cultures, and core businesses. These organizations include forwarders, rail and barge operators, trucking companies, and terminal operators. In this Chapter, we focus only on trucks as a mode of transport for the inland containers transport.

The process of transferring containers using an inland mode of transport is an integral part of the whole port supply chain. This process takes place as follows:

- The container is picked up at a container terminal; and then the truck leave the terminal with the imported full container to the importers' facilities and warehouses.
- At the importers' location, the container is unloaded, and then the empty container is returned back to the same or another terminal or empty container depots.
- When a need for an export arises at any warehouse facility, either close to the previous importer's facility or elsewhere, another empty truck trip needs to take place from the port to the exporter's facility.
- At the exporter's location, the container is loaded and then it is transferred back to the port.

This logistic pattern is illustrated in Figure 6-1. This pattern is very common in many ports of the world.

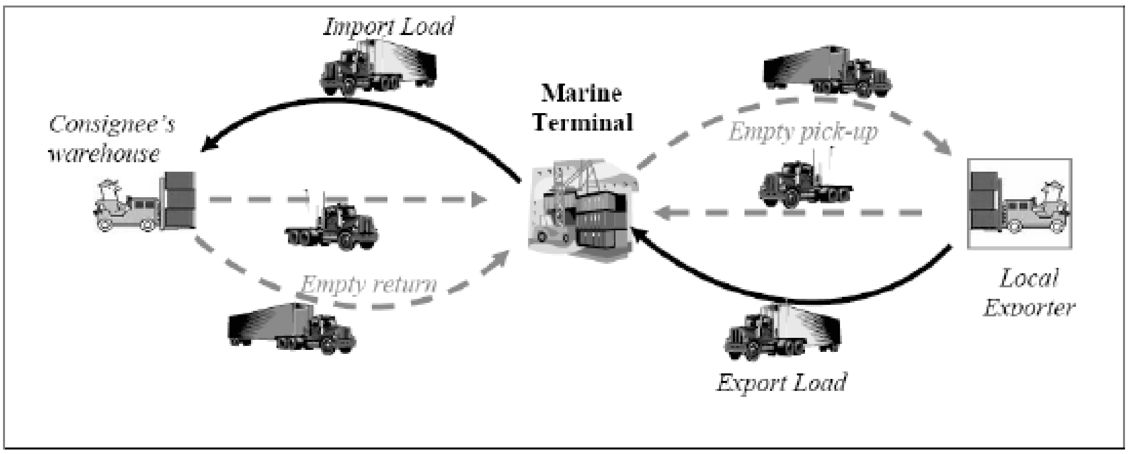

Figure 6-1: The Logistic Pattern of Inland Containers Transport

One important issue to notice in this process is that there is no matching between empty import containers available at one location and customers (exporters) that require an empty container for export at other locations. This comes up due to the absence of collaborations among key players and as a result no information about container status (full or empty) is available. Accordingly, every export-import cycle is associated with two trips with an empty container, these are considered as unproductive trips.

The increase in the number of empty truck trips (unproductive trips) to and from the container terminal causes significant inefficiencies and it leads to the congestion in and around the terminal as well as the roads that feed into the terminal area.

Generally, the management of empty containers requires tackling two main problems. The first problem is the empty storage accumulation at the terminal or empty container depots and the second problem is the increase of unproductive empty trips to and from terminals or empty container depots. These two problems are distinctive although they are interrelated and are attributable to the same driving factors. A solution that addresses the issue of empty storage accumulation by means of any new initiatives introduced at the terminals could have minor effect in reducing the empty trips to and from the terminal. In addition, the port related truck traffic causes not only additional vehicle trips with its associated truck emissions, but also simultaneous delays and congestion at terminal gates as well as roads. Therefore, there is obviously a need to specifically address the issue of excessive unproductive trips made to and from the terminal.

The process of inland container transport is an inter-organizational process since it involves interaction between terminals, inland transporter, freight forwarder, and consignee or exporter. Based on the survey that has been carried out in this study, the interaction between the consignee and the inland transporter can be described in four different phases:

1. The **order placement phase** where the inland transporter receives customers' request to pick-up a container from a container terminal (in case of an import container) and transport it to the customer's facilities, with delivery within a definite time window. Generally, requests are received by fax or email one day before the actual pickup; in addition, last-minute orders sometimes take place by phone. There are no price or timeframe negotiations. Normally, customers are return customers who conduct a large quantity of orders at known tariffs.
2. The **order execution phase** during which the inland transporter announces the pickup of containers. The inland transporter should inform the terminal several hours before the pickup time; accordingly, terminals can prepare for pickup, and arrange their administrative procedures to ease the pickup process. In addition, this phase may include customers' queries about issues regarding the order status. These issues include whether the order has been assigned and picked up already, where the truck is and when can it be expected for delivery, and so on. Generally, this type of queries is carried out by phone.
3. The **physical delivery phase**, where the driver delivers a full or empty container to its destination. In some cases, the truck waits for unloading/loading of the container.

4. The **invoicing phase**, where an invoice is issued to settle the financial status of the customer. An invoice is generally an accumulation of several orders in one invoice, sent after a specified period.

6.2.1 Trucks Planning Process

The actual planning takes place during the second phase (order execution) and the third phase (physical delivery) of the inland transport of containers process which is described in the previous section.

During the order execution phase, the assignment process from orders to trucks is carried out. Normally, the assignment of the first trips is performed one day prior to the actual container delivery. This stage includes a decision on the number of the required trucks. In case the number of orders exceeds the number of available trucks, the inland transporter hires additional trucks from other carriers.

On the day of physical delivery, the second planning stage starts when the trucks finish their first orders of the day. During the physical delivery phase, the planner keeps in contact with the driver in order to monitor the truck location and status. On the other hand, the planner monitors the availability of the containers in the terminal; then the planner assigns convenient trucks to available containers. A schematic illustration of this part of the process is depicted in the activity diagram in Figure 6-2.

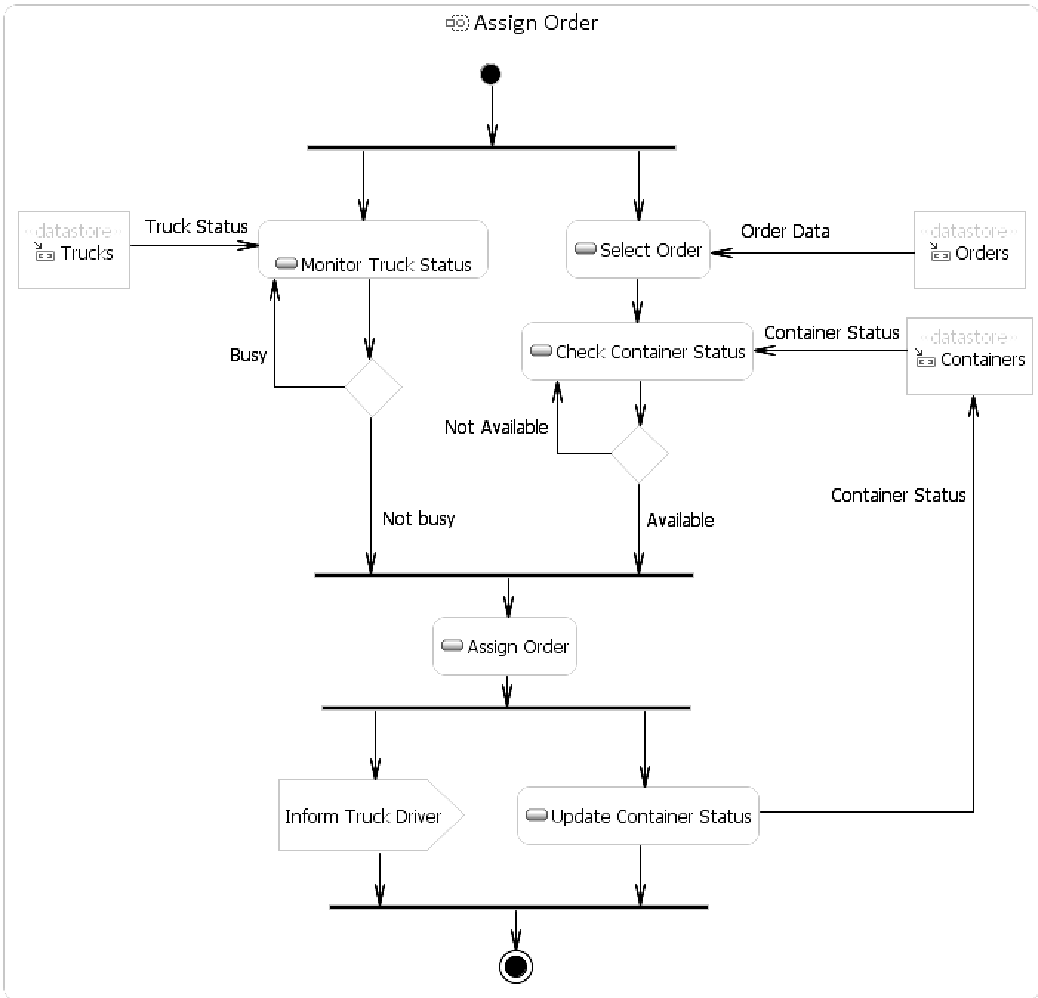

Figure 6-2: Activity Diagram - Assign second (or later) order

The third step is no longer a planning process; rather it is the actual process of transporting containers physically from one location to the other, and back. For an export container, the process would be reverse.

The container status information (present, expected, not available, etc) is essential for the planning process. Before picking up a full container, the inland transporter needs to assure the availability of the container at the terminal. It may happen that the container has not left the ship yet, or it is not released yet due to some administrative issues. The latter is due to administrative procedures, either between the ocean carrier and the customer which generally means non-settlement of payments or because it has not yet released by customs authorities. However, in most of the cases, customers assure that the container is physically present at the terminal before they put an order for picking up the container. Nevertheless, the inland transporter can get container status information through a direct access to the container terminal information system or the port community system. In some cases, this information is provided by phone.

6.2.2 Potential Improvements

A purposeful analysis of the process of inland transport of containers, as described in the previous section, revealed that there are some issues that adversely affect the performance of the process. The surrounding conditions of the process which is characterised by the multiplicity of last-minute placement of orders combined with few ICT-based relationships shapes an environment with a serious amount of manual control and relatively little space for optimisation. The lack of electronic relationships between upstream and downstream partners makes information exchange relatively slow and expensive.

Regarding process optimisation, the lack of prior information about the container status (present, expected, not available, etc) hinders proper planning of the process. This information will only become available sometime during the process, and since only the existing and released containers can be pulled out of the terminal, therefore, the optimization of the process at a reasonable time before operation is hardly achievable. In addition, the large number of daily orders as well as distance to travel per day makes the entire planning of the process impractical.

Despite the fact that container transport is an activity that heavily depends upon external information and coordination, there is a little electronic process integration between the involved parties. Therefore, the process planning greatly relies upon manual operations. Although container terminals have automated systems in place from which a container status check is available; these checks are largely carried out manually by the inland transporter or the freight forwarder.

The activity of assigning a truck to an order is not automated; although, in several cases, it is supported by information systems. This activity needs human intervention by a planner. During the assignment process, the planner combines all the available information and manually assigns a certain truck to a specific order.

In conclusion, areas of potential improvements can be summarized as follows:

- Enhancing the planning of the inland container transport processes by automating the truck assignment activity as well as external information processing tasks. By doing so, a planner will have a larger span of control and allow the possibility of planning more trucks.
- The better truck assignments will result in fewer errors, less unfavourable waiting and crowd at terminal gates, and less empty trips.
- The coordination of the chain that involves the customer (consignee or exporter), the container terminal, and the inland transporter will enable a higher level of optimisation. It no longer solely reacts to (last-minute) orders, but has the potential to anticipate orders. This might result in shorter throughput times, less unnecessary waiting, lower costs (since one can

further optimise resources and routes) and eventually lead to more customers satisfaction.

6.3 The Role of the Proposed ABMPCM

As discussed in Chapter 3, several multi-agent system projects focus on building alternative planning engines to replace traditional architectures. In this regard, agent technology researchers claim that their technologies and designs are more flexible and it can more effectively handle change. On the other hand, Operations Researches community points to the fact that central assignment mechanisms are generally more optimal.

In the context of inland container transport, we perceived the challenge differently. The application of multi-agent system for this domain is not merely an alternative planning tool; it should also satisfy the fundamental need for the real-time task assignment process which is basically an inter-organizational process. The planning of inland container transport is an inter-organizational coordination problem; the planning aspect of the problem is not as complex as its information coordination aspect. Therefore, solutions for such situation should focus on information coordination through communication, rather than focusing only on the optimization aspects. It would be better to make the solution understandable by the human planner**.** As observed from the interviewed companies, no autonomous automated planning system is utilized for planning even in the companies where information systems are extensively used. Instead, they use a decision support system to facilitate the task and the human planner is responsible for taking the final decision. Therefore, we argue that a solution that mimics the human activity (task assignment in this case), and derives solutions through communication, would be understandable and better fits the needs. In this specific area, we presume that the Collaborative Planning Facility (CPF) of our ABMPCM could potentially excel.

6.3.1 The Proposed Collaborative Planning Facility (CPF)

We propose a trusted Collaborative Planning Facility (CPF) framework to facilitate automated coordination of business process activities between members of the port community. The CPF is one of the main components of the ABMPCM framework presented in Chapter 5. We use agent technology for achieving coordination where agents perform the coordination process in the Collaborative Planning Facility (CPF) locally and impartially.

The CPF accepts coordination requests with business policy specifications from the concerned parties and performs local execution of bilateral coordination using intelligent agents. The policy specifies the port community member's preferences and constraints to be used for coordination. Ultimately, a set of agreements is returned to the concerned port community members when successful coordination is achieved. The proposed approach addresses the following issues:

- Members of the port community do not have to provide or maintain their own coordination system.
- The coordination process takes place locally within the CPF framework and thus network delays and insecurities are avoided.
- The CPF allows for feedback from the port community member during an ongoing coordination to reach better decisions.

- Collection and storage of policies within the CPF helps avoid repetitive collection of policy specifications for several coordination processes.

We use a policy-based approach to define evaluation parameters that are necessary for coordination. Basically, policies are sets of high level governing rules which define actions to be taken when certain conditions are met. Each of the concerned port community members provides the CPF with a well-defined policy that contains preferences and constraints specifications. This information is used to initialize the strategy for an autonomous agent, which conducts the coordination on behalf of a port community member.

A notification about the result of the coordination is sent to a coordination manager, which converts the results either into a set of agreements in case of successful coordination, or failure messages to send out to the concerned port community members. At some critical points during coordination, the agents can request an update of consumer preferences or additional information through feedback handler.

The process of coordination is divided into three phases, *pre-coordination*, *coordination*, and *post-coordination*. The *pre-coordination* phase collects the policies and initializes the coordination process by creating customized agents. The *coordination* phase carries out the bilateral coordination process. The *post-coordination* phase creates and sends reply messages to the concerned port community members.

The framework of the Collaborative Planning Facility (CPF) middleware is depicted in Figure 4.4. The CPF provides a trusted facility for coordination through a *Collaborative Planning Facility Interface Agent (CPFIA)*. Each concerned port community member provides the CPFIA with their corresponding policy specifications.

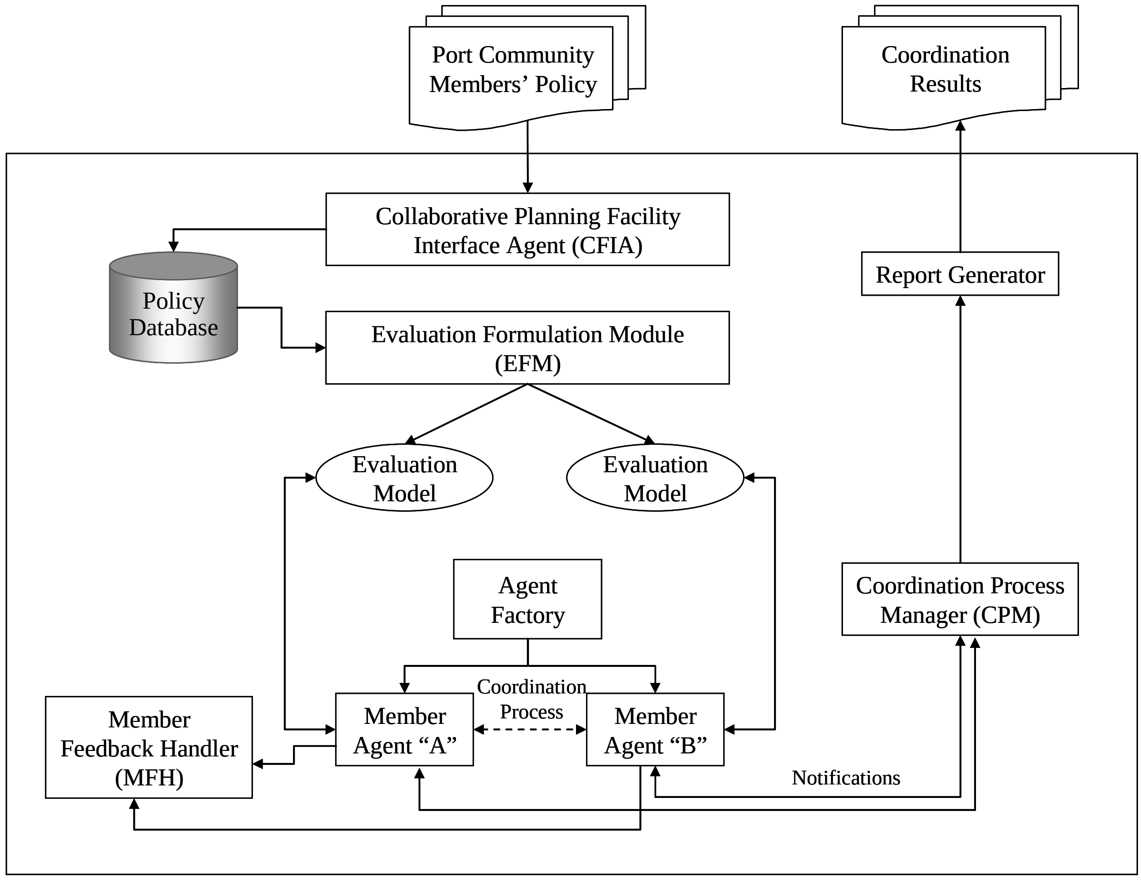

Figure 6-3: The Framework of the Collaborative Planning Facility (CPF).

In the *pre-coordination* phase, the coordination policies received by the CPFIA are processed and stored into a local *Policy Database* enabling easy retrieval and updates of policies. At this point, the *Evaluation Formulation Module (EFM)* uses the policy information from the policy database and initializes an *Evaluation Model (EM)* for each concerned port community member with the strategy and the policy information. The EM applies rules and algorithms to make decisions during the coordination process.

Once the EMs are defined, the *Agent Factory (AF)* is used to create an autonomous agent for each concerned port community member. The agent uses its corresponding EM to independently decide about coordination issues. This approach preserves the privacy of policies provided by each port community member and enables impartial coordination. The *Member Feedback Handler (MFH)* waits for notifications from the agents to communicate with the corresponding parties when the values specified in the policies are exceeded during the coordination process. The concerned parties at this point have the opportunity to redefine the values of the coordination issues, update the constraints, or guide the decision for the next step, which is a powerful feature of the CPF framework.

Upon completion of a successful coordination, the CPF enters the *post-coordination* phase. The *Coordination Process Manager (CPM)* receives notifications from the agents about the status of the coordination process. The CPM sends the necessary information to a *Report Generator*, which generates a set of agreements for the CPFIA to send out to both port community members.

6.4 Prototype Implementation of the CPF

We describe implementation of a partial prototype of the proposed CPF framework to verify our approach to agent-based automated coordination of activities in port community. The usefulness of the proposed Collaborative Planning Facility (CPF) is described through a practical application as a real-time agent-based order assignment system that coordinates the assignment of trucks to containers.

6.4.1 Technology Selection

The main purpose of building the prototype that is introduced in this Chapter is to evaluate the functionalities and design concepts utilised in the CPF. Therefore, we decided to use a toolkit for developing our agent-based middleware rather than engineering the multi-agent system from scratch. Using agent toolkits, and in particular the agent development environment, will allow us to focus on the implementation of the proposed functionalities rather than MAS-specific issues such as agents communication. Therefore, we carried out a survey of the available agent development toolkits in order to select a convenient toolkit to use.

During the last few years, a number of surveys on agent development toolkits have been carried out (Allan, 2010; Nikolai and Madey, 2008; Serenko and Detlor, 2002; Serenko and Detlor, 2003). These surveys revealed that there is a wide variety of agent development toolkits currently available on the market ranging from general agent development platforms to highly specialized tools. The AgentBuilder[1] Website identifies numerous agent toolkits available on the market.

It appears that no single uniform toolkit satisfies the needs of all agent-related applications. Agent toolkits can be categorised into four major groups: mobile agent toolkits (e.g. IBM Aglets[2]), multi-agent toolkits (e.g. Zeus[3] and JADE[4]), general-purpose toolkits (e.g. FIPA-OS[5]), and Internet agent toolkits (e.g. Voyager[6]) (Serenko and Detlor,

[1] http://www.agentbuilder.com/AgentTools/index.html
[2] http://www.research.ibm.com/trl/aglets/
[3] http://sourceforge.net/projects/zeusagent/
[4] http://jade.tilab.com
[5] http://fipa-os.sourceforge.net/index.htm
[6] http://www.recursionsw.com/

2003). We focus on the multi-agent toolkits category for further investigation since our main objective is to build a prototype multi-agent system.

Several different toolkits are available for building multi-agent systems. Examples of multi-agent toolkits that we have identified in our survey are Cougaar[7] (Cognitive Agent Architecture), JADE[8] (Java Agent DEvelopment Framework), MadKit[9] (Multi Agent Development Kit), MAGSY[10], Zeus[11]. A description of these toolkits is introduced in Table 6-1.

[7] http://www.cougaar.org
[8] http://jade.tilab.com
[9] http://www.madkit.org/
[10] http://www.dfki.uni-sb.de/~kuf/magsy.html
[11] http://labs.bt.com/projects/agents/zeus

Multi-Agent Toolkit	Description
Cougaar	It is Java software for facilitating the development of agent based applications that are complex, large scale and distributed. The open source software includes not only the core architecture but also a variety of demonstration, visualisation and management components. It was developed as part of a multi-year DARPA research project into large scale agent systems principally aimed at military logistics.
JADE	It is a framework fully implemented in Java. It simplifies the implementation of multi-agent systems through a middleware that claims to comply with the FIPA[12] specifications and through a set of tools that supports the debugging and deployment phase. The agent platform can be distributed across machines (which do not even need to share the same OS) and the configuration can be controlled via a remote GUI. The configuration can be even changed at run-time by moving agents from one machine to another, as and when required. JADE is completely implemented in the Java language and the minimum system requirement is JAVA version 1.2 (the run time environment or the JDK). JADE is developed by Telecom Italia[13] and distributed under the terms of LGPL-2 (Lesser General Public License Version 2).
MadKit	It is a Java multi-agent platform built upon an organisational model. It provides general agent facilities, such as lifecycle management, message passing and distribution, and allows high heterogeneity in agent architectures and communication languages, and various customisations. MadKit communication is based on a peer-to-peer mechanism which allows developers to develop distributed applications quickly using agent principles. MadKit is free and licensed under the GPL/ LGPL licence.
MAGSY	It is a development platform for multi-agent applications. Each agent in MAGSY has a forward chaining rule interpreter in its kernel. This rule interpreter is a complete re-implementation of an OPS5 system (a rule based language used in expert systems), further enhanced to make it more suitable for the development of multi-agent system applications. MAGSY runs on UNIX, LINUX, SunOS and Solaris systems.
Zeus	It is developed by British Telecommunications (BT) to provide a library of software components and tools that facilitate the rapid design, development and deployment of agent systems. The three main functional components of the Zeus toolkit are an agent component library, agent building tools, and visualisation tools.

Table 6-1: Examples of Multi-Agent Toolkits. [adapted from (Allan, 2010)].

[12] FIPA is an international organization dedicated towards creating and implementing software standards for heterogeneous and interacting agents and agent-based systems. (See: http://www.fipa.org)
[13] http://www.telecomitalia.com/

In order to select a convenient agent building toolkit, we carried out a comparison between the above mentioned toolkits. In this research, we set a number of simple preferable attributes as criteria for the selection of the agent development environment that suite our research context (i.e. purpose, capabilities, and timeframe). These criteria are:

- The domain of application; the proven use of the toolkit in building MASs for a range of various applications in order to assure its convenience for our domain of application.
- The license; the toolkit should be a free open source toolkit with a large user community.
- The programming language; the toolkit preferably supports Java in order to ensure use of our knowledge and experience.
- Compliance with FIPA (the Foundation for Intelligent Physical Agents); the toolkit should be compliant with the FIPA specifications for interoperable intelligent multi-agent systems. Standards developed by FIPA aim to promote the interoperation of heterogeneous agents. Therefore, we decided to select a toolkit that provides FIPA-compliant functionalities in order to assure the interoperability of our prototype with other multi-agent systems.
- The documentation; the availability of a comprehensible documentation and trustworthy support would be an advantage.
- The easiness of learning and using the toolkit.

The comparison is carried out based on the information gathered from the available literature (Allan, 2010; Serenko and Detlor, 2002; Serenko and Detlor, 2003) and Web sites of the identified agent building toolkits. The results of the comparison are detailed in Table 6-2 below. Three toolkits (Cougaar, MadKit, and MAGSY) were excluded from the selection. The reasons for exclusion are: (i) MAGASY is not a java based toolkit; (ii) the three toolkits are not FIPA compliant.

Platform	Primary Domain	License	Programming Language	Operating System	User Support	FIPA Compliant
Cougaar	Multi-agent systems Highly distributed, scalable, reliable, survivable applications Domain independent Large scale distributed, complex, data intensive (can be configured for small-scaled embedded applications	Cougaar Open Source License (COSL) is a modified version of the OSI approved BSD License	Java	Windows 98 Windows NT Windows XP Linux; Mac OS X Java-1.4-capable PDAs	FAQ Tutorials slide shows documentation selected references email support public forums mailing lists	**No**
JADE	Distributed applications composed of autonomous entities	LGPL version 2	Java	Any Java Platform	FAQ mailing list defect list tutorials API Documentation	Yes
MadKit	A generic, highly customizable and scalable platform General purpose multi-agent platform with agent based simulation layer	LGPL for basic libraries GPL for development and non- commercial applications	Java	JVM (Java 2)	FAQ Documentation online forum examples defect list	**Unknown**

Platform	Primary Domain	License	Programming Language	Operating System	User Support	FIPA Compliant
MAGSY	Rules Based Multi-Agent Systems	Free (closed source)	Magsy (production language)	UNIX LINUX SunOS Solaris systems.	Limited documentation Some example (inside installation package) No users support groups No contact even for authors	Unknown
Zeus	Rules engine and scripting environment Distributed multi-agent simulations	Open source (read license)	Visual editors and code generators	Windows 95 Windows 98 Windows NT Windows 2000 Windows XP Linux UNIX-like Oss Solaris	Documentation Author contact	Yes

Table 6-2: Comparison of the Identified Multi-Agent Toolkits.

We carried out further survey for the remaining two toolkits (Zeus and JADE). Accordingly, we decided to use JADE for the development of our agent-based system prototype. The main reasons of excluding Zeus are its limited user support and the lack of comprehensive documentation. In addition of being satisfactory to all the above mentioned selection criteria; JADE shows a number of advantages over Zeus as follows:

- JADE is a well known agent toolkit and it became among the most utilised agent toolkits currently available on the market. Based on the average citations per year, Wohlin (Wohlin, 2007) indicted that the paper in which JADE was presented (Bellifemine *et al.*, 2001) was ranked the 13th among the top twenty most cited articles during the period 1986-2005, and this reflects the popularity of JADE.
- JADE possesses a large user community which is reflected by the very large archive in the electronic forum that contains about 17000 messages discussing problems and ideas concerning the usage and the development of JADE.
- JADE has been employed in many different contexts ranging from academic studies to considerable business applications. For example, JADE has been used for decision support in manufacturing (Cheeseman *et al.*, 2005), simulation of financial markets (Boer-Sorban, 2008), large-scale complex software systems (Chmiel *et al.*, 2005), and asset management systems (Massawe *et al.*, 2010). In addition, several European R&D (Research and

Development) have been based on JADE, for example, E-CAP[14] (E-commerce Agent Platform), IMAGE project (Intelligent Mobility Agent for Complex Geographic Environments), and MONAD[15] (Adaptation Agents for Nomadic Users).

- JADE provides a complete set of documentation for multi-agent systems administration and programming in the JADE environment.
- Learning and using JADE is relatively easy due to the simple and intuitive set of APIs that hide the complexity of the middleware. In addition, tutorials on using JADE are readily available.

In general, JADE brings features and functionalities specifically focused on agent-based system development. An overview on technical and functional features of JADE is introduced in sub-section 6.4.2. An important advantage of using JADE, rather than plain Java, for agent development is the *"agent behaviours*" that JADE provides for agent-based systems developers. A JADE behaviour represents a task that an agent can carry out. Every JADE agent is composed of a single execution thread and all its tasks are modelled and can be implemented as Behaviour objects. JADE provides Behaviours of various types; a short description of the various behaviour types is shown in Table 6-3. JADE agents can switch between different behaviours, and are set up to communicate with other agents via the standardised FIPA protocols.

[14] http://e-cap.sourceforge.net
[15] http://www.cs.helsinki.fi/research/monads/

Type of Behaviour	Short description
Behaviour	Abstract base class for modelling agent tasks.
SimpleBehaviour	A simple atomic behaviour.
OneShotBehaviour	A behaviour that runs just once.
CyclicBehaviour	A behaviour that must be executed forever.
CompositeBehaviour	A behaviour that is made up by composing a number of other behaviours (children).
SequentialBehaviour	A CompositeBehaviour that executes its sub-behaviours sequentially and terminates when all sub-behaviours are done.
ParallelBehaviour	A CompositeBehaviour that executes its sub-behaviours concurrently and terminates when a particular condition on its sub-behaviours is met.
FSMBehaviour	A CompositeBehaviour that executes its children according to a Finite State Machine defined by the user.
WakerBehaviour	Implements a one-shot task that must be executed only once just after a given timeout is elapsed.
TickerBehaviour	Implements a cyclic task that must be executed periodically.

Table 6-3: Various Types of JADE Behaviours

6.4.2 An Overview on Technical and Functional Features of JADE

.JADE[16] (Java Agent DEvelopment Framework) (Bellifemine *et al.*, 2007; Bellifemine *et al.*, 2001; Bellifemine *et al.*, 1999) is a software framework to develop agent-based applications in compliance with the FIPA[17] (The Foundation for Intelligent Physical Agents) specifications for interoperable intelligent multi-agent systems. JADE provides basic middleware-layer functionalities which are independent of the specific

[16] http://jade.tilab.com
[17] http://www.fipa.org

application and which simplify the realization of distributed applications that exploit the software agent abstraction (Wooldridge and Jennings, 1995). JADE has been conceived and developed by Telecom Italia. In February 2000, JADE went open source and was distributed by Telecom Italia under the LGPL[18] (Library Gnu Public License) license (Bellifemine *et al.*, 2007). In order to better facilitate industrial involvement, in May 2003 Telecom Italia Lab and Motorola Inc. created the JADE Governing Board, a not-for-profit organization of companies committed to contributing to the development and promotion of JADE. The JADE Board lists now five members: Telecom Italia[19], Motorola[20], Whitestein Technologies AG[21], Profactor GmbH[22], and France Telecom R&D[23].

JADE is fully developed in Java, therefore, providing a simple and friendly APIs (Application Programming Interface). JADE provides both a development framework and a FIPA-compliant run-time environment. The development framework includes the libraries (i.e. the Java classes) required to develop application agents. It is provided through an Application Programming Interface (API) enabling simple definition of agents and their properties as well as enabling registration with services provided by one or more agent domains.

The run-time environment provides a homogeneous layer that hides to agents (and to application developers also) the complexity and the diversity of the underlying tiers

[18] This license assures all the basic rights to facilitate the usage of the software included in commercial products: the right to make copies of the software and to distribute those copies, the right to have access to the source code, and the right to change the code and make improvements to it.
[19] http://www.telecomitalia.com/
[20] http://www.motorola.com/
[21] http://www.whitestein.com/
[22] http://www.profactor.at/

(hardware, operating systems, types of network, JVM). The run-time environment provides the key services that must be active before agents can be executed. These key services are:

- **The Agent Management System (AMS):** It is the agent that supervises the entire platform. It is the contact point for all agents that need to interact in order to access the white pages of the platform as well as to manage their life cycle. Every agent is required to register with the AMS (automatically carried out by JADE at agent start-up) in order to obtain a valid AID.
- **The Directory Facilitator (DF)**: It is the agent that implements the yellow pages service, used by any agent wishing to register its services or search for other available services.
- **Message Transport Service (MTS):** It manages all message exchange within and between platforms.
- **Agent Communication Chanel (ACC):** It provides an MTS and is responsible for sending and receiving messages on an agent platform.

The main architectural elements of a JADE platform are illustrated in Figure 6-4. The JADE platform is composed of agent containers that can be distributed over the network. Agents live in containers which are the Java processes that provide the JADE run-time and all the services needed for hosting and executing agents. There is a special container, called the *main container*, which represents the bootstrap point of a platform. It is the first container to be launched and all other containers must join to a

[23] http://www.francetelecom.com/en_EN/innovation/

main container by registering with it. The main container is by default named ‘Main Container’ while the others are named ‘Container-1’, ‘Container-2’, etc.

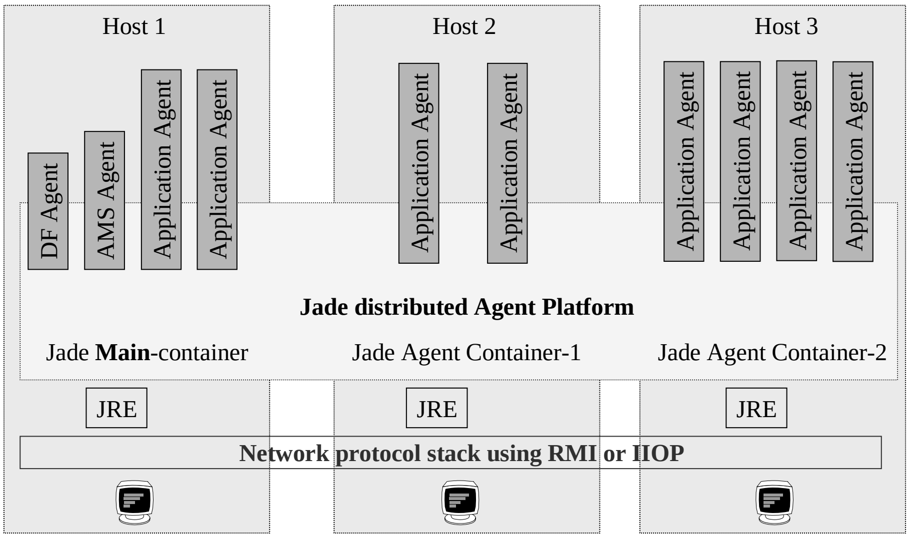

Figure 6-4: The Distributed architecture of a JADE Platform.

Applications with the JADE platform are naturally distributable, and can be hosted across any number of hosts on a network and on any type of JVM, with inter-domain communication enabled using the Directory Facilitators (DFs). Layered on top of this communication platform, JADE dynamically adapts at run-time to choose the most appropriate transport protocol for any agent-to-agent communication. The current protocols supported consist of: JAVA-RMI, JICP (JADE proprietary protocol), HTTP and IIOP. This flexible approach aims to ensure complete support for heterogeneous agent applications.

From the functional point of view, JADE provides the basic services necessary to distributed peer-to-peer applications in fixed and mobile environments. JADE allows each agent to dynamically discover other agents and to communicate with them according to the peer-to-peer paradigm. Agents communicate by exchanging asynchronous messages whose structure complies with the ACL (Agent Communication Language) defined by FIPA. To further support the implementation of complex conversations, JADE provides a set of skeletons of typical interaction patterns to perform specific tasks, such as negotiations, auctions and task delegation. By using these skeletons we get rid of the burden of dealing with all the aspects that are not strictly related to the application logic, such as, synchronization issues, timeouts, and error conditions. To facilitate the creation and handling of messages content, JADE provides support for automatically converting back and forth between the format suitable for content exchange, including XML and RDF, and the format suitable for content manipulation (i.e. Java objects). This support is integrated with some ontology creation tools, e.g. Protégé, allowing programmers to graphically create their ontology.

A very important feature of JADE is the availability of a rich suite of graphical tools supporting both the administration and debugging of applications. By means of these tools, it is possible to remotely control agents, agent conversations can be emulated, exchanged messages can be sniffed, tasks can be monitored, and agent life-cycles can be controlled. These tools are:

- **The JADE RMA (Remote Monitoring Agent)** which is a system tool that implements a graphical platform management console. It provides a visual interface to monitor and administer a distributed JADE platform composed of one or several hosts and container nodes.

- **The Dummy Agent** which is a very simple tool that is useful for sending stimuli, in the form of custom ACL messages using a simple GUI, to test the behaviour of another agent.
- **The Sniffer Agent:** it is used to sniff and display interactions between running agents. This tool is extensively used for monitoring and debugging, or simply documenting conversations between agents. The sniffer allows for the tracking of messages exchanged in a JADE agent platform. When the user decides to sniff an agent, or a group of agents, every message directed to or coming from that agent is tracked and displayed in the sniffer window. The user can view, save, and load, every message track for later analysis. This helps in visualising and understanding the internal processes, which is handy for debugging purposes. It is useful to sniff, monitor and debug conversations between agents.
- **The Introspector Agent:** It is used to debug the behaviour of a single agent. It is used to monitor and control the life-cycle of a running agent and its exchanged messages (i.e. messages in the queue of sent and received messages). It also actively shows running behaviours, which become interesting when agents are equipped with a stack of different behaviours that they execute at the same time. Along with the other tools, this is useful in monitoring the functionalities of the system in order to validate that the system is properly working.

- **The Log Manager Agent:** It is a tool that simplifies the dynamic and distributed management of the logging facility by providing a graphical interface that allows the logging levels of each component of the JADE platform to be changed at run-time.

In terms of its coverage of the FIPA standards, JADE implements the complete Agent Management specification including the key services of AMS, DF, MTS and ACC. These services themselves are implemented as agents. JADE also implements the complete FIPA Agent Communication stack, ranging from the availability of FIPA-ACL for message structure, FIPA-SL for message content expression, plus support for many of the FIPA interaction and transport protocols.

6.5 The Design Phase

6.5.1 The Process of the Prototype Design

There are various methodologies for the design of a Multi-Agent System (MAS); (for an introduction of these methodologies see Wooldridge, 2005). These methodologies are particularly useful when we have a large number of different agents and where there is a lot of freedom in the assignment of roles and responsibilities to agents. In our problem the definition of the agents and their roles and responsibilities is more or less imposed by the business problem. During the design process of our prototype, we adhere to the following steps that most of the methodologies recommend for designing Multi-Agent Systems (MAS):

1. Decomposition of the system into multiple functionalities;

2. Allocation of functionalities to agents;

3. Establishing interaction protocols between agents; and

4. Designing the decision-making capabilities of agents.

We started the design process with a conceptual design which was a first step before the actual building of the prototype. We went through many iterations, different designs and versions in our process. Throughout the different phases we frequently interacted with a trucker company to clarify any ambiguous part of the process.

6.5.2 Agents in the Prototype Design

To define the agents we apply a physical decomposition of the problem, which means that entities in the physical world are represented by agents of the prototype design. Accordingly, we decided that each port community member that is involved in the planning of the inland container transport should be represented as an agent in the prototype design. These agents are mainly employed to monitor the environment of each entity. For example, a truck agent is employed to monitor truck movements and traffic jams, whilst an order agent is employed to monitor container availability and customer preferences. Table 6-4 shows the four main agents and their corresponding role in the system.

Agent	Role
Customer Agent	Negotiate delivery time
	Provides time preferences
	Monitors container arrival
Terminal Agent	Coordinate truck arrivals
	Coordinates container pick up
	Sort container
	Monitors container arrival
	Provide time preferences
Truck Agent	Find best order
	Negotiate with other truck agents
	Monitor container availability
	Monitor traffic jams
	Utilize truck location
Order Agent	Find best truck
	Respect customer preferences
	Monitor truck patterns
	Utilize knowledge about other orders

Table 6-4: Agents and their Corresponding Role in the Prototype Design Phase.

In order to focus on the principal aim of the implementation presented in this chapter which is to show how the proposed CPF can be used for real-time planning of the inland container transport; we decided to implement only two types of agents: Truck agents and Order agents. In addition, this would relatively simplify the development process and made it achievable with the limited resources (time and personnel) of this study.

6.5.3 The Truck Assignment Process

Regarding the business process(s), we model the processes to match the processes as we have observed in real practice in trucker companies. Accordingly, a conceptual level class diagram has been developed for the inland container transport planning process as depicted in Figure 6-5.

In our approach, the truck assignment to an order is carried out in real-time according to a mechanism that takes several issues in consideration. These issues include the order details (minimizing delay of orders), the movements of the truck fleet, reduces empty trips, and potential delays due to traffic jams. In our approach, the prototype design resembles the style that a human planner follows, and then the existing planning process is modelled as agent behaviours. Therefore, the decisions that an agent takes are relatively human-understandable.

In this prototype, only two types of agents were involved, namely, the *TruckAgents* (for each truck in the system) and the *OrderAgents* (for each order in the system). *TruckAgents* follow a specific procedure for truck assignment.

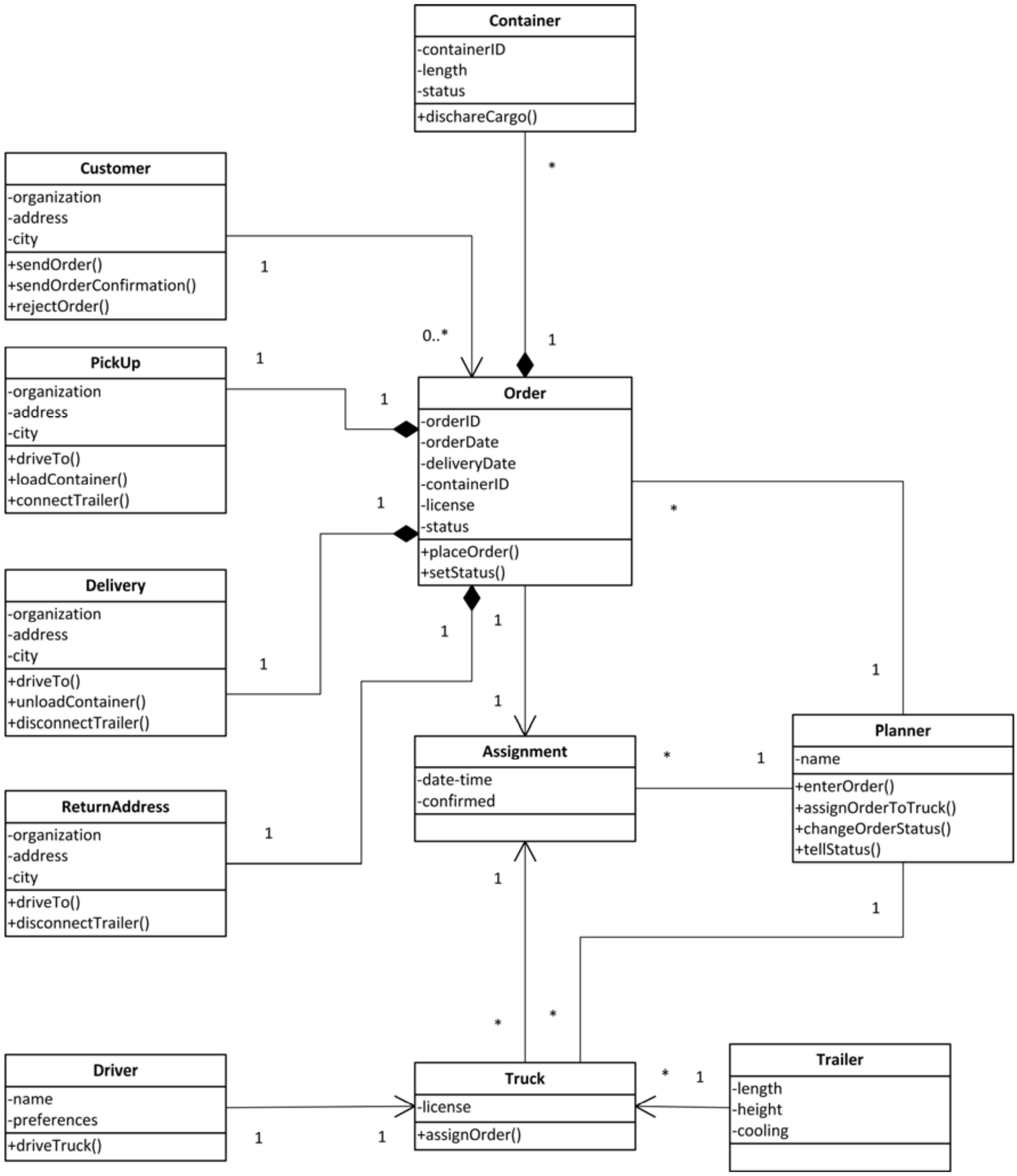

Figure 6-5: A Conceptual Level Class Diagram of the entities in the Container Planning Process.

Each *TruckAgent* monitors its corresponding truck, and as soon as the truck becomes available, the agent begins to search for the next job (order) to execute. This is carried out by contacting *OrderAgents* to get order information, and then the *TruckAgent* computes a score for each order. The scoring mechanism that we implemented in this prototype is described in sub-section 6.5.4 (below) whilst the algorithm (process) for finding a new order is described in sub-section 6.5.5.

The outcome of the scoring operation is an array of orders (jobs) with a list of corresponding scores. This array is ranked according to the score. The *TruckAgent* takes the order that possess the highest rank as a candidate order to be executed, and then it asks for a bid from other *TruckAgents* in the system on that specific order. The scoring incorporates the time trucks are still busy, represented by a truck's ETA (Expected Time of Arrival), and the place where it will become available. If there is no other truck that is more capable to execute this particular order, the *TruckAgent* claims to catch the order and the corresponding truck is assigned to execute this particular order.

In our approach, a *TruckAgent* plays two distinct roles, as an initiator or a participant. As an initiator, a *TruckAgent* corresponds to an available truck and it is looking for a new order. On the other hand, a *TruckAgent* playing the participant role does not look for a new order since it corresponds to a busy truck and it merely responds to CFP (Call for Proposal) that it receives from other initiators. An implementation level AUML class diagram of the initiator role of the *TruckAgent* is depicted in Figure 6-6.

<<Truck Agent>> Initiator, Participant	
Role Initiator	
State Description Truck is empty and trying to find an order	
Action <<Proactive>> Find best order	
Methods	
getCurrentLocation()	setInitatorinDF()
getHomeLocation()	getCurrentOrderDel()
getOrderStatus	getCurrentOrderPickup()
getETA()	getCurrentOrderReturn()
getCurrentOrderID()	getStartTime()
getTruckStatus()	getOrderStatus()
getRate()	
Behaviours InitialBehaviour ListenBehaviour InitiatorBehaviour FindOrderBehaviour UpdateStatusBehaviour SetETABehaviour	
Protocol FIPA Interaction Protocol	

Figure 6-6: An implementation level AUML class diagram of the initiator role of the TruckAgent.

6.5.4 The Scoring Mechanism

As discussed in sub-section 6.5.3 above, a *TruckAgent* selects the suitable order out of the available orders according to a score that it computes for each order. In this section we describe the scoring mechanism that the *TruckAgent* use.

The score (R) that the *TruckAgent* computes for a particular order is composed of a number of score elements (s). For all orders in the system, the *TruckAgent* computes its corresponding scores (R) as follows:

1. Construct a list of orders O:

   $O = [o_1, o_2, o_3, \ldots, o_m]$, where m is the number of orders.

   Only the orders that are ready to be executed are included in the list. Completed orders and those that are in process are excluded from the list.

2. Decide which score elements e to be used as a criteria for the evaluation of alternatives, therefore, the list of score elements E will be:

   $E = [e_1, e_2, e_3, \ldots, e_n]$, where n is number of score elements.

3. For each order in the list O, compute the score (s) corresponding to each score element in E. The outcome will be

$$\begin{bmatrix} s_{11} & s_{12} & s_{13} & s_{14} \\ s_{21} & s_{22} & s_{23} & s_{24} \\ \ldots & \ldots & \ldots & \ldots \\ s_{m} & \ldots & \ldots & s_{mn} \end{bmatrix}$$

4. Look up the weight w for each score element in E, therefore, a list of weights W will be:

$$W = [w_1, w_2, w_3, \ldots, w_n]$$

5. Compute the grand score R for each order in the list O. The matrix then looks like the following:

$$\begin{bmatrix} s_{11} & s_{12} & s_{13} & s_{14} \\ s_{21} & s_{22} & s_{23} & s_{24} \\ \cdots & \cdots & \cdots & \cdots \\ s_{mn} & \cdots & \cdots & s_{mn} \end{bmatrix} \times \begin{bmatrix} w_1 \\ w_2 \\ \cdots \\ w_n \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ \cdots \\ R_m \end{bmatrix}$$

The grand score (*R*) for the (i^{th}) order is calculated according to Equation 1

$$R_i = \sum_{j=1}^{n} w_j s_{ij} \qquad \textbf{Equation 1}$$

Where n is the number of orders to be executed

In the implementation introduced in this Chapter, we used four score elements as criteria for evaluation of alternatives. These score elements are as follows:

a. Delivery Time Window: it is element for the fit to the time period during which the customer prefer to receive the container.

b. Cost Range: it is element for the preferable cost range that the customer is ready to afford for transporting the container.

c. Customer Significance: is an element for the importance of the customer.

d. Empty Distance: is an element the distance of the empty trip to drive to the pickup location of a particular container.

For each element, we used a weight according to the significance of the element. Weights are not equal, for example, in some cases the “empty mileage” or “customer time window” can be expected to outweigh other criteria.

6.5.5 The Find Order Process

TruckAgents in this prototype is responsible for finding a new order for its corresponding truck. Therefore, it starts to search for a convenient order as soon as the truck becomes available. For doing so, the *TruckAgent* follows a course of actions as presented by the sequence diagram depicted in Figure 6-7.

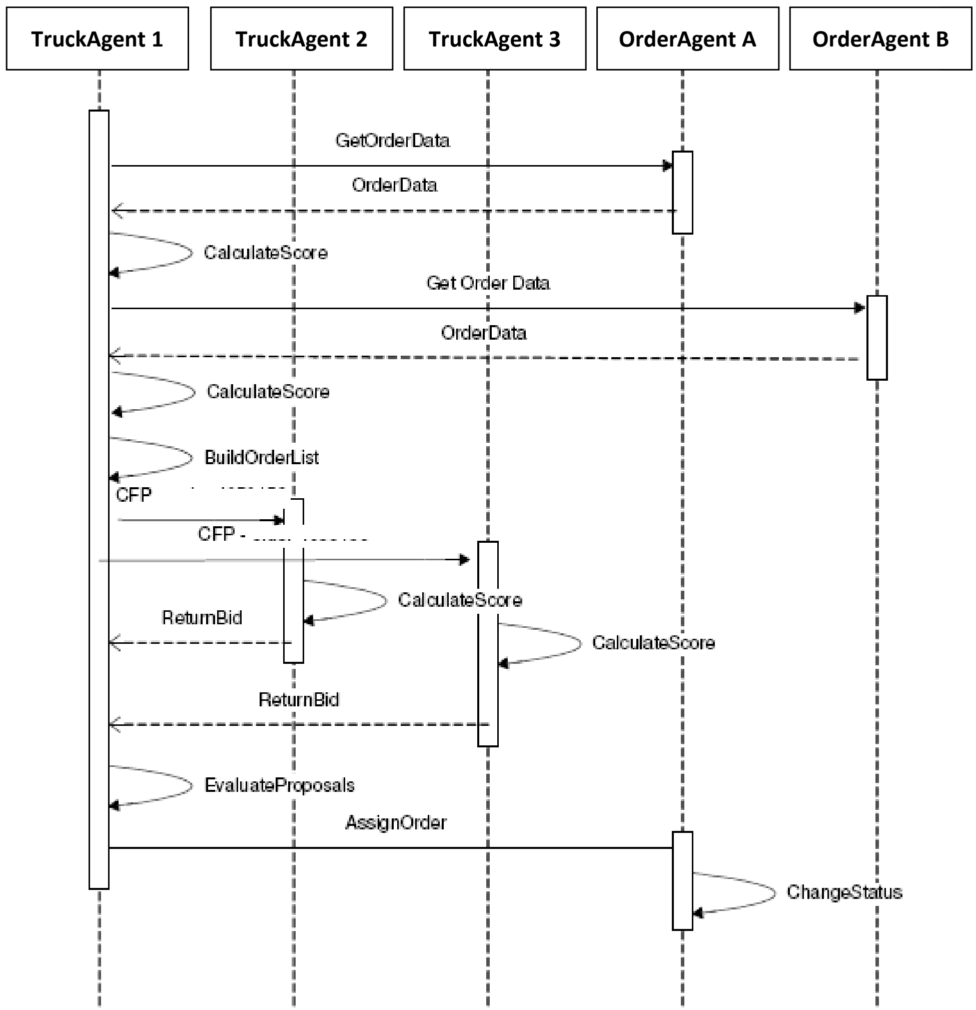

Figure 6-7: The Find Order Process

In our prototype implementation, the CalculateScore process calls four subclasses that each calculates a score for a specific score element (delivery time window, cost range, customer significance, and empty distance). The grand score for the combination of a specific truck and specific order at a certain point in time is calculated according to Equation 1. For our specific implementation, the grand score will be computed as follows:

$$\begin{aligned} GrandScore = & \left(DeliveryTimeWindoWeight \times DeliveryTimeWindScore\right) \\ & + \left(CostRangeWeight \times CostRangeScore\right) \\ & + \left(CustomerSignificanceWeight \times CustomerSignificanceScore\right) \\ & + \left(EmptyDistanceWeight \times EmptyDistanceScore\right) \end{aligned}$$

The *TruckAgent* stores all necessary information of its corresponding particular truck and the calculations that need to be made by the agent. On every status update, the *TruckAgent* calculates the expected ETA (Estimated Time of Arrival) until the moment they will be free to execute the next order. The ETA is utilised for computing scores required to prepare a response for CFPs (Call for Proposal) that might be received from other *TruckAgents*.

6.6 The Development Phase

6.6.1 The Development Environment

As discussed in sub-section 6.4.1, we used JADE (Java Agent Development Environment) for building agents of our prototype. The development process was carried out using Eclipse[24] IDE (integrated Development Environment) which is an open-source multi-platform Java development environment. We used a Toshiba

[24] http://www.eclipse.org

Satellite Laptop with Intel Pentium 1.73 GHz processor and 2 GB of RAM, Windows XP 2002 SP3, and JRE 1.6 to develop the prototype. The scoring mechanism deploys fuzzy (scoring) functions and the FuzzyJ[25] toolkit (Orchard, 2001) was used for development.

6.6.2 Building Agents of the Prototype

The two main agents of the prototype have been developed according to the design phase as described in sub-sections 6.5.2 and 6.5.3. In addition, a control agent (*CoordManager*) has been developed to carry out supportive tasks.

Due to the space limitation of this thesis, a portion of the code that has been developed for the *TruckAgent* is presented in Appendix B to demonstrate the internal of the agent. In particular, the agent behaviour *InitiatorBehaviour* is presented in Appendix C. These two examples show the functionalities of the agent such as the communication with other agents, DF consultation, and internal operations.

In the very beginning of the development process, a *GenericAgent* has been developed as an abstract superclass. This agent possesses the functionalities that should be implemented for each agent of the prototype. For example, the *GenericAgent* has functionality to properly register in the Directory Facilitator (DF) of JADE, to print debug messages, to log messages in an XML file, and to connect to a database. Furthermore, it has behaviour (*LogBehaviour*) for the logging of agent communication messages in an XML file that can be used for any further analysis. In addition, it has termination behaviour (*DoTerminateBehaviour*). The *DoTerminateBehaviour* brings the

[25] http://www.nrc-cnrc.gc.ca/eng/licensing/iit/non-commercial10.html

agent termination as a behaviour that is only executed when all the running behaviours are finished or blocked. It also include deregistration of the agent from the DF.

6.6.2.1 The CoordManager

The *CoordManager* is a control agent that performs managerial tasks. Upon the execution of the prototype, it first starts the *CoordManager* which in turn initializes the environment and creates the *OrderAgents* and *TruckAgents*. The *CoordManager* connects to a database and performs a query on the orders and trucks that are enrolled in the system. Accordingly, it creates a number of *OrderAgents* that match the number of requested orders; it then creates a sufficient number of *TruckAgents*. Furthermore, the *CoordManager* monitors the orders that are scheduled for execution and the trucks that are still active in the system.

6.6.2.2 The TruckAgent

The *TruckAgents* dynamically communicate with *OrderAgents* to find the best order to execute, and then it negotiates on the selected order with other agents. To realize these tasks, *TruckAgents* are equipped with a number of agent behaviours that allow communication with other agents and the execution of the definite tasks such as those tasks related to the search and selection of a new order. The agent behaviours that has been developed for the *TruckAgent* and their corresponding function is shown in Table 6-5.

Behaviour	Functionality
InitialBehaviour	Initial behaviour that starts all behaviours of the TruckAgent.
ListenBehaviour	Cyclic behaviour that listens to messages from other agents
InitiatorBehaviour	Cyclic behaviour that listens to a message that triggers the search for a new order. Accordingly, the TruckAgent starts to play the initiator role. At any point of time, there is only one initiator.
ParticipantBehaviour	Cyclic behaviour that listens and responds to messages that request a CFP (Call for Proposal). These messages are normally sent by other TruckAgent that plays an initiator role.
AssignOrderBehaviour	Behaviour that assigns a specific order to a TruckAgent.
UpdateLocBehaviour	Behaviour that updates the location of the truck that corresponds to the TruckAgent.
FindOrderBehaviour	Behaviour that is needed to find a suitable order – this behaviour is typically started by the InitiatorBehaviour or the ParticipantBehaviour.
DelayFindOrderBehaviour	Waker behaviour that only runs in case a TruckAgent is idle, still searching for a new suitable order, and not yet gone home.
HitTheRoadBehaviour	Waker behaviour that changes the status of the TruckAgent from “waiting” to “moving” after a specific time. This behaviour is useful in the simulation scenarios.
UpdateStatusBehaviour	Cyclic behaviour that updates the status and location of the TruckAgent in the database, in reaction to messages coming from a tracking system
SetETABehaviour	Behaviour that calculates the new ETA (Expected Time of Arrival) of the TruckAgent.

Table 6-5: Agent Behaviours of the TruckAgent

6.6.2.3 The OrderAgent

The *OrderAgents* encapsulate all information about the order. This information includes the preferable delivery time (time range), the customer category, the locations of pickup, delivery and return with their corresponding distances in kilometres, and the status of the container availability at the pickup terminal. The *OrderAgent* is supposed to automatically check the availability of the container from the terminal; therefore, the *OrderAgent* in this prototype is equipped with behaviour to check the container status from a database. The agent behaviours that has been developed for the *OrderAgent* and their corresponding function is shown in Table 6-6.

Behaviour	Role
InitialBehaviour	Initial behaviour that starts all behaviours of the OrderAgent
ListenBehaviour	Cyclic behaviour that listens to messages from other agents
CheckAvailabilityBehaviour	Behaviour that performs an order availability check.
DelayCheckAvailabilityBehaviour	Ticker behaviour that starts the CheckAvailabilityBehaviour after a definite period of time.

Table 6-6: Agent Behaviours of the OrderAgent

6.7 Summary

This chapter presents our CPF framework and an implementation of a partial prototype of the framework to validate the applicability of our approach to intelligent agent-based coordination in a reliable framework. We describe some background on the inland transport of containers as a domain of application. We then present our CPF

(Collaborative Planning Facility) framework and its functionalities. The role of the CPF in the coordination of business process activities has been introduced through the partial implementation of a prototype application.

The evaluation of the CPF is introduced in the next Chapter. In this regard, we employed an agent-based simulation environment which is introduced in the next Chapter.

Chapter 7

Agent-Based Simulation Model

7 Agent-Based Simulation Model

7.1 Introduction

The port supply chain is considered as a special kind of supply chain and it is little studied in the research literature. In the view of SCM (Supply Chain Management), some improvement not only can bring profits to one actor but also to other actors in the port supply chain. In this regard, simulation is the effective means that allows practitioners in the port supply chain to examine improvements, and accordingly decisions can be made wisely.

In this chapter, an agent-based approach for simulation of the port supply chain is introduced. We present a multi-agent framework for simulating port supply chain operations, with the vision of helping to improve overall port supply chain performance and coordination. It can be used to study the port community supply chain and the effect of various business scenarios on the supply chain performance. Its rationalization, characteristics, and the system's architecture that combines agent technology with business rules and business process modelling are presented.

In this chapter, we introduce an assessment of the truck assignment prototype which is proposed in Chapter 6. The purpose of this evaluation is to examine the effectiveness of the prototype as a component of the ABMPCM (Agent-based Middleware for Port Community Management) described in Chapter 5.

In Chapter 6, we introduced a prototype implementation of one component of the proposed ABMPCM (Agent-based Middleware for Port Community Management) described in Chapter 5. In this approach, we argue that agent technology can provide an intelligent solution to the improvement of port supply chain.

In this chapter, we examine the effectiveness of that prototype using the proposed simulation framework. The implementation of a prototype of the simulation framework and the results of the simulation experiments is presented.

7.2 Why Develop a Port Community Simulation Environment

The idea of building a simulation environment stems from the need to evaluate the proposed ABMPCM. The issue of evaluation of information systems has been an active subject of research for many years. However, there is no definite method to properly evaluate an (inter-organisational) agent prototype with all its complexities.

The evaluation methods for information systems can be represented as ranging on a continuum from experimental to naturalistic (observational) methods, as depicted in Figure 7-1 (Kushniruk, 2002).

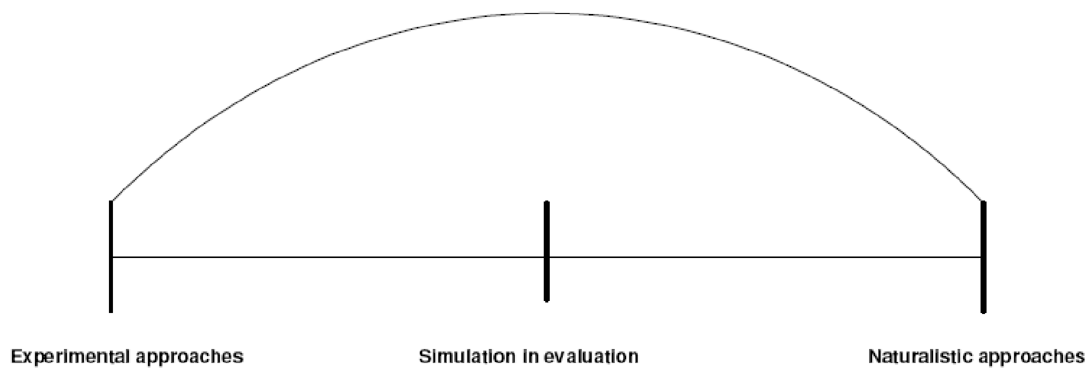

Figure 7-1 Spectrum of prototype validation methods

At one end, we find the experimental approach where an attempt is made to control all factors in the evaluation and vary only a single independent variable (e.g., the presence or absence of information system). There have also been a number of arguments made that a high degree of variable control may be neither feasible nor desirable when testing systems in real-world contexts. Experimental approaches might involve laboratory testing of subjects interacting with computer systems with only one or few aspects (e.g., user interface) manipulated during the testing, with the test being conducted under controlled artificial conditions in a laboratory.

At the other end of the continuum depicted in Figure 7-1 come the evaluative studies that involve observational and naturalistic approaches. From this perspective, activities and interactions with systems may be monitored or recorded, using audio or video. Here the approach is designed to be as unobtrusive as possible, with little or no experimental control.

From Figure 7-1, evaluations that involve simulation techniques are half-way along the continuum. Evaluations that are based on simulation may allow for a high degree of experimental control while also maintaining a high degree of realism in the situations presented to subjects during testing. Accordingly, we decide to employ evaluation using the simulation approach. In addition, this decision is also supported by the fact that the field test of the proposed ABMPCM is extremely complex.

Regarding the proposed ABMPCM, we have tried to employ the observational approach in order to evaluate the middleware. However, the design of the middleware as a multi-agent system and its inter-organisational nature hinder this process. Actors of the port community were reluctant to adopt a solution whose functioning they do not fully understand or trust. In our domain of application, there are possibly a large number of independent actors involved in the business process. To make the evaluation reliable, a considerable number of actors must be in favour of the solution. If not, it will be difficult to get the system running successfully. We decided to develop the simulation environment in order to facilitate the evaluation process of our Multi-Agent system.

7.3 Requirements for Simulating the Port community Supply Chain

In order to model and simulate the workflow of the port community supply chain, it is necessary to build first a functional model of the port community business process, then running the model for a predefined time span to simulate the workflow. We view the simulation model of the port community as having the following characteristics:

- The model should include all the port community actors that are involved in the distributed business process. Actors include, for example, the port authority, the inland transporter, etc. The model also includes the specific workflow and planning the management procedures for each actor.
- The model should include the material flows, the information flows, and the financial flows.
- The simulation model should provide a visual aid for the user to monitor the changing states of the modelled elements. For example, it would be useful to monitor the number of containers in the container terminal in order to examine the flow of containers that goes through the terminal.

- The simulation results should be stored in a persistent storage for subsequent comparison of simulation runs.

7.4 The simulation environment

As described in Chapter 2, a port supply chain is composed of several autonomous or semi-autonomous business entities (actors). Each business entity has its capability and capacity and can be assigned to or take certain types of tasks, according to its organizational roles. Actors of the port supply chain can be viewed as agents while their capability, capacity, and organizational roles can also be modelled as agents. Therefore, we decided to use a Multi-Agent System to simulate the port community supply chain. The Multi-Agent System handles the coordination and the communication among agents to collaboratively accomplish tasks. Each agent is responsible for one or more activities in the port supply chain and each interacting with other agents in planning and executing their responsibilities.

7.4.1 Conceptual simulation framework

In this section, we describe the conceptual simulation model. We first describe the basic architecture of the framework, and then we describe the general simulation structure.

The simulation model presented in this work is an agent-based model, which means that the simulation consists of a group of agents that interact with each other as the system evolves through time. In our simulation model, the course of the simulation is event based. An event is an instantaneous occurrence that may change the state of the system. Event based means that after an event the agent can perform an action resulting in a state transition, after which the state of the agent remains unchanged until the next event. The state of the system is described by the state of all the agents in the system.

The simulations we perform are non-terminating. This means that there is no 'natural' event that specifies the length of each simulation run. We choose to apply the replication/deletion method, which means that we consider several finite simulation runs. Every simulation run starts and ends with an empty system. This implies that the system needs some time to reach steady state (the warm-up period) and at the end of the simulation leaves steady-state (the cool-down period). Both periods (warm-up and cool-down) are omitted from the simulation data during the analysis.

7.4.2 General structure of the simulation model

The agent-based system for simulation of the port community supply chain is composed of five main building blocks, namely, the data sets, the workflow management unit, the port community supply chain unit, the simulation control unit, and the GUI (Graphical User Interface) unit. Figure 7-2. Illustrate the framework of the port community supply chain simulation system. We briefly explain the parts of the simulation model as follows.

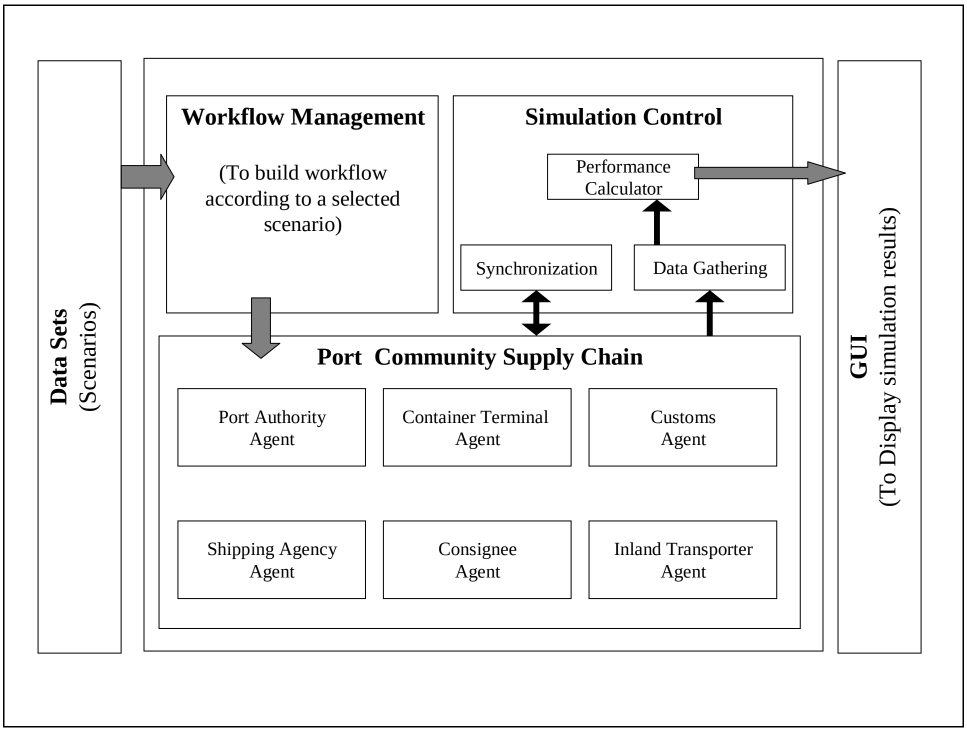

Figure 7-2: The Simulation Framework

- **Data sets.** On initialization of the simulation all information regarding a specific scenario is taken from a data set.
- **Workflow management unit.** It builds a workflow that corresponds to the selected scenario.
- **Port community supply chain unit.** The group of agents that represent port community supply chain. In our model, we use only six agents to represent six actors of the port community.
- **Simulation control unit.** The main function of this unit is to maintain the same view of *"time"* within the simulation environment and to keep all the simulation agents synchronized to each other and to the simulation time. Another function of this unit is to calculate performance measures of each

agent of the port community agents and for the whole supply chain as well. Performance measures are calculated using the simulation results gathered by the data collector.

- **The user interface.** It is the interface through which the performance measures are displayed.

7.4.3 The Generic Agent for Port Community Simulation

We have defined a generic agent for port community simulation, which is then specialized to perform different activities within a port community. For example, a *"container terminal"* agent is different from a *"consignee"* agent or an *"inland transporter"* agent. The group of specialized agents correspond to actors in the port community supply chain. All the agents in our Agent-based system communicate with each other through messages.

In our simulation framework, we regard each actor in the port community supply chain as an intelligent agent that owns a number of resources, can execute business processes (activities), and communicate with other members of the port community. Therefore, we conceptualized the generic agent for port community simulation as an agent that has the ability to:

- Execute processes, thus acting upon the environment
- Manage its own resources.
- Communicate with other agents by receiving and sending messages.

The architecture of the generic agent for port community simulation is depicted in Figure 7-3.

User Interface
Business Process Management (Workflow Management Engine)
Agent Knowledge (Self-knowledge, domain knowledge, acquaintance)
Communication Interface (Language, Protocol, and Ontology)

Figure 7-3: The architecture of the generic simulation agent

- **Communication interface:** this module manages the inter-agent communication. It handles incoming and outgoing messages exchanged between the agents. The communication environment allows the exchange of messages within the port community through an appropriate infrastructure (e.g. message exchange channels) and a protocol.
- **Agent Knowledge:** this module manages the agent's knowledge which include:

  a. **Self-knowledge:** the agent's self-knowledge defines the agent's knowledge about itself and its associated resource. This includes the name, address, and competence of the agent, and the local data used by the agent to execute its tasks.

  b. **Acquaintance-agents knowledge (social knowledge):** the agent's social knowledge contains information about the other agents, which allows it to interact with them. The social knowledge includes the name, address, and competence of the acquaintance agents. This knowledge helps the current agent to select the relevant agents as sub-contractors for processing tasks.

c. **Domain knowledge (expertise):** It contains methods describing the local tasks that can be performed by the agent.
d. **Cooperation knowledge:** this knowledge describes the cooperation protocol used by the agent to cooperate with the agents of the environment.

- **Business Process Management:** In this module, the workflow engine executes processes of an agent's business process model, and thus updates its workflow state.
- **User-interface:** It is the agent's interface to the user. Each agent in the simulation framework communicates with the external world via this interface.

The ability to execute processes is realized with a workflow management engine. Once preconditions and triggering events for some process become true, at some point of time, the workflow management engine will fire the execution of that process. The workflow management engine will operate independently for each agent.

The communication ability is realized by an agent communication protocol and ontology. The communication protocol will define the rules and business interaction of the port community supply chain. Moreover, port community supply chain ontology is required to guarantee a common understanding of the notions used during interactions between port community members.

7.5 Prototype Implementation of the Simulation Framework

In section 7.4, a framework for using a multi-agent system to simulate the workflow in port community supply chain has been presented. The framework possesses some new features that satisfy the requirements for simulating the port community supply chain as described in section 7.3. All of these new features offer new and improved simulation functionality within the distributed environment of the port community. However, a theoretical specification alone is not adequate if the research is to prove useful. Therefore, the purpose of the implementation is to build demonstration software that will act as a Proof of Concept for the theoretical research defined. The prototype is then used for the validation work.

This section will look at both implementation and validation. With regard to implementation, it will define the different aspects of the work - hardware, software - and how the components implementations were achieved.

7.5.1 Simulation Platform Selection

In this section, we introduce the reasons that lead to the selection of JADE to model the port community simulation.

Currently, quite a large number of agent-based simulation tools are available either commercially or as open source. For examples, commercial packages include AnyLogic[1], Arena[2] while the open source packages include Swarm[3], Netlogo[4], and

[1] http://www.xjtek.com/

Repast. These packages provide good modelling elements and various useful functionalities for statistical analysis. However, these packages are inconvenient to model all the characteristics of the workflow of the port community supply chain. Their inconvenience stems from their limited capability to model material flow (i.e. cargo or container in our case) or to integrate the management procedures of an organization. Therefore, the use of a multi-agent paradigm is recommended for building a multipart (complex) modelling environment.

In this work, we regard JADE as convenient toolkit to create and deploy an agent organization which models the workflow of port community supply chain. JADE has a distributed system topology with peer-to-peer networking, and software component architecture with agent paradigm. The network topology affects how various components are linked together, whereas the component architecture specifies the component's expectations from one another (see sub-section 6.4.2 for a description of JADE).

In Chapter 2, we discussed the port community and its main characteristics. A port community can be regarded as a networked organization, or as a *"Networked Enterprise"*. We indicated that the business processes in port community shows typical characteristics of a *"Networked Enterprise"* where control is passed from one actor to the other triggered by events or information flow, and the processes do not belong to one single enterprise.

[2] http://www.arenasimulation.com/
[3] http://www.swarm.org/index.php/Main_Page
[4] http://ccl.northwestern.edu/netlogo/

Accordingly, the characteristics of the simulation model presented in this chapter resemble the decentralized organization of autonomous agents in a dynamic environment. The ease of developing such a model using JADE has been evaluated and JADE was found to be convenient for building decentralized organization of autonomous agents in an open and dynamic environment (Yoo and Glardon, 2009).

In addition, JADE provides the following characteristics:

- **Generic agent architecture:** JADE has no specific agent architecture. It provides a generic agent structure in which different business activities can be encapsulated in an autonomous software agent. All tasks of agents are modelled as behaviour objects. The behaviour implements either the entire task or sub-tasks for implementing more complex ones. This separates the business logic of the autonomous software agent from any other concerns such as agent communication and collaboration.
- **Agent organization modelling:** JADE platform provides several utilities for managing agent collaboration in a dynamic environment. These utilities include the Agent Communication Channel (ACC), The Directory Facilitator (DF), and the notion of agent services.
- **ACL messages semantics:** JADE includes FIPA-ACL and interaction protocols where FIPA agent communication specifications deal with Agent Communication Language (ACL) messages.
- **Dummy Agent:** The dummy agent is useful for the integration of the human user interaction.

The above mentioned features were the reasons for choosing JADE in this work to create the multi-agent model of the port community supply chain. However, since we are aiming to “simulate” the port community supply chain, the main issue is that JADE is not a simulation framework. JADE primarily aims to simplify the implementation of distributed multi-agent systems; it does not provide simulation infrastructure or capabilities. Therefore, it has neither a scheduler nor a notion of a *"clock"* to maintain the same view of *"time"* within the simulation environment and to synchronize agents' behaviours together and to the simulation steps as well. In addition, JADE lacks tools for collecting statistical data for analysing the changes of the agents' internal states. Consequently, we either have to code the functionalities that provide a simulation infrastructure, or to study the possibility of merging JADE with other tools that provides the missed capabilities. We preferred to go through the latter option; therefore, the agent-based simulation platforms were reconsidered in view of merging with JADE.

Many agent-based simulation platforms have been reviewed by several researchers. For example, review and comparison for MASON[5], NetLogo, Swarm, and Repast were carried out by implementing example models in each (Railsback *et al*., 2006). The main focus of the study was the ease of use of the platform, though, the comparison was based on several criteria such as model structures, scheduling capability, execution speed, and other general issues related to agent development. The comparison concluded that Repast (The REcursive Porous Agent Simulation Toolkit) is a good Java simulation platform which can guarantee good performance on execution speed, but it has received poor feedback for modelling agents and their structure. One main feature of Repast is the run-time GUI that allows the user to control the simulation during run-time by pausing, resetting and restarting different models. Repast provides also statistical functions such as graphs which are helpful for simulation monitoring and for real-time monitoring. Through defined variables for monitoring, graphs can be designed during development time. During run-time these graphs are displayed with actual variable's values and its history can be recorded. Repast supports the usage of external statistical programs for monitoring the simulation at run-time. In addition, in Repast any Java class without an agent-based structure can be defined as agent. (See sub-section 7.5.2 below for a brief description of Repast).

Considering these features, Repast was regarded in this work as a good candidate to provide the missed simulation capabilities in combination with JADE.

7.5.2 An Overview on Technical and Functional Features of the REPAST Platform

The REcursive Porous Agent Simulation Toolkit (REPAST)[6] is a free open source toolkit for agent modelling and simulation (North *et al.*, 2007; Collier *et al.*, 2003). Repast focuses on modelling social behaviour, but is not limited to social simulation. Repast was created at the University of Chicago in close collaboration with Argonne National Laboratory. Subsequently, responsibility for the ongoing development of Repast was

[5] http://cs.gmu.edu/~eclab/projects/mason/
[6] http://repast.sourceforge.net/repast_3/index.html

assumed by the Repast Organization for Architecture and Design (ROAD)[7]. ROAD is a non-profit volunteer group led by a board of directors that includes members from a wide range of government, academic, and industrial organizations.

Repast is available in both pure Java and pure Microsoft .NET forms. Both the executable code and source code for Repast are distributed free of charge under a variation of the Berkeley Software Distribution (BSD) license, which does not require user model source code to be released. The Repast user community is large and growing. These users have applied Repast to a wide variety of applications including social systems, evolutionary systems, market modelling, and industrial analysis.

The Repast system is composed of six modules. A component diagram for Repast is shown in Figure 7-4. These modules are as follows (North *et al.*, 2007):

- The **Engine Module** is a fixed module that is responsible for controlling the activities in a simulation. It contains Engine Controller, Scheduler, Action, and Agent components.
- The **Logging Module** is a fixed module that is responsible for recording simulation results.
- The **Interactive Run Module** is a fixed module that is responsible for managing simulation runs under the direct control of a user.
- The **Batch Run Module** is a fixed module that is responsible for completing a set of simulation runs without requiring the direct intervention of a user.

[7] http://repast.sourceforge.net/

- The **Adaptive Behaviours Module** is a flexible module that is responsible for providing adaptive components for implementing agent behaviours.
- The **Domains Module** is a flexible module that is responsible for providing area specific functions.

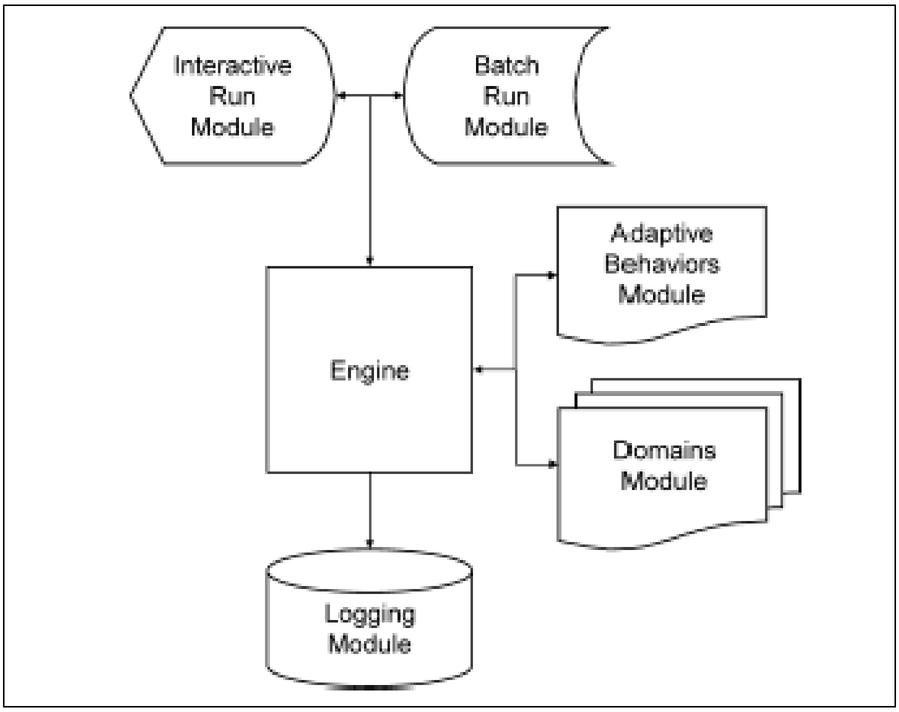

Figure 7-4: The Repast components (North *et al.*, 2007).

There are three implementations of Repast, namely, Repast for Java (Repast J), Repast .NET, and Repast for Python Scripting (Repast Py). In this section, we introduce a description of Repast J implementation which was used in combination with JADE in our simulation model presented in this Chapter

Repast J is a Java language implementation of the Repast specification that provides a software framework for creating agent-based simulations where the common infrastructural abstractions expressed in the framework are implemented as Java classes. The simulation is then created by combining and extending these classes. The Repast J classes are organized into Java packages that express the Repast specification. It is important to note that most of the classes described below are extensible so that features such as custom user interfaces, for example, can be easily created. The packages relevant to the specification can be described as follows (North *et al.*, 2007):

- The **Engine Package**. The classes in the Engine Package are responsible for setting up, manipulating, and driving simulations. This package contains the controller hierarchy by which an agent simulation is started, paused, stepped, stopped, and restarted. This package also contains the classes that make up a discrete event scheduling mechanism, including classes for the *Schedule* itself and the primitive *Actions* that are scheduled. In Repast J, agents can be built using one of a variety of Repast J base classes or they can be constructed using generic objects.
- The **Analysis Package**. The classes in the Analysis Package are responsible for gathering, recording, and charting data. With its gathering and recording responsibilities, this package implements the Logging Module, while the charting functionality implements the graphing portion of the Interactive Run Module.
- The **GUI Package**. The Graphical User Interface (GUI) Package classes are responsible for the graphical animated visualization of the simulation as well as providing the capability to take snapshots of the display and make QuickTime movies of the visualization as it evolves over time.

- The **Parameter Package**. This Package is responsible for defining parameter spaces and iterating through them (i.e., running the simulation with different sets of parameters each time in an automated way).
- The **Adaptation Package**. The Adaptation Package implements the Adaptive Behaviours Module. This package includes tools such as genetic algorithms and neural networks.
- **Domains Module-Related Packages**. Repast J provides several packages beyond the more infrastructural ones described above. These packages provide the basis for constructing particular types of simulations and agents, such as simulations with agents that interact over a grid or torus topology.

7.5.3 Combining JADE and Repast

In this work, we regard JADE as convenient toolkit to create and deploy an agent organization which models the workflow of port community supply chain. To compensate for the missed essential simulation functionalities, we decided to use JADE in combination with Repast to develop the simulation environment that fits our requirements. Although Repast is meant for agent-based simulations and not for developing agent systems and it is not FIPA compliant. In addition, the agents within Repast are not the same kind of agents within JADE.

The idea of combining Jade with Repast has been addressed by very few publications. One approach for integrating JADE with Repast was to combine the JADE agent platform with simulation functions provided by Repast for rapidly developing an environment for the simulation of complex agent model (Yoo and Glardon, 2009). They show that the combination of JADE and Repast makes it possible to construct complex models of multi-agent organisation whose execution states can be observed from users and the global simulation results can be used for performance analysis. In particular the following modifications were added (Yoo and Glardon, 2009):

- Creation of specialized JADE agent (SCAgent) with new capabilities.
- Creation of specialized synchronization behaviour.
- Creation of message scheme that include temporal aspects (*TickerACLMessage*) by modification of the JADE ACL message scheme.
- Implementation of specialized JADE behaviour (*ObservableBehaviour*) in order to use Repast-provided display functions.

In this approach, the Repast framework controls the simulation steps during simulation and allows the possibility to showing changes in agents states that results from the agents activities and communication taking place on the JADE which is the other side of the agent platform. The overall configuration of the composed architecture is illustrated in Figure 7-5 (Yoo and Glardon, 2009). The simulation step control from Repast arrives at *SCAgent* in the JADE environment by passing through the corresponding *agentController.* After activation by its controller (resume method), a *SCAgent* accomplish its activities, which must be done within a simulation cycle, and then suspends itself, while waiting for the next activation from its *agentController*. If an agent includes an *ObservableBehaviour* object, changes in the internal agent state can be observed from the Repast monitoring environment. Every agent on the composed platform is able to communicate using either standard ACL message or extended *TickerACL* message

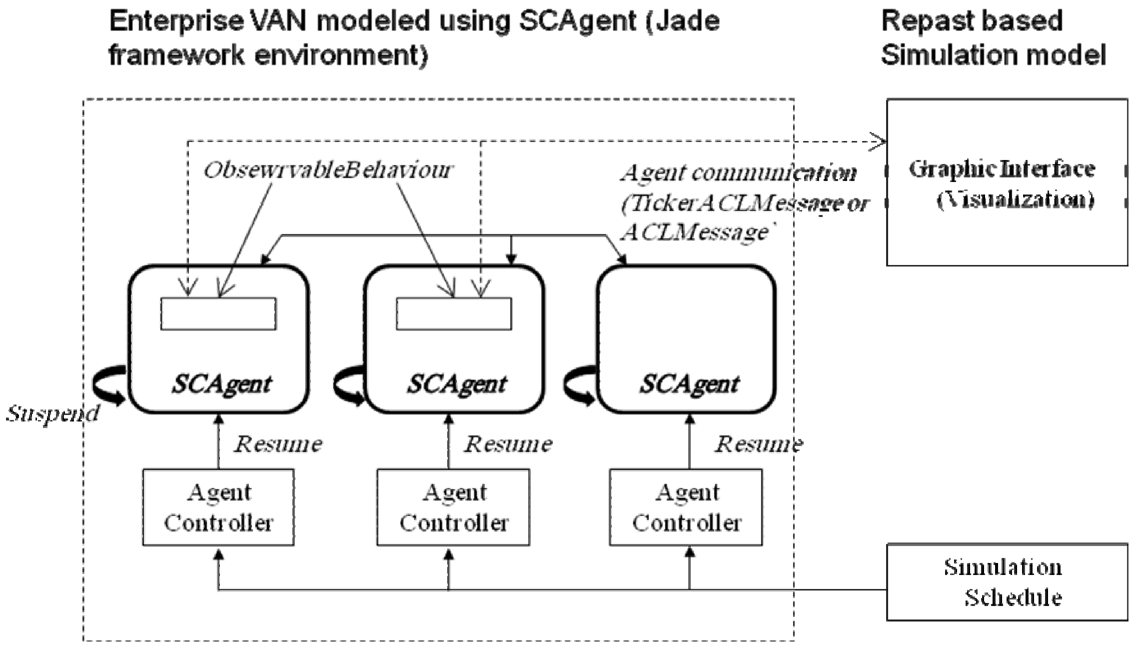

Figure 7-5: The Overall Configuration of the composed architecture of JADE-Repast. (Yoo and Glardon, 2009)

In another approach, integration of JADE with Repast Simphony (Repast S) was employed in the JRep simulation platform (Görmer *et al*., 2011). JRep aims to provide the machinery to model the behaviour and interactions on the agent level (micro level), as well as the instruments to investigate the emerging effects on the overall system (macro level). JRep uses the strengths of JADE and Repast S platforms, by combining their complementary features. JRep platform handles (time) synchronization with the environment, scheduling of agents, and registering of new agents. The functions, flexibility, and scalability of JRep platform was presented and evaluated. An agent-based airport scenario was introduced as a proof of concept for modelling and simulating in the JRep platform; for validating scalability and performance properties, a simple coin flip scenario was used.

In JRep, the main idea to integrate JADE Agents into Repast S is to use an agent wrapper class that is implemented in Repast S. The architecture of JRep is depicted in Figure 7-6. It contains the following components:

RepastAgentRepresentation: Each agent has exactly one representation within Repast S, which is responsible for sensing the environment (Perception) and performing actions. Therefore, the class *RepastAgentRepresentation* possesses the methods perception() and action() (see Figure 7-6). Furthermore, the class *RepastAgentRepresentation* informs agents each new simulation tick.

JADEAgent: The JADE agents can be connected to Repast S (and/or the appropriate RepastAgentRepresentation) directly by references. The communication between agents and the Repast S simulation environment must be possible in both directions. First agents are informed about the new system time of Repast S through the *scheduleAgent()* method of the RepastAgentRepresentation class and then they can act directly on the Repast S platform (see Figure 7-6).

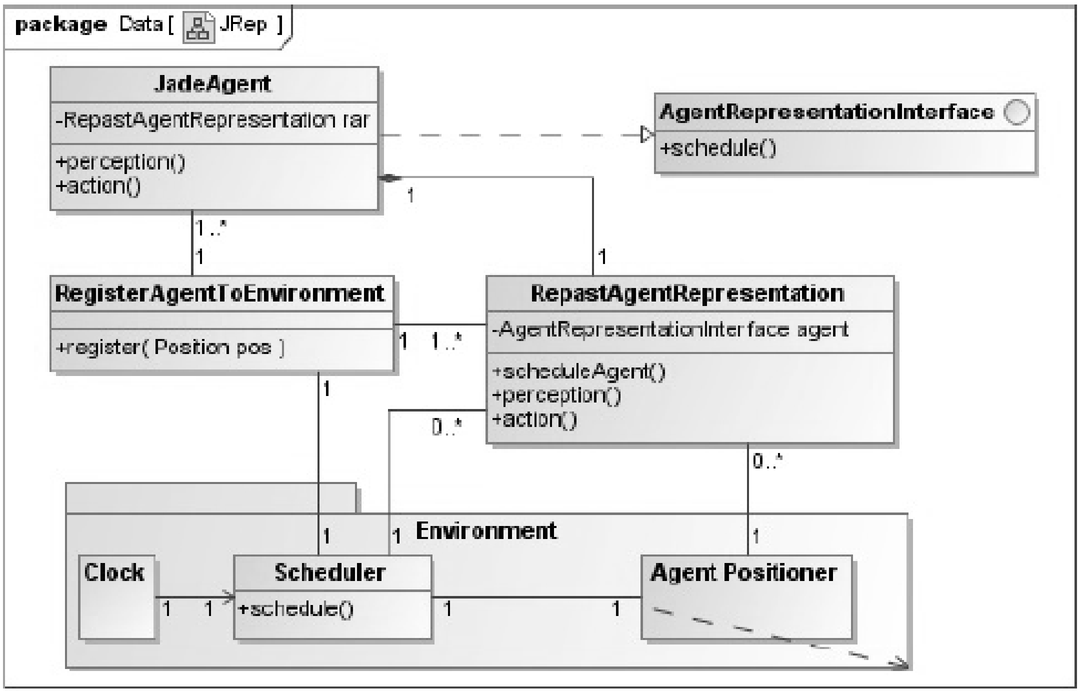

Figure 7-6: Class diagram of the JRep approach (Görmer et al., 2011)

7.5.4 Analysis of the Port Community

A series of interviews have been carried out in order to get insights about the inter-organisational business process in the port community. A process-oriented view was applied, and interviewees were asked to describe their activities in the process of moving the cargo through the port (e.g. which activities are carried out, when information is exchanged, and what information is exchanged). Meeting minutes were prepared for each interview to summarize the discussion. To assure the quality of the interview, copies of meeting minutes were sent to all interviewees for revision and correction as well as amending any missed information. In addition, more focused interviews were conducted with selected managers and users to validate the simulation model developed in this research work.

Interviews had been conducted with professional personnel from companies drawn from the Alexandria (Egypt) community working in the field of maritime transport and logistics and involved in international trade. These companies include shipping lines (7 companies), shipping agents and the customs clearance agents (15 companies), freight forwarders (8 companies), and key persons in the port authority and customs. These companies are the ones that positively responded to the request for holding a face-to-face interview.

The process description from interviews was combined and numerous main processes have been identified from this information. A business process model was developed for each main process that included the relevant actors, the conducted activities, and the information exchanged during these activities. This method was used to clarify the relationships among the actors and the information exchange in the process. The information flow during import is illustrated in Figure 7-7 and the information flow during export is illustrated in Figure 7-8.

The business process models revealed that numerous different activities are carried out upon cargo arrival. Many of these activities are interrelated and require coordination for smoothing the process. When one service is delayed, other services must be rebooked and resources should be reallocated. Delays also entail a risk that the other services could not be available later, which would cause further delays and disturbances to operations (related to cargo or vessel) time table.

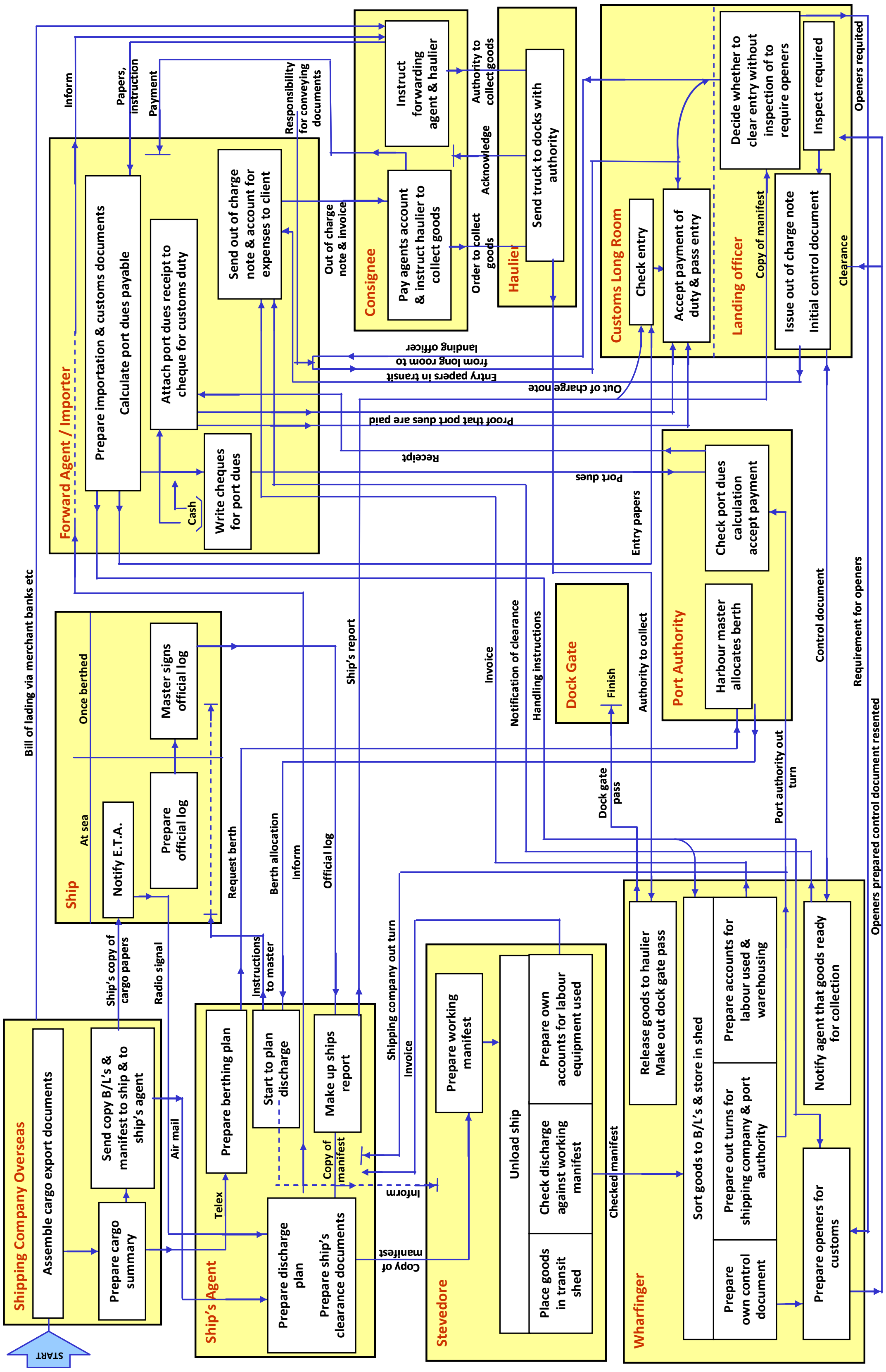

Figure 7-7: Information Flow in the Port Community (The Import Process)

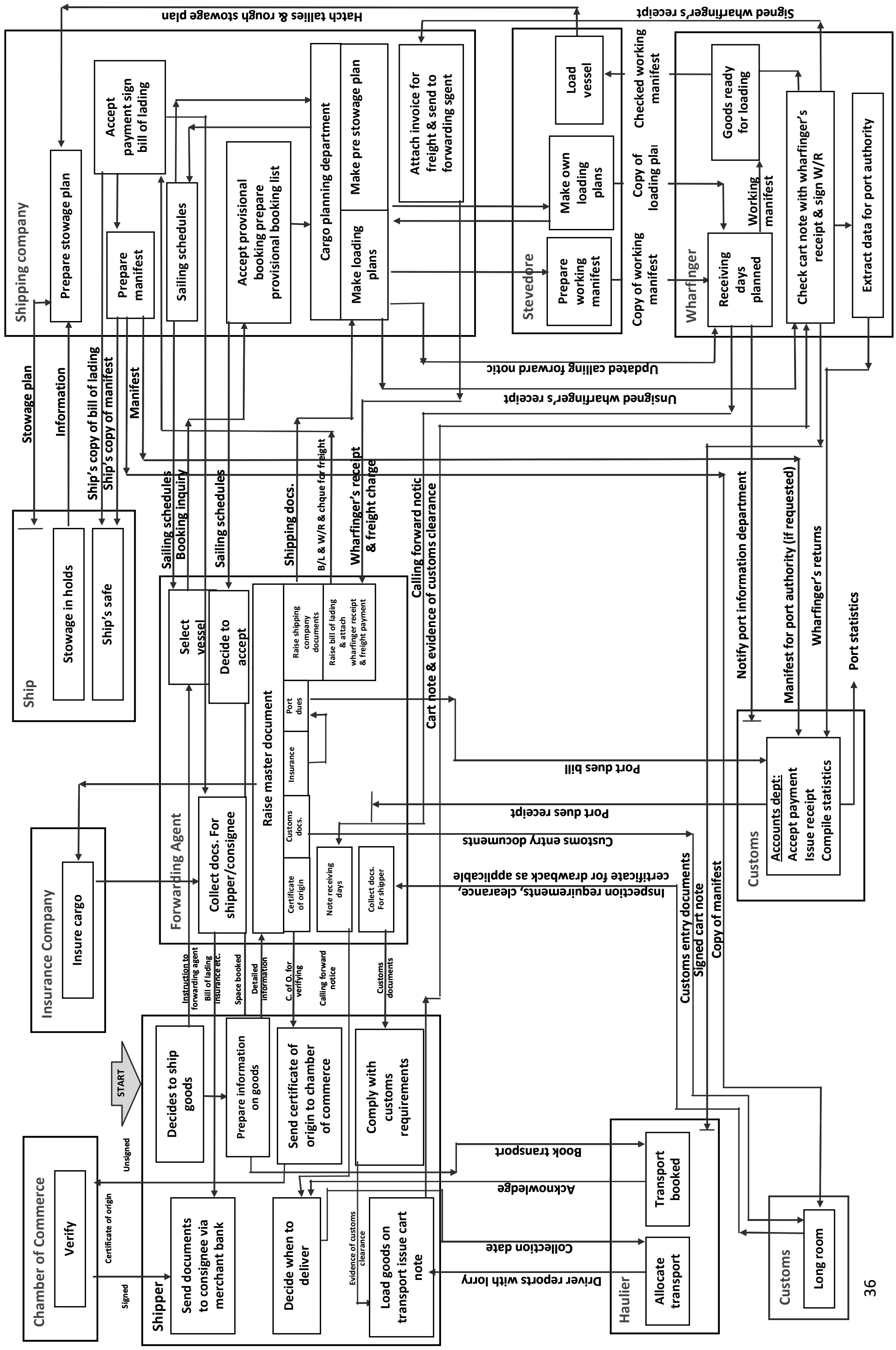

Figure 7-8: Information Flow in Port Community (The Export Process)

7.6 The Design Phase of the Simulation Environment Prototype

7.6.1 The Prototype Overview

With regard to the scope of the implementation, it was decided that it should include all five aspects of the framework:

- Data sets (Simulation Scenarios).
- Workflow management.
- Port community supply chain model.
- Simulation control.
- The user interface.

Each one needs to be a workable part, in order for the whole simulation framework to be implementable.

7.6.2 Simulation Scenario

Based on the business process models that were developed as described in sub-section 7.5.4, we selected the cargo import process for implementation in the prototype of the simulation environment. In this scenario, we consider only the activities that are directly related to the cargo flow. Therefore, we implement a simplified business process that can be generalised for any port community.

In this scenario, the *"simplified business process of import"* was modelled as an agent-based system. The six port community actors that are involved in this business process were modelled as agents. In addition, *"ship"*, *"container"*, and *"truck"* were modelled as Java classes to represent the physical flow of the cargo. To exemplify the information flow, the *"custom declaration"* and the *"custom certificate"* were modelled as Java classes. The tasks that should be carried out by each actor were identified. A description of these tasks is illustrated in Table 7-1.

The simulation model developed in this research work is verified and validated through focused interviews with practitioners in the port community of the Port of Alexandria as described in sub-section 7.5.4 above. Model verification was performed to assure that the *"simplified business process of import"* in terms of agent behaviours is correctly reflected in the simulation model. Model validation was performed to assure that the model is an accurate representation of the real-world system.

The simulation model developed in this research work is verified and validated through two phases. First, the flow diagram of entities, the decision making and functioning stages of each agent in the system are verified. Next, the model output is examined for reasonableness. The simulation model is found to operate as intended.

Port Community Actor	Tasks
Shipping Agent	• Send berthing application of expected ships to the PORT AUTHORITY. • Receive end of operation notification from the CONTAINER TERMINAL. • Inform the CONSIGNEE that the cargo arrived and ready for clearance.
Port Authority	• Receive berthing application from the SHIPPING AGENT. • Ask the CONTAINER TERMINAL to schedule expected ships. • Inform the SHIPPING AGENT about scheduled time for expected ship. • Receive ships according to schedule. • Inform the CONTAINER TERMINAL about ship arrivals to start operation.
Container Terminal	• Receive requests from PORT AUTHOITY to schedule an expected ship. • Schedule expected ships. • Inform PORT AUTHORITY about schedule of an expected ship(s). • Operate ship(s) for loading and discharging containers. • Inform the SHIPPING AGENT about end of ship operation.
Custom House	• Receive request for customs certificate from CONSIGNEE. • Issue a “certain number” of custom certificates.
Consignee	• Receive end of operation notification from CONTAINER TERMINAL. • Send truck request to INLAND TRANSPORTER. • Receive cargo arrival confirmation from INLAND TRANSPORTER
Inland Transporter	• Receive a truck request from CONSIGNEE. • Find a truck. • Assign truck and operate it. (Each trip take 1 day could be changed according to distance). • Inform Consignee that cargo was delivered.

Table 7-1: Daily Tasks for each Actor in the Simulation Scenario.

7.6.3 Agents in the Prototype Design

To define the agents we apply a physical decomposition approach, which means that entities in the real world are represented by agents of the prototype design. Accordingly, we decided that each port community member that is involved in the port supply chain should be represented as an agent in the simulation environment.

We classify different agents in the port supply chain simulation into two broad categories; namely, Community Agents and Control Agents. Community Agents are involved in actual business processes of the port community supply chain, and control agents help in coordinating the simulation environment in an efficient manner with the use of messages. In the following subsections we briefly describe each of these agents.

7.6.3.1 Control Agents

Control agents are a group of agents employed to help in coordinating the simulation environment. In our model, a group of three agents; these agents are as follows:

- The ***SimAgent***: it is the agent responsible for starting and executing simulations. In particular, it sets a specific start date, initialises the agent platform, and then starts the simulator. It then starts the *SyncAgent* and the *TimeAgent*. It has only three behaviours: a ListenBehaviour, which listens to completion messages from the ManagementAgent; a RestartBehaviour, that is executed to restart the simulation for a new simulation day; and a DelayRestartBehaviour which automatically executes after 24 (simulated) hours, or when an event occurs.

- The ***SyncAgent***: It takes care of the needed communication with the Repast platform.
- The ***TimeAgent***: It has functionalities and behaviours to handle date and time within the simulation environment. Furthermore, it controls the speed of simulation to allow for running the simulation at a faster or slower - pace. The synchronisation with the simulator is handled by the SyncTimeBehaviour. Other agents contact the TimeAgent to find out the current time in the simulated world, this is handled by the ListenBehaviour of the agent.

7.6.3.2 Community Agents

Community agents are a group of agents corresponding to the port community actors that are involved in this business process. In our model, a group of six agents constitute an agent organization that models a subset of the port community supply chain. These agents are as follows:

- The *PortAuthority* agent: it plays the role of the port authority.
- The *ShippingAgent* agent: it plays the role of the shipping agency company.
- The *InlandTransporter* agent: it plays the role of the truck company.
- The *CustomHouse* agent: it plays the role of the customs house.
- The *ContainerTerminal* agent: it plays the role of the container terminal
- The *Consignee* agent: it plays the role of the consignee.

For each agent in the community agents group, responsibilities in the simplified business process were designed as agent behaviours. These agent behaviours are described in sub-section 7.7.3.3 below.

7.7 The Development Phase of the Simulation Environment Prototype

The following sub-sections describe how the prototype was implemented. It details the hardware platform, software configuration, structures used and how JADE and Repast was integrated together to ensure implementation of the simulation functionalities.

7.7.1 The Development Environment

As discussed in sub-section 7.5.1, we used “JADE 3.7” (Java Agent Development Environment) in combination with the “Repast J” implementation of “Repast 3” for building agents of our prototype. The development process was carried out using Eclipse[8] IDE (integrated Development Environment) which is an open-source multi-platform Java development environment. We used a Toshiba Satellite Laptop with Intel Pentium 1.73 GHz processor and 2 GB of RAM, Windows XP 2002 SP3, and JRE 1.6 to develop the prototype. In addition, we used Java Excel API[9] to provide agents with the capability to manipulate Excel spreadsheets. The Java Excel API is an open source java API enabling developers to read, write, and modify Excel spreadsheets dynamically.

[8] http://www.eclipse.org

[9] http://jexcelapi.sourceforge.net/

7.7.2 JADE-Repast Integration

In this work, as discussed in sub-section 7.5.1, we employed JADE-Repast combination for building the agent-based simulation model. Two different approaches for combining JADE and Repast has been described in sub-section 7.5.3. In this regard, we followed the approach described by Yoo and Glardon (Yoo and Glardon, 2009) with some modifications.

In this approach, they employed a behaviour (*ObservableBehaviour)* through which changes in the internal agent state can be observed from the Repast monitoring environment. They implemented this behaviour as an object that is injected into the agent at runtime using O2A (Object-to-Agent) functionality. This functionality is included in the JADE platform to allow direct communication with agents from external code. In our implementation, we developed the *ObservableBehaviour* as an internal agent behaviour using the FSMBehaviour type behaviour provided by JADE. The FSMBehaviour is a composite behaviour that executes its children according to a Finite State Machine defined by the user.

Another modification is in the approach that we follow to use the capabilities provided by Repast in order to introduce the notion of time into our JADE model. Repast already possesses an internal clock, where each Tick represents a simulation step. However, in JADE there is no given internal clock. We created a JADE agent (*SyncAgent*) that acts in a Repast environment and it is accordingly synchronized with Repast's internal clock. At each tick, the *SyncAgent* sends a message to all the agents to inform them about the start of a new simulation day. In addition, the *SyncAgent* is acquainted with the agents in the simulation environment and it keeps track of the actions performed by each agent. It receives a confirmation message from each agent to inform that the tasks of the current simulation day have been finished. In the *SyncAgent*, we implemented the Schedule class provided by Repast in a way that the clock is only incremented after it receives a confirmation message from all the agents in the environment.

7.7.3 Building Agents of the Prototype

The agents of the simulation environment prototype have been developed according to the specifications of the design phase as described in section 7.6.

7.7.3.1 Building the Generic Agent (PCAgent)

In sub-section 7.4.3, the structure of a generic agent for port community simulation is presented along with its capabilities. In this sub-section, implementation of such agent is introduced.

The Port Community Agent (PCAgent) has been implemented as a Jade-Agent with capabilities (behaviours) that we have defined as essential for any agent in the simulation environment.

7.7.3.2 Building Control Agents

The agent behaviours that have been developed for the control agents and their corresponding function is shown in Table 7-2.

Agent	Behaviour	Functionality
SyncAgent	ListenBehaviour	Cyclic behaviour that listens to messages from other agents and the Repast platform.
	InformBehaviour	OneShotBehaviour that sends the message received from the Repast platform, to the right agent in the simulation environment.
TimeAgent	ListenBehaviour	Cyclic behaviour that listens to messages from other agents.
	SyncTimeBehaviour	Cyclic behaviour that syncs the time and simulation rate with the Repast platform.
LogAgent	LogBehaviour	Ticker behaviour that performs different log activities (on screen, Excel spreadsheets, *etc*)
	ListenBehaviour	Cyclic behaviour that listens to messages from other agents.
SimAgent	ListenBehaviour	Cyclic behaviour that listens to messages from other agents.
	RestartBehaviour	OneShotBehaviour that restarts the simulator on the next simulation day; it also creates the proper simulator log file.
	DelayRestartBehaviour	Waker behaviour that triggers the RestartBehaviour after a specified amount of time.

Table 7-2: Agents of the control group and their corresponding functionalities.

7.7.3.3 Building Community Agents

A group of six agents have been developed as an agent organization that models a subset of the port community supply chain. Namely, these agents are the *PortAuthority* agent, the *ShippingAgent* agent*,* the *InlandTransporter* agent*,* the *CustomHouse* agent*,* the *ContainerTerminal* agent*,* and the *Consignee* agent. In this sub-section, we briefly describe the role of each agent and the corresponding behaviours to conduct it.

The ***PortAuthority*** Agent

The ***PortAuthority*** agent is responsible for scheduling the ship arrivals and departures. It communicates with the *ContainerTerminal* agent to organize ship arrivals. On the other hand, it communicates with *ShippingAgent* agent to receive information about the expected arrivals of ships. The agent behaviours that has been developed for the *PortAuthority* and their corresponding function is shown in Table 7-3.

Behaviour	Functionality
InitialBehaviour	Initial behaviour that starts all behaviours of the *PortAuthority* Agent.
ListenBehaviour	Cyclic behaviour that listens to messages from other agents
RcvBerthingApp	AchieveREResponder behaviour that checks and accepts berthing application from shipping lines
BerthingBehaviour	CyclicBehaviour that schedule a ship for berthing in the port.
askContTerm	AchieveREInitiator behaviour that consults the container terminal agent about its operation status in order to schedule a container ship for berthing.
RequestShipOperation	AchieveREInitiator that informs the container terminal about the arrival of ship and triggers the terminal to start the ship operation.
AssignBerthBehaviour	OneShotBehaviour which inform the shipping line agent about the berth assigned for a specific ship.

Table 7-3: Agent Behaviours of the PortAuthority Agent

The ***ShippingAgent*** Agent

The *ShippingAgent* agent coordinates the arrangements for ship arrival, berthing, discharge/loading and departure of the vessel with regulatory bodies. Therefore, it communicates with the *PortAuthority* to arrange for ship arrivals and with the *ContainerTerminal* agent to arrange for ship operation (loading and unloading). In addition, it communicates with the *Consignee* agent to coordinate cargo (i.e. containers) arrival and departure. The agent behaviours that has been developed for the *ShippingAgent* and their corresponding function is shown in Table 7-4.

Behaviour	Functionality
InitialBehaviour	Initial behaviour that starts all behaviours of the *ShippingAgent* Agent.
ListenBehaviour	Cyclic behaviour that listens to messages from other agents
PrepareBerthApp	AchieveREInitiator behaviour that prepare the berthing application according to the schedules of the expected time of arrival of the incoming ships. Then, it sends it to the port authority
EndOfShipOperation	CyclicBehaviour that monitors the ship operation and informs the concerned consignee about container arrival.

Table 7-4: Agent Behaviours of the ShippingAgent Agent

The ***InlandTransporter*** Agent

In our specific business process for import, the *InlandTransporter* agent performs the transport of containers from terminal to consignee premises. It allocates a truck of its fleet to transport a specific container. To carry out this task, it communicates with the Consignee agent to receive requests for a truck. The agent behaviours that has been developed for the *InlandTransporter* and their corresponding function is shown in Table 7-5.

Behaviour	Functionality
InitialBehaviour	Initial behaviour that starts all behaviours of the *InlandTransporter* Agent.
ListenBehaviour	Cyclic behaviour that listens to messages from other agents
RcvTruckRequest	AchieveREResponder behaviour that receives requests for trucks from consignee
AssignTruckBehaviour	A one shot Behaviour that assigns a specific truck to carry out a specific request.
OperateTruck	A one shot behaviour that set a truck object into operation mode.

Table 7-5: Agent Behaviours of the InlandTransporter Agent

The ***CustomHouse*** **Agent**

In our model, the *CustomHouse* agent performs only an administrative control. It simulates the process of issuing a custom certificate as a prerequisite for container clearance. It communicates with the consignee agent to perform this role. The agent behaviours that has been developed for the *CustomHouse* Agent and their corresponding function is shown in Table 7-6.

Behaviour	Functionality
InitialBehaviour	Initial behaviour that starts all behaviours of the CustomHouse Agent.
ListenBehaviour	Cyclic behaviour that listens to messages from other agents
RcvCustomDeclaration	AchieveREResponder behaviour which receive and check the customs declaration delivered y a consignee. Accordingly, it initiates the custom clearance process.
ProcessCustomDeclaration	OneShotBehaviour to start simulating the custom clearance process and issuing the custom certificate.

Table 7-6: Agent Behaviours of the CustomHouse Agent

The ***ContainerTerminal*** Agent

The *ContainerTerminal* agent manages the ship operations (loads/discharges) and the storage of containers in the yard. It communicates with both the *PortAuthority* and the *ShippingAgent.* The agent behaviours that has been developed for the *ContainerTerminal* Agent and their corresponding function is shown in Table 7-7.

Behaviour	Functionality
InitialBehaviour	Initial behaviour that starts all behaviours of the *ContainerTerminal* Agent.
ListenBehaviour	Cyclic behaviour that listens to messages from other agents
ShipCalls	AchieveREResponder behaviour informs the port authority about the status of ship operation.
RcvShipOperation	AchieveREResponder behaviour that receives a notification from the port authority about the arrival of a ship. It then triggers the ship operation process to start.
StartShipOperation	OneShotBehaviour that simulates the ship operation process.

Table 7-7: Agent Behaviours of the ContainerTerminal Agent

The ***Consignee*** Agent

The *Consignee* agent acts as the owner of the imported goods in a specific container(s). It is responsible for completion of formalities regarding cargo clearance procedures and to get the container out of the terminal. Therefore, it communicates with the *CustomHouse* agent (for formalities) and with the *InlandTransporter* agent (for haulage). The agent behaviours that has been developed for the *Consignee* Agent and their corresponding function is shown in Table 7-8.

Behaviour	Functionality
InitialBehaviour	Initial behaviour that starts all behaviours of the *Consignee* Agent.
ListenBehaviour	CyclicBehaviour that listens to messages from other agents
CargoArrivalNotification	CyclicBehaviour that receives a notification from the shipping agent about the container arrival. Accordingly, it triggers requesting a custom certificate and a truck.
RequestCustomCertificat	AchieveREInitiator behaviour that apply for a custom certificate from the customs house.
RcvCustomCertificate	CyclicBehaviour that receive a notification from the customs house; and then inform the inland transporter that the container is available.
RequestTruck	AchieveREInitiator behaviour that requests a truck from an inland transporter.

Table 7-8: Agent Behaviours of the Consignee Agent

7.8 The Assessment of the Truck Assignment Prototype

In this section, we introduce an assessment of the truck assignment prototype which is proposed in Chapter 6. The purpose of this evaluation is to examine the effectiveness of the prototype as a component of the ABMPCM (Agent-based Middleware for Port Community Management) described in Chapter 5. The prototype assessment has been carried out through simulation using the agent-based system for port community which is described in sections 7.6 and 7.7 above.

7.8.1 The Simulation Experiments

A set of simulation experiments were conducted to illustrate the effectiveness of the agent-based truck assignment system prototype that is implemented as described in Chapter 6. The prototype of the port community simulation environment, introduced in sections 7.6 and 7.7 above, is employed to carry out the simulation experiments. We used a Toshiba Satellite Laptop with Intel Pentium 1.73 GHz processor and 2 GB of RAM, Windows XP 2002 SP3, and JRE 1.6 to perform the simulation experiments.

The context for the simulation consists of trucking operations to serve demands for inland transport of containers. In the experiments, we simulate five inland transporters denoted as A, B, C, D, and E; and each has its own fleet of 5, 7, 8, 10, 12 trucks respectively.

The simulation experiments have been carried out to compare the trucking operation in two different scenarios; namely, **scenario I** (indicate the current PCSs without coordination) and **scenario II** (with our proposed prototype that supports coordination).

In scenario I (without coordination facility), the inland transporters serve the incoming orders independently according to their contractual agreements with specific customers. This means that there is no coordination with other inland transporters. Merely the prototype of the port community simulation environment is employed to perform simulation experiments for this scenario.

In scenario II (with coordination facility), we assume that the five trucking companies agree to use our proposed middleware as a trusted coordination facility. Therefore, we employed the proposed truck assignment prototype in this simulation scenario for scheduling truck jobs as a common fleet. This is equivalent to "pooling" where resources (i.e. the trucks) are grouped together for the purposes of maximizing advantage and/or minimizing risk to the users.

As described in Chapter 6 (Section 6.5), two types of agents are involved in the agent-based truck assignment prototype; namely, the *TruckAgent* (for each truck in the system) and the *OrderAgent* (for each order in the system). In the context of our simulation experiments, *TruckAgents* are competing and cooperating. They are cooperating in the sense that they agree to share resources (i.e. the trucks) and to share information (i.e. the orders). On the other hand, they are competing in the sense that each *TruckAgent* attempts to get an order through a specific bidding procedure for truck assignment (as described in sub-sections 6.5.3, 6.5.4, and 6.5.5).

The simulation setting for scenario II is illustrated in Figure 7-9. In this setting, the agents of the proposed prototype prepare a truck plan (in our case a plan containing only one assignment), then it is sent to the simulator where its execution is carried out. Upon completion of the assigned task, a feedback from the simulator is sent to the agent system again. The messages exchanged between the MAS prototype and the simulator are logged in files for further performance analysis.

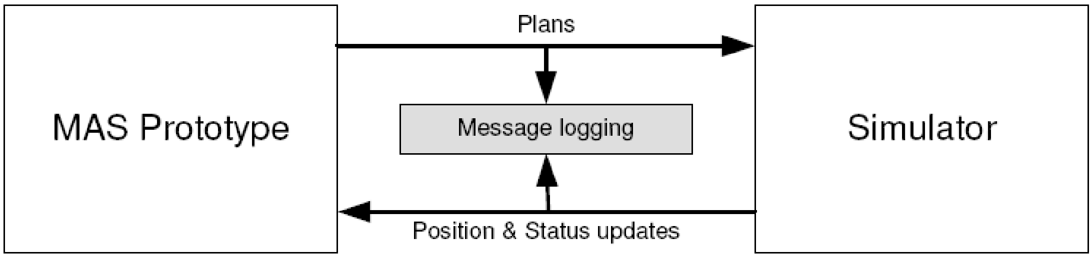

Figure 7-9: The Simulation Setting for Scenario II.

The initial parameters for the simulation experiments have been obtained through several discussions with practitioners in the port community of the Port of Alexandria (EGYPT), and they were continuously tuned during model validation and verification.

In the inland transport of containers, a very basic form of uncertainty is the job arrivals over time. Consequently, we employ a random number generator in this simulation experiment to generate truck orders. Unfortunately, we did not have access to operational data from any real inland transporter which would have allow us to analyse the data in order to identify how the job arrivals fit to a specific type of standard mathematical distribution such as the normal or exponential distributions.

Alternatively, we rely on the information gathered during the interviews that we have carried out with inland transporters. The demand for trucks changes according to the seasonal variation of the container shipping. This means that there are differentiable peak and off-peak seasons. Accordingly, we decided that it is safe to assume that demands for inland transport of containers arise according to normally distributed inter-arrival time distribution. Normal distribution fits the nature of the demand for trucks rather than the exponential distribution since the interviewees did not confirm that the demand shows a constant rate of decay.

According to the information that we have gathered during field visits and interviews, the average inter-arrival time of orders varies from 20 minutes to 45 minutes according to the size of the company. We have tried several random samples with different mean and standard deviation. We found that the random sample with mean = 30 minutes and standard deviation = 8 produces more reasonable output.

To formalize the structure of the simulation experiments, we made some formal assumptions as follows:

- All information about each order is available at the time of its announcement. This information include: the pick-up location, the customer location, return drop-off location, and the required time windows for arrival at each of these three locations.
- The Loading and unloading of the container at the terminals and customer premises takes a constant time period of 1 hour.

- Time window violations are allowed. This assumption has been made based on our field observations. Truck companies are not strict regarding time windows; therefore, time window violations are not penalized in our simulation experiments.
- The time windows at the customer locations are always 2 hours while it span a full working-day at the terminal. The start of the time window for each order occurs throughout the working hours of the day, and roughly follows a uniform distribution.
- All trucks are equivalent. The truck speed is constant.
- Ten working hours are taken as the standard length of a working-day for trucks.

The simulation experiments involve 10 replicas. Each replica contains 120 randomly generated orders every simulation day. For each simulation run, the principal measures of performance examined in the analysis include the following:

- The **number of empty trips**; indicates the number of truck trips with an empty container.
- The **number of late arrivals**; indicates the number of orders delivered or picked-up beyond, or too close to, the end of the preferred time window.
- The **number of on time arrivals**; indicates the number of orders delivered or picked-up on time (i.e. within the preferred time window).

These three measures can be considered as KPIs not only for the inland transport community but also for the port supply chain. They are also related to port performance as explained below.

As discussed in Chapter 2, the business process in the port supply chain is an inter-organisational business process that spans organisational boundaries. The port supply chain of services encompasses various types of logistics service providers (e.g. storage, customs clearance, inland transport, and distribution), customers (e.g. shippers, consignees, and shipping companies), and relevant government regulatory authorities (e.g. port authorities, customs, and quarantine). Since the goal of a contemporary port is to satisfy customers with greater effectiveness and efficiency than the competitors; therefore, the measurement of port supply chain performance needs to incorporate these performance aspects. It should not be focused only on individual functional areas, but rather on the different parties involved in the port supply chain.

In that sense, the above mentioned three measures can be considered as a lower level KPIs that affect the measurement of the overall performance of the port supply chain of services.

Furthermore, as indicated in Chapter 2, integration of contemporary ports (terminals) in global supply chains is essential and such integration should lead to performance improvements and to competitive advantages for the port (terminal) that fulfils its modern role in the logistics era. The recognition that ports are increasingly integrated in supply chains is illustrated in several studies (Paixão and Marlow, 2003; Marlow and Paixão, 2003; Bichou and Gray, 2004). Bichou and Gray (2004) indicate that adopting a logistics approach to the measurement of port performance is beneficial to port efficiency because it directs port strategy towards relevant value-added logistics activities. In addition, they indicated that port and terminal integration in supply chain may involve the extent to which the port plans and organizes activities, processes and procedures beyond its boundaries and monitors performance in such activities.

In that sense, the measures concerned in our simulation experiment (as mentioned above) are beneficial for ports to monitor the performance of the inland transport activity of the supply chain. In particular, the relationship of these measures to port performance can be explained as follows.

The empty trips measure: As discussed in Chapter 6 (Section 6.3), every export-import cycle is associated with two truck trips with an empty container; these are considered as unproductive trips. The increase in the number of empty truck trips (unproductive trips) to and from the container terminal causes significant inefficiencies and it leads to the congestion at the gates and around the terminal as well as the roads that lead to the terminal area. The port authorities (or terminal operators) are definitely concerned with the problems of congestion and the related problems such as inefficient operations, delays at the gates, road congestion, and increased pollution. Ports must demonstrate a high level of environmental performance not only in view of community support, but also in view of attracting trading partners and potential investors.

Deliver / pick-up time: This measure is concerned with the performance of the land interface of the port. Expected arrivals of trucks at the port (terminal) are matched with capacity planning data of equipments (e.g. straddle carriers) and release orders of containers that need to be picked up. Therefore, the on time arrivals of trucks at the port (terminal) would reduce the re-planning process that the terminal may undertake in the case of late arrivals. Accordingly, this would enhance the port productivity (performance) providing that the re-planning process would cost the port time and money.

Moreover, pick-up and delivery of goods often accounts for a large part of the port congestion and inefficiency. The average time spent by a trailer waiting for its load to be located, handled, and to get its clearance should not exceed a specific short period of time. The improvement of the on time arrivals at the port (terminal) would positively contribute to that criterion.

7.8.2 The Simulation results

The simulation runs have been carried out at a speed five times faster than normal, which means that a working day of ten hours can be simulated in two hours. In total, we simulated 10 replicas for each of the two scenarios with 30 different days. This means that we have simulated 10*2*30 different days, which results in a total of 600 days. Since we could perform 12 simulation days per 24 hours, simulation runs spanned a total of 50 days.

The above mentioned performance measures were obtained from the 10 replica runs. The simulation results of scenario I (no coordination) are summarized in Table 7-9. The table shows the average values of the 10 replicates. The results of the scenario I simulation runs is illustrated in Figure 7-10.

Inland Transporter	Number of Trucks	Empty Trips	Late Arrivals	On time Arrivals	Number of Daily Orders
A	5	13.93	9.32	4.83	14
B	7	19.86	11.24	9.13	20
C	8	22.86	15.23	8.3	23
D	10	29.23	6.3	23.17	29
E	12	34.2	11.2	23.27	34

Table 7-9: The Daily Average Values of Scenario I Simulation Runs.

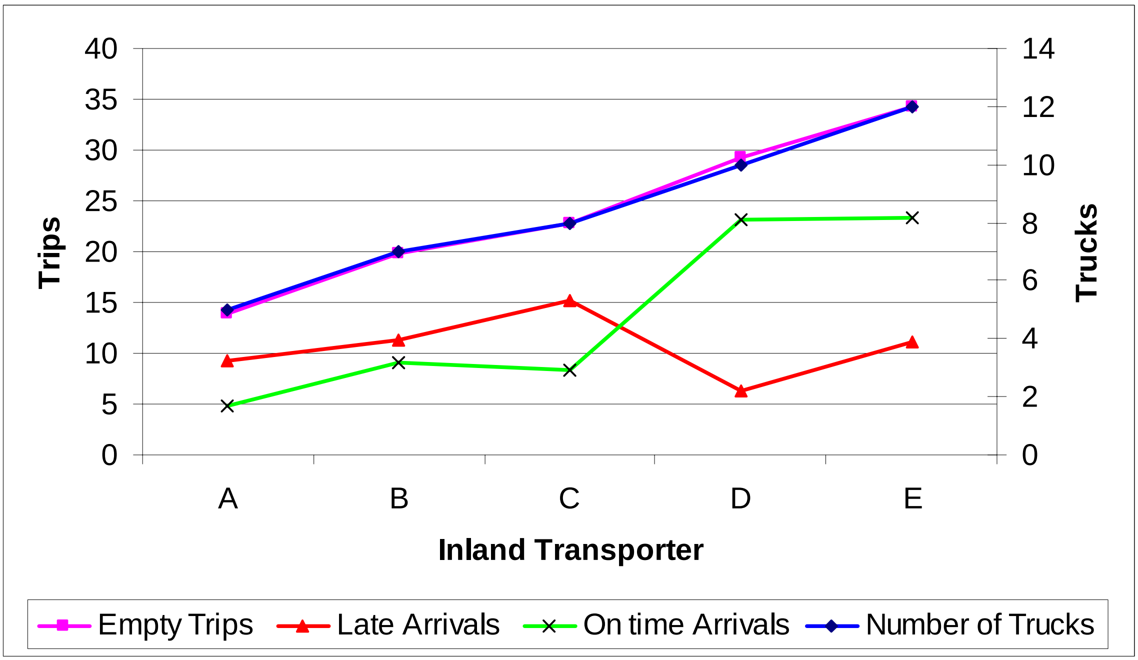

Figure 7-10: The results of the scenario I simulation runs

Regarding scenario II (with coordination), the same performance measures were obtained from the 10 replicate runs. The simulation results of scenario II are summarized in Table 7-10. The table shows the average values of the 10 replicates. The results of the scenario II simulation runs are illustrated in

Figure 7-11.

Inland Transporter	Number of Trucks	Empty Trips	Late Arrivals	On time Arrivals	Number of Daily Orders
A	5	9.45	7.66	6.34	14
B	7	16.87	10.2	9.8	20
C	8	18.06	11.81	11.19	23
D	10	28.95	5.04	23.96	29
E	12	29.72	10.86	23.14	34

Table 7-10: The Daily Average Values of Scenario II Simulation Runs

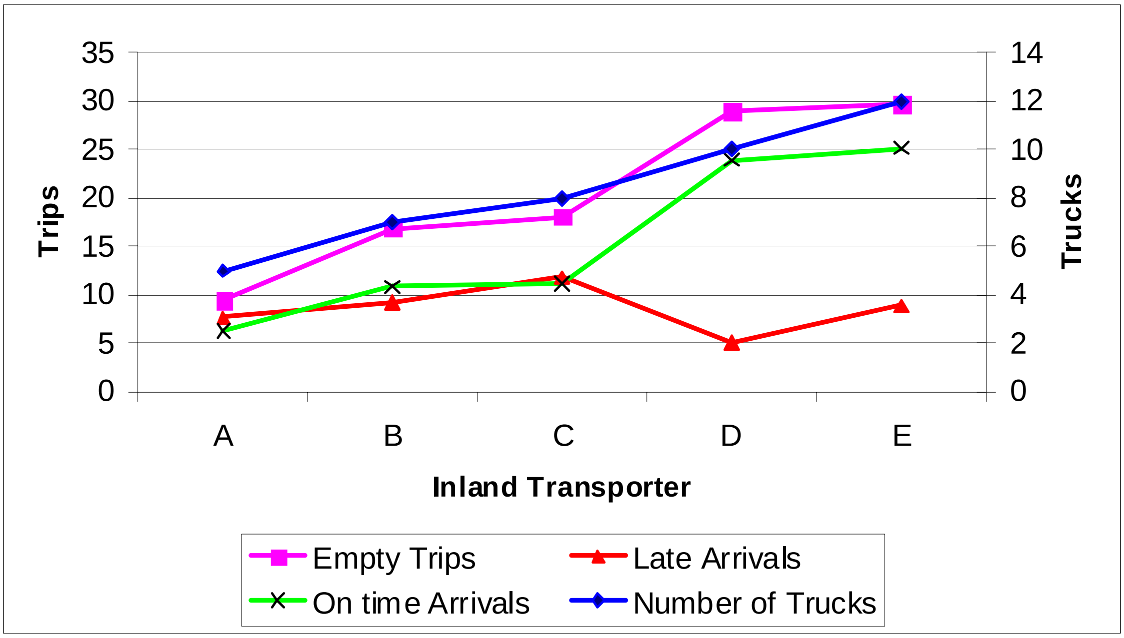

Figure 7-11: The results of the scenario II simulation runs

The analysis of the simulation results of the two scenarios revealed that the number of orders that each inland transporter carries out is on average the same in both scenarios. This observation might be interpreted as implying that using the proposed truck assignment prototype does not affect the market share of the inland transports. The reason why each of the inland transports carries the same number of orders is because of the daily maximum number of orders that each inland transporter can serve. This daily maximum number of orders was reached in scenario I; therefore, inland transports can not serve any more orders in scenario II. However, the proposed prototype introduces some improvements for the whole community of inland transporters.

The first improvement that the proposed prototype introduces is the reduction of the number of empty trips. In scenario II, the average number of empty trips is 20.61 trips per day, while it is 24.02 trips per day in scenario I. This implies that introducing coordination might reduce the number of empty trips by value of about 14.2 %. The comparison of the simulation results of scenario I and scenario II for the number of empty trips is illustrated in Figure 7-12.

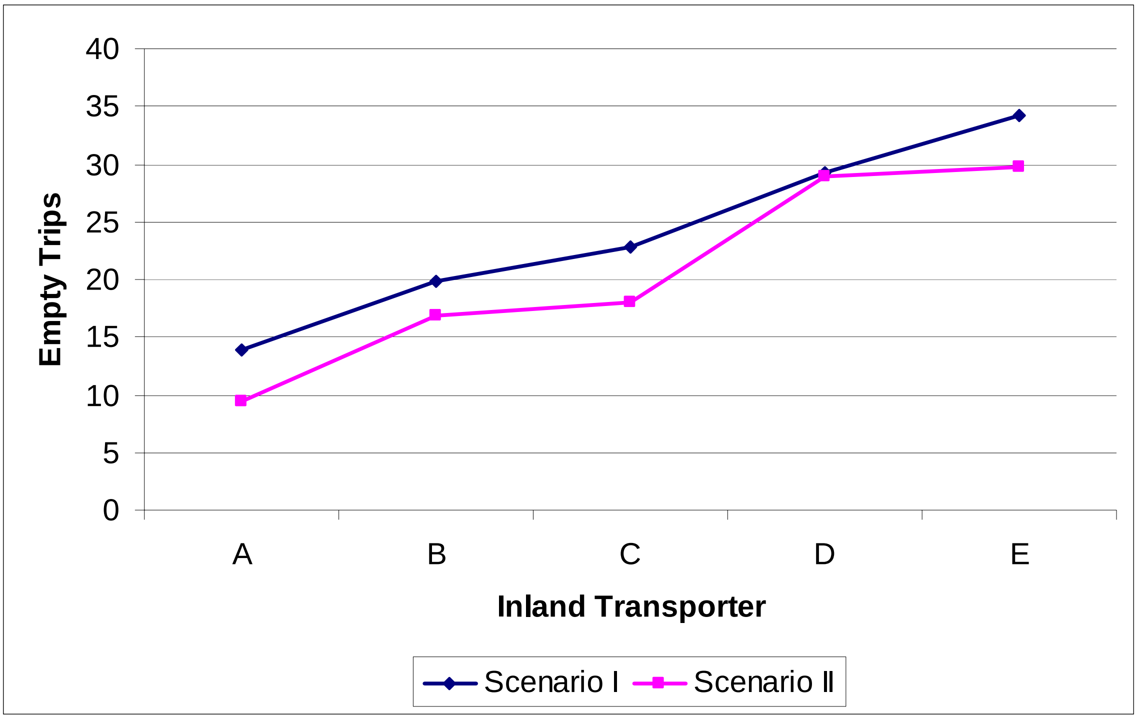

Figure 7-12: The Comparison of the Simulation Results of Scenario I and Scenario II for the Number of Empty Trips

In order to evaluate the results, a t-test was carried out to assess whether there is a significant change in the number of empty trips due to the employ of the proposed agent-based prototype. The output of the statistical analysis is illustrated in Table 7-11 and Table 7-12.

Paired Samples Statistics				
	Mean	N	Std. Deviation	Std. Error Mean
No. of Empty Trips in Scenario I	24.016	5	7.93	3.546
No. of Empty Trips in Scenario II	20.61	5	8.63	3.859

Table 7-11: Paired Statistics for Number of Empty Trips.

Paired Samples Test: No. of Empty Trips in Scenario I - No. of Empty Trips in Scenario II							
Paired Differences							
			95% confidence interval of the Differences				
Mean	Std. Deviation	Std. Error Mean	Lower	Upper	t	df	Sig. (2-tailed)
3.41	1.884	0.843	1.067	5.745	4.043	4	0.0156

Table 7-12: Output of t-test for Number of Empty Trips.

The statistical analysis revealed that a significant decrease in the number of empty trips occurred (t = 4.043, p = 0.0156) in simulation of scenario II (i.e. with the agent-based prototype) compared to scenario I (i.e. without the agent-based prototype). This can be considered as evidence that the agent-based middleware enhances the performance of community of inland transporters.

A second point of improvement that the prototype introduces is the reduction of the number of late arrival trips. In scenario II, the average number of late arrivals is 9.11 trips per day, while it is 10.66 trips per day in scenario I. This implies that the number of late arrivals is improved by 14.54% on the average. The comparison of the simulation results of scenario I and scenario II for the number of late arrival trips is illustrated in Figure 7-13.

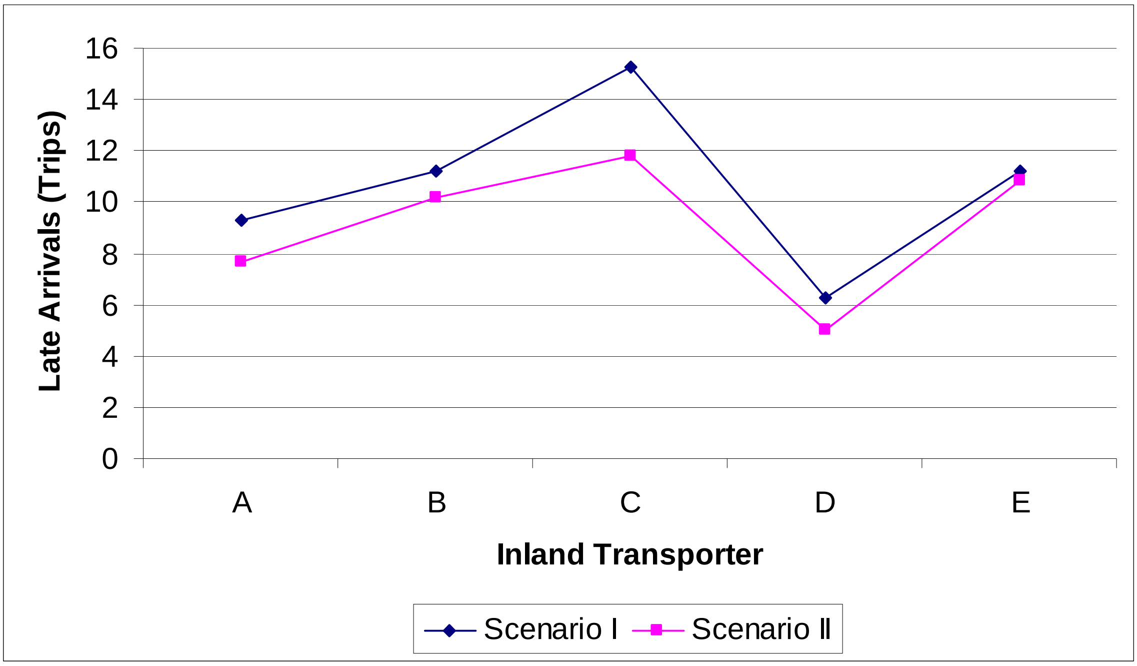

Figure 7-13: The comparison of the simulation results of scenario I and scenario II for the number of late arrival

In order to evaluate the results, a t-test was carried out to assess whether there is a significant change in the number of late arrivals due to the employ of the proposed agent-based prototype. The output of the statistical analysis is illustrated in Table 7-13 and Table 7-14.

Paired Samples Statistics				
	Mean	N	Std. Deviation	Std. Error Mean
No. of Late Arrivals in Scenario I	10.658	5	3.25	1.453
No. of Late Arrivals in Scenario II	9.114	5	2.75	1.229

Table 7-13: Paired Statistics for Number of Late Arrivals.

Paired Samples Test: No. of Late Arrivals in Scenario I - No. of Late Arrivals in Scenario II							
Paired Differences							
Mean	Std. Deviation	Std. Error Mean	95% confidence interval of the Differences				Sig.
			Lower	Upper	t	df	(2-tailed)
1.54	1.15	0.514	0.113	2.975	2.99	4	0.04

Table 7-14: Output of t-test for Number of Late arrivals.

The statistical analysis revealed that a significant decrease in the number of late arrivals occurred (t = 2.99, p = 0.04) in simulation of scenario II (i.e. with the agent-based prototype) compared to scenario I (i.e. without the agent-based prototype).

Accordingly, this statistically significant reduction in the number of late arrivals is accompanied by a third point of improvement. This can be inferred from the slight improvement in the number of on time arrivals of 8.34 % on the average. The comparison of the simulation results of scenario I and scenario II for the number of on time arrivals is illustrated in Figure 7-14.

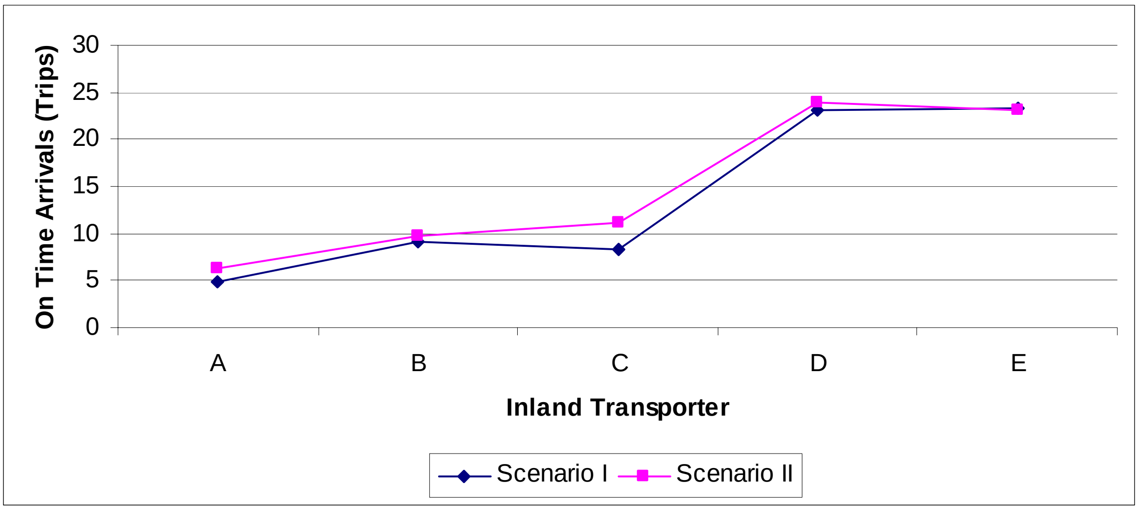

Figure 7-14: The comparison of the simulation results of scenario I and scenario II for the number of on time arrivals

In order to evaluate the results, a t-test was carried out to assess whether there is a significant change in the number of on time arrivals due to the employ of the proposed agent-based prototype. The output of the statistical analysis is illustrated in Table 7-15 and Table 7-16.

Paired Samples Statistics				
	Mean	N	Std. Deviation	Std. Error Mean
No. of On Time Arrivals in Scenario I	13.74	5	8.80	3.935
No. of On Time Arrivals in Scenario II	14.89	5	8.11	3.627

Table 7-15: Paired Statistics for Number of On Time Arrivals.

Paired Samples Test: No. of On Time Arrivals in Scenario I - No. of On Time Arrivals in Scenario II							
Paired Differences							
Mean	Std. Deviation	Std. Error Mean	95% confidence interval of the Differences		t	df	Sig. (2-tailed)
			Lower	Upper			
-1.15	1.14	0.5098	-2.555	0.2634	-2.26	4	0.087

Table 7-16: Output of t-test for Number of On Time Arrivals.

The statistical analysis revealed that the decrease of the number of on time arrivals in simulation of scenario II (i.e. with the agent-based prototype) compared to scenario I (i.e. without the agent-based prototype) is statistically insignificant (t = -2.26, p = 0.087). This can be interpreted by the weight that we used for the score elements in this experiment. The weight of the “Empty Distance” score element was higher than the weight of the “Delivery Time Window” score element. Accordingly, this affects the grand score that the *TruckAgent* calculates to select an order.

The above discussion of the simulation results shows that the proposed prototype introduces some statistically significant improvements for the whole community of inland transporters. However, when we analyse the simulation results from the user’s perspective (i.e. the inland transporter perspective), we find that the transporters do not equally benefit from using the prototype.

For example, the inland transporter “E” shows a 3.3% decrease in the number of late arrivals (Figure 7-13); which is not a considerable improvement compared to other transporters. This can be attributed to the comparatively good performance of the inland transporter “E” in scenario I (without coordination); it shows a comparatively low ratio of late arrivals (32.94%) out of the average daily orders.

Another example is the on time arrivals measure (Figure 7-14) where the inland transporters “D” and “E” shows relatively slight improvement (3.30% and 0.56% respectively). This can be attributed to the performance of the inland transporters “D” and “E” in scenario I (without coordination). They show comparatively high ratios of on time arrivals (79.9% and 68.44% of the average daily orders respectively) with regard to the other inland transporters.

These results imply that the coordination functionality provided by the proposed prototype might be bounded by a certain level of improvement. The implications of such results would deter some users to adopt the system in real application. Therefore, further extensive experiments with different business scenarios are necessary to gain more insights on the performance of the proposed prototype.

The improvements that the simulation experiments demonstrate for the whole port community can be attributed to the distribution of the orders available for the community. In scenario I, each trucker serves only orders from customers with which they have a contractual agreement; and the truck location is not considered. In scenario II, the truck delegates its corresponding TruckAgent to select the best order that fits its status; especially the truck location. This means that the orders that are assigned for a trucking company in scenario I are not the same orders that are assigned for the same company in scenario II; though they are of the same number.

The simulation results revealed that the proposed system facilitates coordination and collaboration in such a way that it improves the performance of the inland transport process in three directions as described above. However, it is worth mentioning that there are other alternatives to improve the truck assignment process; such as, centralized control with simple heuristic rules (e.g. distance-based allocation). Normally, practices in operations research (OR) outperform agent-based approaches in settings where all information is known in advance (static settings). However, agent-based approaches are expected to outperform these traditional methods in situations with a lot of uncertainty, (Jennings and Bussmann, 2003). In the experiments presented in this Chapter, a very basic form of uncertainty is that of job arrivals over time; we therefore adopt the agent-based approach.

A comparative study has been carried out to examine the performance of an agent-based solution and an on-line optimization approach with respect to handling uncertainty in the context of a drayage company at the Port of Rotterdam (Mahr *et al.*, 2008). They compare the performance of an agent-based solution approach with an on-line optimization-based approach across four scenarios of varying job arrival uncertainty. They conclude that when less than 50% of all jobs are known at the start of the day then an agent-based approach performs competitively with an on-line optimization approach. These results can help transportation companies decide when to adopt an agent-based approach, and when to use an on-line optimization tool, depending on the level of uncertainty that job arrivals exhibit in their daily business.

The results of the simulation experiments as introduced in this Chapter imply that the proposed middleware successfully serves as the platform which enables the implementation of coordination and collaboration mechanisms. Although there may be other ways to achieve this, the middleware presented in this study has its unique merits since it adopts the agent-based approach. It brings a number of advantages such as: (i) the possibility for distributed computation; (ii) a decentralized multi-agent architecture can cope with multiple dynamic events; (iii) commercial companies may be reluctant to provide proprietary data needed for global optimization while agents can use local information; and (iv) inter-company cooperation can be more easily facilitated by agents. In addition, the transportation domain is inherently decentralized (trucks, customers, companies etc.); therefore, multi-agent models suit it particularly well (Fischer *et al.*, 1995).

7.9 Summary

In this chapter, we introduce an agent-based simulation framework as an effective instrument for practitioners in the port community to examine potential port supply chain improvements before its real implementation. The framework presents a comprehensive simulation environment based on five aspects: Data sets (Simulation scenarios), workflow management, port community supply chain model, simulation control, and user interface.

A prototype of the simulation framework has been implemented using JADE in combination with Repast. In the implementation, we combined the capabilities of the two disparate agent development tools to provide necessary functionalities of our proposed simulation framework.

The simulation prototype was employed to examine the effectiveness of the truck assignment prototype that is proposed in Chapter 6. A series of simulation experiments were conducted and involve 10 replicas. Each replica contains 120 randomly generated orders every simulation day.

A simplified business process for cargo import was selected for simulation experiments. The simulation experiments were conducted for two scenarios, one scenario employs the truck assignment prototype while the other scenario does not. The results of the simulation experiments revealed that the use of the agent-based truck assignment system significantly reduces the number of empty trips and the number of late arrivals.

Chapter 8

Conclusion and Future Work

8 Conclusion and Future Work

This chapter concludes the thesis by summarising the achievements of the research programme. The chapter proceeds to discuss the limitations of the research, and considers areas for further refinement.

8.1 Achievements of the Research

The research presented in this thesis aimed to employ agent-based approach for business process management in port community. It also aimed to design and evaluate an agent-based solution to improve the performance of business processes in port community supply chain.

The research has met all of the objectives originally outlined in chapter 1 (Section 1.2). The following summarises the achievements of the research.

1. A critical survey of the current situation from two different perspectives, business view and technical view. This corresponds to research objectives no. 1 and no.2.

   This survey has been carried out in order to investigate the appropriateness of agent technology to business needs of the port community environment. This survey is a fundamental requirement of the overall aim of the research, and therefore, it is a fundamental contribution in its own right.

On the business level, we investigate the maritime transport industry and in particular the changing role of seaports. Accordingly, a number of information and coordination requirements to deal with port community management have been identified. On the other hand, three technologies have been introduced as appropriate combination for providing solutions for port community management. These technologies are the workflow management, agent technology, and Web services.

2. A broad analysis of the Port Community Systems (PCSs) was carried out unlike any previous investigation of this topic. This corresponds to research objective no.3.

   This analysis aimed to assess the supply chain orientation of port community systems. The assessment focuses on the business process management capabilities of the existing PCSs.

   The analysis includes PCSs that are currently implemented in a number of the top ports of the world. The analysis revealed that most of the currently implemented PCSs lack efficient functionalities to facilitate the management of port supply chain.

   These kinds of analyses we have carried out in this research programme have not been carried out elsewhere in the literature. It examine the PCSs from a novel perspective, which is their role to serve the whole port community supply chains. Therefore, we consider this analysis a fundamental contribution to knowledge since it provides further insights into we know ever.

In addition, this assessment is fundamentally a prerequisite of the proposal of “new” solution. Therefore, we perceive it a fundamental contribution in its own right. Furthermore, this kind of analysis revealed other areas that need further research; such as, the factors that affects new IT solutions adoption in the context of port community supply chains.

3. We define the framework of the Agent-based Middleware for Port Community Management (ABMPCM) as solution for business process performance improvement. This corresponds to research objectives no. 4.

   The ABMPCM enable automation of the management of the distributed collaborative business processes in maritime port communities. The ABMPCM introduces business process management facilities that current port community systems need. It comprises five components; the Request Handling Facility (RHF), the Collaborative Planning Facility (CPF), the Process Management Facility (PMF), the Monitoring and Visibility Facility (MVF), and the Disruption Management Facility (DMF). The framework applies principles from autonomic computing, workflow management, and agent technology paradigms to enable outsourcing of the port community management tasks to the different facilities of the ABMPCM framework.

   The contribution of proposing the ABMPCM is that it fills the gap in functionalities due to the shortcomings of the exiting port community systems, as revealed from the analysis stated in contribution no. 2 above.

4. We define the Collaborative Planning Facility (CPF) framework, which is a core component of the proposed ABMPCM. This corresponds to research objectives no. 5.

   The CPF is provided to facilitate automated coordination of business process activities between members of the port community. We use agent technology for achieving coordination where agents perform the coordination process in the CPF locally and impartially. The CPF offers a powerful feature of allowing for feedback from the port community member during an ongoing coordination. We validate the effectiveness of the framework through a practical application of an experimental prototype for real-time agent-based truck assignment system that coordinates the process of inland container transport.

   The contribution of the prototype implementation is that it demonstrates the practical effectiveness of a core component of our proposed ABMPCM.

5. We define a multi-agent framework for the simulation of port supply chain processes. This corresponds to research objectives no. 5.

   The evaluation of the proposed multi-agent system is one of the original aims of this research, as stated in Chapter 1. However, the design of the middleware as a multi-agent system and its inter-organisational nature hinder its experimentation in the field. Actors of the port community were reluctant to adopt a solution whose functioning they do not fully understand or trust. Therefore, we decided to develop a simulation environment in order to facilitate the evaluation process of our Multi-Agent system, and we started by defining this framework.

The framework is designed as a multi-agent system in which members of the port community supply chain are modelled as a group of agents. The framework combines agent technologies with business rules and workflow management. An experimental prototype of the simulation framework has been implemented. In this regard, we integrate two disparate agent frameworks, in order to realize some of the functionalities of the framework. These frameworks are JADE[1] (Java Agent DEvelopment Framework) (Bellifemine *et al.*, 2007; Bellifemine *et al.*, 2001; Bellifemine *et al.*, 1999) and Repast[2] (The REcursive Porous Agent Simulation Toolkit).

The contribution of this simulation framework is that we implement it by combining two disparate agent development toolkits (JADE and Repast). In addition, this simulation framework is useful to build a simulation environment to experiment different business scenarios in the port community supply chain. This would be useful for decision makers and practitioners in the domain of maritime transport.

6. We conduct simulation experiments using the port community simulation prototype to assess the effectiveness of the agent-based truck assignment system (i.e. CPF prototype). This corresponds to research objectives no. 5.

[1] http://jade.tilab.com
[2] http://repast.sourceforge.net/

The simulation results revealed that the proposed middleware successfully serves as the platform which enables the implementation of coordination and collaboration mechanisms. Accordingly, the performance of the inland transport process is improved in three directions; the number of empty truck trips is reduced, the number of in time arrivals is increased, and accordingly the number of late arrivals is decreased.

The contribution here is that the results of the simulation experiments prove that our agent-based approach introduces a collaboration facility which effectively enhances the performance of business processes in the port community. Although there may be other ways to achieve that, the middleware presented in this study has its unique merits since it adopts the agent-based approach. It brings a number of advantages such as: (i) the possibility for distributed computation; (ii) a decentralized multi-agent architecture can cope with multiple dynamic events; (iii) commercial companies may be reluctant to provide proprietary data needed for global optimization while agents can use local information; and (iv) inter-company cooperation can be more easily facilitated by agents. In addition, the transportation domain is inherently decentralized (trucks, customers, companies etc.); therefore, multi-agent models suit it particularly well (Fischer et al., 1995).

It is therefore, considered that the research has made a contribution to knowledge within the domain of agent technology application for port supply chain management.

8.2 Limitations of the Research

Although the objectives of the research programme have been met, a number of decisions had to be made which imposed limitations upon the work. These decisions were typically either practically based or due to time restrictions. The key limitations of the research are summarised below.

1. A number of practical limitations exist with the analysis study of the port community systems (PCSs). The limited information about technologies employed in PCSs and the lack of information about user satisfaction hindered a more thorough evaluation of the PCSs.

2. Insufficient time and resources were available to implement all the components of the ABMPCM as well as the agent-based simulation framework. Therefore, the development was only performed using a partial prototype.

3. The lack of a practical evaluation of ABMPCM with real participants. This would have enabled a comprehensive analysis of system performance in real time. Unfortunately, the lack of port community members (companies) that agree to get involved in the evaluation experiments, meant system evaluation could only be performed theoretically in a simulated environment.

Despite these limitations, the research programme has made valid contributions to knowledge and provided considerable proof of concept for the ideas proposed.

8.3 Suggestions for Future Work

There are five key areas where continuation of the research should be focused. This work was considered outside the scope of this research or was considered too complex to complete within the research timeframe. These suggestions are detailed below:

1. A study on adoption of the ABMPCM. Although the ABMPCM has promising benefits for the whole port community, a lot of companies were reluctant to adopt it in an experimental study. We perceive that investigating the factors affecting the user acceptance of port community systems is very important. We suggest that conducting such research following a qualitative research approach would be useful.

2. Define and implement the Request Handling Facility (RHF), the Process Management Facility (PMF), the Monitoring and Visibility Facility (MVF), and other components of our proposed ABMPCM.

3. Extension of the ABMPCM to make its facilities available as Web services. In such configuration, each facility of the ABMPCM would be invoked as a service independent of the other facilities. In addition, it would be possible to distribute the facilities where each facility would be outsourced and provided by a trusted service provider.

4. To develop an ontology for port community supply chain management. This ontology is required to guarantee a common understanding of the notions used during interactions between port community members. The ontology should

contain all useful concepts that are used in maritime transport and supply chain management. Ontology plays a major role in supporting the information exchange processes in maritime transport. In general, it provides a shared and common understanding of the domain of knowledge, communication, etc.

5. To implement a complete prototype of the agent-based simulation framework. This prototype should be validated and extensively evaluated for various business scenarios. The simulation model should be verified and validated with respect to the actual data collected over a certain period of time (e.g. a year) of operation of the real system. A successful prototype would be useful for practitioners in the field of supply chain management to understand the complex dynamics of the port supply chain.

References

9. References

1. Aksentijević, S., Edvard, T. and Bojan, H. (2009). 'Importance of Organizational Information Security in the Port Community Systems'. *Proceedings of the 32nd International Convention on information and communication technology, electronic and microelectronics,* ACTA Press.

2. Allan, R. J. (2010) *Survey of Agent Based Modelling and Simulation Tools*, Warrington: Science and Technology Facilities Council, Daresbury Laboratory, Daresbury Science and Innovation Campus, Technical Report DL-TR-2010-007

3. Alonso, G., Agrawal, D., Abbadi, E., Mohan, C., Gnthr, R. and Kamath, M. (1995) Exotica/FMQM: A persistent message-based architecture for distributed workflow-management *Proceedings of IFIP WG 8.1 Workgroup Conference on Information Systems Development for Decentralized Organizations (ISDO95), Trondheim, Norway.*

4. Alonso, G., Fiedler, U., Lazcano, A., Schuldt, H., Schuler, C. and Weiler, N. (1999). 'WISE: An Infrastructure for E-Commerce '. *Proceedings of the Informatik'99 Workshop "Enterprise-wide and Cross-enterprise Workflow Management: Concepts, Systems, Applications"*, Paderborn, Germany:

5. Applegate, L. M., Bartlett, N., Chang-Leow, D. and Siong, N. B. (2004) Harvard Business School Case 802-003. *PSA: The World's Port of Call.*, Harvard Business School (Revised from original July 2001 version.)

6. Awad, A. (2007). 'BPMN-Q: A Language to Query Business Processes'. *Proceedings of* EMISA *2007*, Nanjing, China:

7. Bagchi, P. K. and Paik, S. K. (2001) 'The Role of Public-Private Partnership in Port Information Systems Development'. *International Journal of Public Sector Management,* **14**(6), 482-499.

8. Bakos, J. Y. (1991) 'Information Links and Electronic Marketplaces: The Role of Interorganizational Information Systems in Vertical Markets'. *Journal of Management Information Systems,* **8**(2), 31-52.

9. Baltzersen, P., Hagaseth, M. and Kvamstad, B. (2009). 'EFFORTS: Service Oriented Architecture for Interoperable Port Systems'. *Proceedings of CRITICAL ISSUES IN THE PORT AND MARITIME SECTOR*, Antwerp:

10. Barrett, M. and Konsynski, B. (1982) 'Inter-Organizational Information Sharing Systems'. MISQ *Quarterly,* **6**(1), 3-105.

11. Bellifemine, F., Caire, G. and Greenwood, D. (2007) *Developing Multi-Agent Systems with JADE (Wiley Series in Agent Technology)*: Wiley.

12. Bellifemine, F., Poggi, A. and Rimassa, G. (2001) 'Developing multi-agent systems with a FIPA-compliant agent framework'. *Software: Practice and Experience,* **31**(2), 103-128.

13. Bellifemine, F., Rimassa, G. and Poggi, A. (1999). 'JADE: A FIPA Compliant Agent Framework'. *Proceedings of fourth international conference and exhibition on the Practical Applications of Intelligent Agents and Multi-Agent systems (PAAM 99)*, London:

14. Beresford, A. K. C., Gardner, B. M., Pettit, S. J., Naniopoulos, A. and Wooldridge, C. F. (2004) 'The UNCTAD and WORKPORT Models of Port Development: Evolution or Revolution?' *Maritime Policy & Management,* **31**(2), 93-107.

15. Bichou, K. and Gray, R. (2004) 'A Logistics and Supply Chain Management Approach to Port Performance Measurement'. *Maritime policy and management,* **31**(1), 47-67.

16. Bichou, K. and Gray, R. (2005) 'A Critical Review of Conventional Terminology for Classifying Seaports'. *Transportation research, Part A, Policy and practice,* **39**(1), 75-92.

17. Blake, M. B. (2001). 'Agent-Based Workflow Configuration and Management of On-Line Services'. *Proceedings of the International Conference on Electronic Commerce Research (ICECR-4), 567–588.*,

18. Boer-Sorban, K. (2008) *Agent-Based Simulation of Financial Markets: A Modular, Continuous-time Approach* Thesis, (Ph. D.). Erasmus University

19. Boertien, N., Goedvolk, E.-J., Hammink, E., Hulsebosch, B., Kolff, T., Koolwaaij, J., Linde, M., Middelkoop, E., Oosterhout, M., Posthuma, P., Salden, W. and Zandbelt, H. (2002) *Blueprint for a Virtual Port, an Integrated View on Next Generation Internet in the Port Of Rotterdam*, Rotterdam: Virtuele Haven Consortium,

20. Bolcer, G. A. and Kaiser, G. (1999) 'SWAP: Leveraging the Web to Manage Workflow'. *IEEE Internet Computing,* **3**(1), 85-88.

21. Bontekoning, Y. M. (2006) *Hub Exchange Operations in Intermodal Hub-And-Spoke Operations: Comparison of the Performances of Four Types of Rail-Rail Exchange Facilities*. Thesis, (Ph. D.). Technische Universiteit Delft

22. Borghoff, U. M., Bottoni, P., Mussio, P. and Pareschi, R. (1997). 'Reflective Agents for Adaptive Workflows'. *Proceedings of Proceedings of the Second International Conference on the Practical Application of Intelligent Agents and Multi-Agent*,

23. Bürckert, H., Fund, P. and Vierke, G. (2000) 'Holonic Transport Scheduling with Teletruck '. *Applied Artificial Intelligence,* **14**(7), 697—725.

24. Carbone, V. and De Martino, M. (2003) 'The changing role of ports in supply-chain management: an empirical analysis'. *Maritime Policy & Management,* **30**(4), 305-320.

25. Casati, F., Shan, E., Dayal, U. and Shan, M.-C. (2003) 'Service-Oriented Computing: Business-Oriented Management of Web Services'. *Communications of the ACM,* **46**(10).

26. Cash, J. and Konsynski, B. (1985) 'IS redraws competitive boundaries'. *Harvard Business Review,* **63**(2), 134-142.

27. Chatterjee, A., Hummer, J. E., Clarke, D. B. and Ney, S. M. (1997) 'Landside Access to Seaports in Urban Areas: A Case Study'. *In Transportation Research Record: Journal of the Transportation Research Board, No. 1602, TRB, National Research Council, Washington, D.C.,* pp. 49–56.

28. Cheeseman, M. J., Swann, P., Hesketh, G. B. and Barnes, S. (2005) 'Adaptive manufacturing scheduling: a flexible and configurable agent-based prototype'. *Production Planning and Control* **16**(5), 479-487.

29. Chmiel, K., Gawinecki, M., Kaczmarek, P., Szymczak, M. and Paprzycki, M. (2005) 'Efficiency of JADE agent platform'. *Sci. Program.,* **13**(2), 159-172.

30. Choi, H. R., Keceli, Y. and Cho, M. J. (2008) 'An Analysis Present Condition of Information Systems in Turkish Ports and Direction of Improvement'. *International Journal of Digital Content Technology and its Applications* **2**(1).

31. Choi, H. R., Kim, H. S., Park, B. J., Park, N. K. and Lee, S. W. (2003) 'An ERP Approach For Container Terminal Operating Systems'. *Maritime Policy and Management,* **30**(3), 197-203.

32. Choi, K. and Jang, M. (2007) ' An xml-based process repository and process query language for integrated process management'. *Knowledge and Process management,* **14**(4), 303-316.

33. Christiaanse, E. and Rodon, J. (2005) 'A multilevel analysis of eHub adoption and consequences'. *Electronic Markets,* **15**(4), 355–364.

34. Ciborra, C. (2000) *From Control to Drift: The Dynamics of Corporate Information Infrastructures*: Oxford University Press.

35. Collier, N., Howe, T. and North, M. J. (2003). ‘Onward and upward: The transition to Repast 2.0. ‘Proceedings *of First Annual North American Association for Computational Social and Organizational Science Conference,* Pittsburgh, PA USA:

36. Cross, N. (2007) 'Forty Years of Design Research'. *Design Research Quarterly,* **1**(2), 3-5.

37. Curbera, F., Khalaf, R., Mukhi, N., Tai, S. and Weerawarana, S. (2003) 'Service oriented computing: The next step in Web services'. *Communications of the ACM,* **46** (10).

38. Damsgaard, J. and Lyytinen, K. (2001) 'Building Electronic Trading Infrastructures: A Public or Private Responsability?' *Journal of Organizational Computing & Electronic Commerce,* **11**(2), 131-151.

39. Davidson, N. (2005). 'A Global Capacity Assessment and Needs Analysis '. *Proceedings of 39th Terminal Operating Conference,* Antwerp, Belgium.

40. Davidsson, P., Henesey, L., Ramstedt, L., Törnquist, J. and Wernstedt, F. (2005) 'An analysis of agent-based approaches to transport logistics'. *Transportation Research Part C,* **13**(4), 255—271.

41. de Langen, P. W. and Chouly, A. (2004) 'Hinterland Access Regimes in Seaports'. *European Journal of Transport and Infrastructure Research,* **4**(4), 361-380.

42. De Souza, G. A., Beresford, A. K. C. and Pettit, S. J. (2003) 'Liner Shipping Companies and Terminal Operators: Internationalization or Globalization? ‘Maritime *Economics and Logistics* **5**(4), 393–412.

43. Douma, A. M., Moonen, H. M., van Hillegersberg, J., van de Rakt, B. J. A. and Schutten, J. M. J. (2006) 'Designing agent based inter-organizational systems: Business and IOS alignment in the port of Rotterdam'.

44. Douma, A. M., Schutten, J. M. J. and Schuur, P. C. (2009) 'Waiting profiles: An efficient protocol for enabling distributed planning of barge rotations along terminals in the port of Rotterdam'. *Transportation Research Part C,* **17**(2), 133–157.

45. Durfee, E., Lesser, V. R. and Corkill, D. D. (1989) 'Trends in cooperative distributed problem solving'. *IEEE Transactions on Knowledge and Data Engineering,* **1**(1), 63-83.

46. Eder, J. and Liebhart, W. (1996). 'Workflow Recovery'. *Proceedings of the 5th IFCIS International Conference on Cooperative Information Systems (CoopIS 96)*, Brussels, Belgium. IEEE Computer Society Press.

47. Faratin, P., Sierra, C. and Jennings, N. R. (1998) 'Negotiation decision functions for autonomous agents'. *Robotics and Autonomous Systems,* **24** (3-4), 159-182.

48. Ferber, J. (1999) *Multi-agent systems: An introduction to Distributed Artificial Intelligence.* : Addison-Wesley, London.

49. Filos, E. and Ouzounis, V. (2000) 'Virtual Organizations: Technologies, Trends, Standards and the Contribution of the European RTD Programmers'. *International Journal of Computer Applications in Technology,* Special Issue: "Applications in Industry of Product and Process Modeling Using Standards".

50. FIPA (2002) *FIPA ACL Message Structure Specification*, Technical Report SC00061G

51. FIPA (2002) *FIPA SL Content Language Specification*, The Foundation for Intelligent Physical Agents (FIPA), Technical Report SC00008I

52. Fischer, K., Muller. J. P., Pischel, M. and Schier, D. (1995). 'A model for cooperative transportation scheduling.' *Proceedings of the First International Conference on Multiagent Systems.* Menlo Park, California. AAAI Press / MIT Press.

53. Frankel, E. G. (1987) *Port planning and development,* New York, U.S.A.: John Wiley & Sons.

54. Franklin, S. and Gasser, A. (1997). 'Is it an agent, or just a program: a taxonomy for autonomous agents'. *Proceedings of the 3rd International Workshop on Agent Theories, Architectures, and Languages*,

55. Franz, T., Voss, S. and Rölke, H. (2007). 'Market-mechanisms for integrated container terminal management'. *Proceedings of the 6th international conference on computer and IT applications in the maritime industry (COMPIT 2007)*, Cortona, Italy:

56. Gambardella, L. M., Rizzoli, A. E. and Zaffalon, M. (1998) 'Simulation and Planning of an Intermodal Container Terminal'. *Special Issue of Simulation Journal in Harbour and Maritime Simulation,* **21**(2), 107-116.

57. Ganek, A. and Corbi, T. (2003) 'The Dawning of the Autonomic Computing Era'. *IBM Systems Journal,* **42** (1), 5-18.

58. Georgakopoulos, D. (1998). 'Collaboration Management Infrastructure for Comprehensive Process and Service Management'. *Proceedings of International Symposium on Advanced Database Support for Workflow Management*, Enschede, The Netherlands:

59. Geppert, A. and Tombros, D. (1998). 'Event-based Distributed Workflow Execution with EVE'. *Proceedings of Middleware '98*, The Lake District, England:

60. Gibon, P., Clavier, J. F. and Loison, S. (1999). 'Support for Electronic Data Interchange, in Infrastructures for Virtual Enterprises'. *Proceedings of PRODNET Working Conference for Virtual Enterprises (PRO-VE'99)*, Boston: Networking Industrial Enterprises. IFIP TC5 WG5.3.

61. Görmer, J., Homoceanu, G., Mumme, C., Huhn, M. and Müller, J. (2011). 'JREP: Extending Repast Simphony for JADE Agent Behavior Components'. *Proceedings of IEEE / WIC / ACM International Conferences on Intelligent Agent Technology*, Lyon, France.

62. Grefen, J., Pernini, B. and Sanchez, G. (Eds.) (1999) *Database support for workflow management: The WIDE project*: Kluwer Academic Publishers, ISB7923-8414-8.

63. Grefen, J., Vonk, E., Boertjes, P. and Apers, P. (1998). 'Two-Layer Transaction Management for Workflow Management Applications'. *Proceedings of the 8th International Conference on Database and Expert System Applications*, Toulouse, France.

64. Grefen, P. (1998). 'Advanced Transaction Management in WIDE'. *Proceedings of International Symposium on Advanced Database Support for Workflow Management,* Enschede, The Netherlands:

65. Grefen, P. (2006). 'Towards Dynamic Interorganizational Business Process Management'. *Proceedings of the 15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'06)*,

66. Grizell, P. (2001) 'An evaluation of Port Community Systems: What can the Port of the Netherlands learn and use of other platforms'. *NedCargo,* (October 2001), Appear in: Smit, S., A Comparison of Port Community Systems A framework to compare Port Community Systems and an application to the Port Community Systems of Hamburg, Rotterdam and Antwerp, MSc Thesis in Maritime Economics and Logistics, Erasmus University, Rotterdam, 2004.

67. Hakim, S. W., Bowman, G. W. and Siedenstat, P. (2005) *Privatizing Transportation Systems. ,* Westport, CT: Praeger Publishers.

68. Hanseth, O. (2000) 'The Economics of Standards' in Associates, C. C. A. (ed.) *From Control to Drift: The Dynamics of Corporate Information Infrastructures*: Oxford University Press, 56-70.

69. Hanseth, O. and Lyytinen, K. (2006) 'Theorizing about the Design of Information Infrastructures: Design Kernel Theories and Principles'. *Sprouts: Working Papers on Information Systems,* **4**(12), 1-48.

70. Hanseth, O., Monteiro, E. and Hatling, M. (1996) 'Developing Information Infrastructure Standards: The Tension Between Standardization and Flexibility'. *Science, Technology and Human Values,* **21**(4), 407-426.

71. Henesey, L. (2002). 'Enhancing Terminal productivity through Artificial intelligence: Multi-Agent Approach.' *Proceedings of the 26th Terminal Operators Conference (TOC-2002)*, Antwerp, Belgium. :

72. Henesey, L. (2006) *Multi-Agent Container Terminal Management*. Thesis, (PhD Thesis). Blekinge Institute of Technology

73. Henesey, L., Davidsson, P. and Persson, J. (2009) 'Agent based simulation architecture for evaluating operational policies in transshipping containers'. *Autonomous Agents and Multi-Agent Systems,* **18**(2), 220 - 238.

74. Hevner, A. R., March, S. T., Park, J. and Ram, S. (2004) 'Design Science in Information Systems Research.' *MIS Quarterly,* **28** (1), 75–105.

75. Hock-Hai, T., Bernard, C. Y. T. and Kwok-kee, W. (1997) 'Organizational Transformation Using Electronic Data Interchange: The Case of TradeNet in Singapore'. *Journal of Management Information Systems,* **13**(4), 27.

76. Hogg, P., Chilcott, M., Nolan, B. and Srinivasan, K. (2004). 'An evaluation of Web services in the design of a B2B application'. *Proceedings of the 27th conference on Australasian computer science*,

77. Hollingsworth, D. (1995) *The Workflow Reference Model* Workflow Management Coalition, Technical Report TC00-1003, Issue 1.1, June 1995.

78. Huang, C. J., Trappey, A. J. C. and Yao, Y. H. (2006) 'Developing an agent-based workflow management system for collaborative product design '. *Industrial Management and Data Systems,* **106**(5), 680–699.

79. Huang, G. Q., Huang, J. and Mak, K. L. (2000) 'Agent-based workflow management in collaborative product development on the internet'. *Computer-Aided Design,* **32**(2), 133–144.

80. Itmi, M. M. D., Pecuchet, J.-P., Serin, F. and Villefranche, L. (1995). 'Ecoproblem solving for containers stacking'. *Proceedings of International Conference on Intelligent Systems for the 21st Century*, IEEE Transactions on Systems, Man, and Cybernetics.

81. Jeffrey, K. (1999) 'Recent Developments in Information Technology for Container Terminals'. *Cargo Systems, London.*

82. Jennings, N. (1996) 'Applying Agent Technology. '*Plenary presentation at PAAM'96. 1996*.

83. Jennings, N. and Bussmann, S. (2003) 'Agent-based control systems: Why are they suited to engineering complex systems? '*Control Systems Magazine,* **23**(3), 61–73.

84. Jennings, N., Sycara, K. and Wooldridge, M. (1998) 'A roadmap of agent research and development'. *Autonomous Agents and Multi-Agent Systems,* **1**(1), 7-38.

85. Jennings, N. R. (2000) 'On agent-based software engineering'. *Artificial Intelligence,* **117**(2), 277-296.

86. Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Sierra, C. and Wooldridge, M. (2001) 'Automated negotiation prospects, methods and challenges'. *International Journal of Group Decision and Negotiation,* **10** (2), 199-215.

87. Jimenez-Martinez, J. and Polo-Redondo, Y. (2004) 'The influence of EDI adoption over its perceived benefits'. *Technovation,* **24**(1), 73-79.

88. Johnston, H. R. and Carrico, S. R. (1988) 'Developing Capabilities to Use Information Strategically.' *MIS Quarterly,* **12**(1), 37-48.

89. Kambil, A. and van Heck, E. (2002) *Making Markets: How Firms Can Design and Profit from Online Auctions and Exchanges*: Harvard Business School Press, Boston, MA,

90. Keceli, Y., Choi, H., Cha, Y., Aydogdu, Y. V. and Kim, H. (2008) 'A Study on User Evaluation of PORT-MIS.' *Entrue Journal of Information Technology,* **7**(2), 165–175.

91. Keceli, Y. and Choi, H. R. (2008) 'Level of information systems in Turkish public ports and direction of improvement'. *Int. J. Logistics Systems and Management,* **4**(6).

92. Keceli, Y., Choi, H. R., Cha, Y., Aydogdu, Y. V. and Kim, H. (2008). 'A Study on Adoption of Port Community Systems According to Organization Size'. *Proceedings of the 2008 Third International Conference on Convergence and Hybrid Information Technology - Volume 01, IEEE Computer Society*,

93. Keceli, Y., Choi, H. R., Cha, Y. S. and Aydogdu, Y. V. (2008). 'A Study on User Acceptance of Port Community Systems'. *Proceedings of International Conference on Value Chain Sustainability (ICOVACS 2008)*, zmir, Turkey:

94. Keceli, Y., Choi, H. R., Kim, H., Kwon, H. K. and Choi, P. J. (2006). 'Improvement of Port Information Systems for TCDD Ports'. *Proceedings of 4th Logistics and Supply Chain Congress*, Izmir, Turkey:

95. Keceli, Y., Choi, H. R. and Park, N. K. (2007) 'Analysis of Success Factors of Information Systems Development in Kumport and Implications for Other Turkish Ports'. *WSEAS Transactions on Information Science and Applications,* **4**(5), 1014-1047.

96. Khare, R. and Rifkin, A. (1999) 'XML: A Door to Automated Web Applications'. *IEEE Internet Computing,* **1**(4), 78-87.

97. Kia, M., Shayan, E. and Ghotb, F. (2000) 'The importance of information technology in port terminals operations'. *International Journal of Physical Distribution & Logistics Management,* **30**(3/4), 331-344.

98. King, J. L. and Konsynski, B. (1990) Teaching Case 9–191–26 *Hong Kong TradeLink: News from the second city*, Boston: Harvard Business School,

99. Kleindorfer, P. R. and Saad, G. H. (2005) 'Managing Disruption Risks in Supply Chains'. *Production and operations management,* **14**(1), 53.

100. Klingemann, J. W. and Aberer, J. K. (1999). 'Deriving Service Models in Cross-Organizational Workflows'. *Proceedings of RIDE-Information Technology for Virtual Enterprises*, Sydney, Australia.

101. Kreger, H. (2001) IBM Web Services *Web Services Conceptual Architecture (WSCA 1.0)*, IBM Software Group,

102. Kumar, K. and van Dissel, H. G. (1996) 'Sustainable collaboration: Managing conflict and cooperation in interorganizational systems'. *MIS Quarterly,* **20**(3), 279-300.

103. Kuo, M. H. (2007) 'An intelligent agent-based collaborative information security framework'. *Expert Systems with Applications,* **32**(2), 585–598.

104. Kushniruk, A. (2002) 'Evaluation in the design of health information systems: application of approaches emerging from usability engineering'. *Computers in Biology and Medicine* **32**(3), 141-149.

105. Lee, G. G. and Lin, H. F. (2005) 'Influence of environmental and organizational factors on the success of internet-based interorganizational systems planning'. *Internet Research,* **15**(5), 527-543.

106. Lee, T.-W., Park, N.-K. and Lee, D.-W. (2002) 'Design of Simulation System for Port Resources Availability in Logistics Supply Chain'. *Proceedings of International Association of Maritime Economists Annual Conference*, Panama City, Panama:

107. Lipinski, M. E. and Clarke, D. B. (2008) 'Resolution of Land Use and Port Access Conflicts at Inland Waterway Ports'. *In Transportation Research Record: Journal of the Transportation TRB 2008 Annual Meeting*.

108. Lipnack, J. and Stamps, J. (1994) *The age of the network: organizing principles for the 21st century,* New York, NY: John Wiley & Sons.

109. Liu, S. (1998) 'Strategic scanning and interpretation revisiting: foundations for a software agent support system – Part 2. Scanning the business environment with software agents'. *Industrial Management & Data Systems,* **98** (8), 362-72.

110. Lokuge, P. and Alahakoon, D. (2007) 'Improving the adaptability in automated vessel scheduling in container ports using intelligent software agents'. *European Journal of Operational Research* **177**(1), 985-2015.

111. Long, A. (2009) 'Port Community systems. '*World Customs Journal,* **3**(1), 63-67.

112. Madhusudan, T. (2005) 'An agent-based approach for coordinating product design workflows'. *Computers in Industry,* **56** (3), 235–259.

113. Maes, P. (1995) 'Artificial life meets entertainment: life like autonomous agents'. *Communication of the ACM,* **38** (11), 108-114.

114. Mahr, T., Srour, J., deWeerdt, M. M. and R. Zuidwijk (2008). 'Agent Performance in Vehicle Routing when the Only Thing Certain is Uncertainty'. *Proceedings of 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Estoril, Portugal:

115. Malone, T. W. and Crowston, K. (1994) 'Interdisciplinary study of coordination'. *ACM Computing Surveys,* **26**(1), 87-119.

116. March, S. and Smith, G. (1995) 'Design and natural science research on information technology'. *Decision Support Systems,* **15**(4), 251-266.

117. Marianos, N., Lambrou, M. A. and Nikitakos, N. (2007). 'Emergent Port Services in the Context of Information and Communication Technologies'. *Proceedings of the International Association of Maritime Economist (IAME) Conference*, Athens, Greece:

118. Markus, M. L., Steinfield, C. and Wigand, R. (2006) 'Industry-wide Information Systems Standardization as Collective Action: The Case of the US Residential Mortgage Industry'. *MISQ Quarterly,* **30**(Special Issue on Standard Making), 439-465.

119. Marlow, P. B. and Paixão, A. C. (2003) 'Measuring lean ports performance'. *International Journal of Transport Management,* **1**(4), 189-202.

120. Martin, J. and Thomas, B. J. (2001) 'The container terminal community'. *Maritime Policy & Management,* **28**(3), 279-292.

121. Massawe, L. V., Kinyua, J. and Aghdasi, F. (2010) An implementation of a multi-agent based RFID middleware for asset management system using the JADE platform Durban, South Africa, IEEE Computer Society.

122. Mes, M., Van der Heijden, M. and Van Harten, A. (2007) 'Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems'. *European Journal of Operation Research,* **181**(1), 59-75.

123. Mes, M., Van der Heijden, M. and Van Hillegersberg, J. (2008) 'Design choices for agent-based control of AGVS in the dough making process'. *Decision Support Systems,* **44**(4), 983-999.

124. Mila, S. (2007). 'Information Technologies in Ports: Port Community Systems'. *Proceedings of the 25th World Ports Conference of IAPH*, Houston, Texas, USA: The International Association of Ports and Harbours (IAPH).

125. Mila, S. (2009) 'Port Community System (PCS), its present & future.' *Sail to the Future,* **2**(1).

126. Miller, M. J. (1996) 'Your Own Private Internet'. *PC Magazine,* **15**(5), 29.

127. Monteiro, E. and Hanseth, O. (1995) 'Social shaping of information infrastructure: on being specific about the technology' in Orlikowski, W. J., Walsham, G., Jones, M. R. & DeGross, J. I. (eds.) *Information Technology and Changes in Organizational Work*: Chapman & Hall ,

128. Myers, M. D. (1997) 'Qualitative Research in Information Systems'. *MIS Quarterly*

129. Nikolai, C. and Madey, G. (2008) 'Tools of the Trade: A Survey of Various Agent Based Modeling Platforms'. *Journal of Artificial Societies and Social Simulation,* **12**(2).

130. North, M. J., Collier, N. T. and Vos, J. R. (2007) 'Experiences Creating Three Implementations of the Repast Agent Modeling Toolkit'. *ACM Transactions on Modeling and Computer Simulation,* **16**(1), 1-25.

131. Notteboom, T. (2008) Discussion paper / OECD, 2008:10 The *Relationship between Seaports and the Intermodal Hinterland in Light of Global Supply Changes: European Challenges*, Paris: OECD,

132. Notteboom, T. and Winkelmans, W. (2002). 'Stakeholders relations management in ports: dealing with the interplay of forces among stakeholders in a changing competitive environment '. *Proceedings of International Association of Maritime Economists Annual Conference (IAME 2002)*, Panama City:

133. Notteboom, T. and Winkelmans, W. (2004) Factual Report on the European Port Sector: FR-WP1 Overall *Market Dynamics and Their Influence on the Port Sector*, Brussels: European Sea Ports Organisation (ESPO),

134. Notteboom, T. E. and Rodrigue, J.-P. (2005) 'Port regionalization: towards a new phase in port development'. *Maritime Policy & Management,* **32**(3), 297 - 313.

135. Nwana, H. (1996) 'Software agents: an overview'. *Knowledge Engineering Review,* **11**(3), 205-244.

136. O'Brien, P. D. and Wiegand, M. E. (1998) 'Agent based process management: Applying intelligent agents to workflow'. *The Knowledge Engineering Review,* **13** (2), 161–174.

137. OASIS (2006) 'WS-BPEL (Web Services Business Process Execution Language) 2.0 '. *At: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel.*

138. Odgers, B. R., Shepherdson, J. W. and Thompson, S. G. (1999). 'Distributed workflow co-ordination by proactive software agents.' *Proceedings of Intelligent Workflow and Process Management. The New Frontier for AI in Business IJCAI99 Workshop (IJCAI-99).*

139. Oliveira, E., Fischer, K. and Stepankova, O. (1999) 'Multi-agent systems: Which research for which applications'. *Robotics and Autonomous Systems,* **27** (1), 91–106.

140. Orchard, R. (2001) Fuzzy Reasoning in Jess: The FuzzyJ Toolkit and FuzzyJess *Third International Conference on Enterprise Information Systems (ICEIS), pp. 533-542. .* Setubal, Portugal.

141. Ouzounis, V. and Tschammer, V. (1999). 'A Framework for Virtual Enterprise Support Services"'. *Proceedings of the 32nd International Conference on Systems and Sciences (HICSS32)*, Maui, Hawaii:

142. Paixão, A. C. and Marlow, P. B. (2003) 'Fourth generation ports – a question of agility?' *International Journal of Physical Distribution & Logistics Management,* **33** (4), 355 - 376.

143. Panayides, P. M. (2002) 'Economic organization of intermodal transport'. *Transport Reviews 22*(4), 401 - 414.

144. Panayides, P. M. (2006) Container ports and global supply chains: Integration parameters and performance outcomes Invited Speech to Faculty Seminar, National University of Singapore, June 15.

145. Panayides, P. M. and Song, D.-W. (2006). 'Supply chain orientation and port performance'. *Proceedings of International Association of Maritime Economists Conference (IAME)*, Melbourne, Australia.

146. Peffers, K., Tuunanen, T., Rothenberger, M. and Chatterjee, S. (2008) 'A Design Science Research Methodology for Information Systems Research'. *Journal of Management Information Systems,* **24**(3), 45-77.

147. PORTEL (2009) *Inventory of Port Single Windows and Port Community Systems.* SKEMA, Sustainable Knowledge Platform for the European Maritime and Logistics Industry.

148. Posti, A., Häkkinen, J. and Tapaninen, U. (2011) 'Promoting information exchange with a port community system – case Finland' in Kersten, W., Blecker, T. & Jahn, C. (eds.) *international Supply Chain Management and Collaboration Practices*,

149. Premkumar, G., Ramamurthy, K. and Crum, M. R. (1997) 'Determinants of the EDI adoption in the transportation industry'. *European Journal of Information Systems,* **6**(2), 107-121.

150. PSA (2006)

News Release in October 2006 (http://www.portnet.com/pdf/061010-national-it-award.pdf),

151. Railsback, S., Lytinen, S. and Jackson, S. (2006) 'Agent-based Simulation Platforms: Review and Development Recommendations'. *SIMULATION,* **82**(9), 609-623.

152. Rebollo, M., Julian, V., Carrascosa, C. and Botti, V. (1999). 'A Multi-Agent System for the Automation of a Port Container Terminal'. *Proceedings of Autonomous Agents 2000 workshop on Agents in Industry*, Barcelona, Spain:

153. Robinson, R. (2002) 'Ports as Elements in Value-Driven Chain Systems: The New Paradigm'. *Maritime Policy & Management,* **29**(3), 241 - 255.

154. Rodon, J. and Ramis-Pujol, J. (2006). 'Exploring the Intricacies of Integrating with a Port Community System'. *Proceedings of 19th Bled eConference eValues*, Bled, Slovenia.

155. Sandholm, T. W. (1999) 'Distributed Rational Decision Making' in Weiss, G. (ed.) *Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence*: MIT press, 201—258.

156. Serenko, A. and Detlor, B. (2002). 'Agent Toolkits: A General Overview of The Market and an Assessment of Instructor Satisfaction with Utilizing Toolkits in the Classroom. 'Proceedings *of Working Paper #455.*, McMaster University, Hamilton, Ontario, Canada

157. Serenko, A. and Detlor, B. (2003) 'Agent toolkit satisfaction and use in higher education'. *Journal of Computing in Higher Education,* **15**(1), 65-88.

158. Smit, S. (2004) *A Comparison of Port Community Systems A framework to compare Port Community Systems and an application to the Port Community Systems of Hamburg, Rotterdam and Antwerp*. Thesis, (M. Sc.). Erasmus University

159. Song, D.-W. and Panayides, P. M. (2008) 'Global Supply Chain and Port/Terminal: Integration and Competitiveness'. *Maritime Policy & Management* **35**(1), 73-87.

160. Sridharan, U. V., Caines, W. R. and Patterson, C. (2005) 'Implementation of supply chain management and its impact on the value of firms'. *Supply Chain Management-an International Journal* **10**(3-4), 313-318.

161. Srour, J., van Oosterhout, M. P. A., Zuidwijk, R. A. and Van Baalen, P. (2008) 'Port Community System Implementation: Lessons Learned from International Scan.' *In Transportation Research Board 87th Annual Meeting,* (Washington DC.).

162. Star, S. L. and Ruhlender, K. (1996) 'Steps Toward an Ecology of Infrastructure: Design and Access for Large Information Spaces'. *Information Systems Research,* **7**(1), 11-134.

163. Straub, D., Boudreau, M.-C. and Gefen, D. (2004) 'Validation Guidelines For IS Positivist Research.' *Communications of the Association for Information Systems,* **13** (Article 24), 380–427.

164. Suzuki, Y. and Williams, L. R. (1998) 'Analysis of EDI resistance behavior'. *Transportation Journal,* **37**(4), 36-44.

165. Thurston, T. and Hu, H. (2002). 'Distributed Agent Architecture for Port Automation'. *Proceedings of twenty-Sixth International Computer Software and Applications Conference*, Oxford, UK: IEEE Computer Society.

166. Tijan, E., Kos, S. and Ogrizović, D. (2009) 'Disaster recovery and business continuity in port community systems'. *Pomorstvo,* **23**(1), 243–260.

167. Trappey, A. J. C., Trappey, C. V. and Lin, F. T. L. (2006) 'Automated silicon intellectual property trade using mobile agent technology'. *Robotics and CIM,* **22** (3), 189–202.

168. UNCTAD (1992a) Report by the UNCTAD secretariat, *Development and Improvement of Ports: The Principles of Modern Port Management and Organisation*, Geneva, Switzerland: United Nations Conference on Trade and Development (UNCTAD),

169. UNCTAD (1992b) Report by the UNCTAD secretariat, *Port Marketing and the Challenge of the Third Generation Port*, Geneva, Switzerland: United Nations Conference on Trade and Development (UNCTAD),

170. van Baalen, P., Zuidwijk, R. and Van Nunen, J. (2008) 'Port Inter-Organizational Information Systems: Capabilities to Service Global Supply Chains'. *Foundations and Trends in Technology, Information and Operations Management,* **2**(2-3), 81-241.

171. van Baalen, P. J., van Oosterhout, M., Tan, Y. H. and van Heck, E. (2000) *Dynamics in setting up an EDI community : experiences from the port of Rotterdam,* Delft, The Netherlands: Eburon Publishers.

172. van Der Horst, M. R. and De Langen, P. W. (2008) 'Coordination in hinterland transport chains: A major challenge for the seaport community'. *Maritime Economics and Logistics,* **10**(1-2), 108-129.

173. van Heck, E. and Ribbers, P. M. (1999). 'The Adoption and Impact of EDI in Dutch SME's'. *Proceedings of the 32nd Hawaii International Conference on System Sciences*, Hawaii:

174. Verschuren, P. and Hartog, R. (2005) 'Evaluation in Design-Oriented Research'. *Quality and Quantity,* **39**(6), 733-762.

175. W3C (2002) 'Web Services Architecture Working Group. Web Services Architecture Requirements'. *http://www.w3.org/TR/2002/WD-wsa-reqs-20020429*.

176. Wagenaar, R. W. (1992). 'Business network redesign: Lessons from the Port of Rotterdam Simulation game"'. *Proceedings of 5th International Conference on EDI, 390-404*, Slovenia: Bled:

177. Wang, M., Wang, H. and Xu, D. (2005) 'The design of intelligent workflow monitoring with agent technology'. *Knowledge-Based Systems,* **18** (6), 257–266.

178. Wang, S., Shen, W. and Hao, Q. (2006) 'An agent-based web service workflow model for inter-enterprise collaboration'. *Expert Systems with Applications,* **31**(4), 787–799.

179. Weiss, G. (1999) *Multi-agent systems: A modern approach to distributed artificial intelligence.* : The MIT Press, Cambridge, Massachusetts, London, England.

180. WfMC (1999) Workflow Management Coalition *Terminology and Glossary*, Workflow Management Coalition Specification, WFMC-TC1011, Issue 3, Feb. 1999

181. Wil, M. P. and Aalst, V. (1999) 'Process-Oriented architectures For Electronic Commerce & Interorganizational Workflow'. *Information Systems* **24**(8), 639-671.

182. Wohlin, C. (2007) 'An analysis of the most cited articles in software engineering journals'. *Information and Software Technology,* **49**(1), 2-11.

183. Wooldridge, M. (1998) 'Agent-based computing'. *Interoperable Communication Networks,* **1**(1), 71-97.

184. Wooldridge, M. (2005) *An introduction to multi-agent systems.* : John Wiley & Sons, Ltd.

185. Wooldridge, M. and Jennings, N. R. (1994). 'Agent theories, architectures, and languages: a survey'. *Proceedings of ECAI-Workshop on Agent Theories, Architectures and Languages*, Amsterdam, The Netherlands.

186. Wooldridge, M. J. and Jennings, N. R. (1995) 'Intelligent Agents: Theory and Practice'. *Knowledge Engineering Review,* **10**(2), 115–152.

187. Xu, D., & Wang, H. (2002) 'Multi-agent collaboration for B2B workflow monitoring '. *Knowledge-Based Systems,* **15**485–491.

188. Yi, D. W., Kim, S. H. and Kim, N. H. (2002). 'Combined Modeling with Multi-Agent Systems and Simulation: Its Application to Harbor Supply Chain Management'. *Proceedings of Thirty-fifth Hawaii International Conference on System Sciences*, Hawaii:

189. Yin, R. K. (1994) *Case Study Research,* Thousand Oaks, Ca. US.: Sage Publishing.

190. Yoo, M.-J. and Glardon, R. (2009) 'Combining JADE and Repast for the Complex Simulation of Enterprise Value-Adding Networks' in Michael, L. & Jorge, J. G.-S. (eds.) *Agent-Oriented Software Engineering IX*: Springer-Verlag, 243-256.

191. Zhuge, H. (2003) 'Workflow- and agent-based cognitive flow management for distributed team cooperation'. *Information and Management,* **40** (5), 419–429.

Appendix A

Organizations in Port Communities

Appendix A: Organizations in Port Communities

Organization	Role(s)
Bank	The Bank facilitates the payment of service charges by the stakeholders to the port and other statutory charges like excise/stamp duties and taxes to Customs and other government agencies.
Barge operator (or Inland shipping operator)	Operator of inland shipping vessels. The inland shipping operator is a logistic service provider focused on the broad service offering in container transport between seaports and inland terminals via inland vessels. They aim to offer frequent reliable services with large vessels between the large number of terminals in the seaport and one or more inland terminals in the hinterland. They seek for optimal occupancy of their vessels with full load containers.
Carrier (road / rail / barge)	This is the party that performs, for import, the shift (transport) of the containers or cargo from terminal to consignee (or designated CFS[1]) in the port of discharge. For export, this is the party that performs the transport of the containers or cargo from shipper (or designated CFS) to terminal in the port of loading.
Container Agent	They represent the container Lines and provide the steamer agent with details of the containers belonging to them in the vessel. They also give the delivery order to the Clearing Agent/Importer for clearing the container. Similarly, the liner gives the authorization for loading containers to a vessel to the clearing agent. They recover terminal handling charges and container detention charges and also issue bill of lading and agent’s delivery order to the consignees.

[1] Container Freight Station (CFS) or Inland Container Depot (ICD) is a shed where cargoes from several different consignors are received, aggregated and stuffed into a container; or where cargoes fro several consignees are unpacked from a container for delivery. Functionally there is no difference between CFS and ICD. While CFS is an off dock facility located near the servicing port, ICD is generally located outside the port towns.

Organization	Role(s)
Container Terminal Operator	The terminal operator manages the terminal operations and receives, stores, loads/discharges and delivers the containers/cargo within the terminal. They collect the terminal handling charges and storage charges while the port collects the vessel related charges.
Customs	Customs performs both administrative and physical controls, and a legitimate authority in charge of collecting duty and taxes. Customs also enforces the provisions of the Customs Act governing imports and exports of cargo, arrival and departure of vessels and prevention of smuggling. Customs grants entry inward / entry outwards permission for the vessel and clear goods for both import and export.
Freight Forwarder (merchant haulage) / Customs Agent / Clearance Agent	A party that can act as an agent for the shipper or the consignee. The forwarder performs the task of organizing the dispatch of goods including the necessary documentation. A forwarder has to arrange transport, Customs formalities, and insurance of goods during transport, etc. on behalf of a shipper or consignee. A Clearance agent helps the importer/exporter in getting the customs clearance for his consignments. A Clearance agent declares the goods with Customs, and gets goods examined by Customs, pays the duty into the Customs designated bank and takes the physical possession of the cargo from the Custodian after paying all Port dues/charges.
Importer (receiver or consignee) / Exporter (shipper or consignor)	Importers / exporters are the owners of the goods being imported / exported. They are mainly responsible for completion of customs and Port formalities either themselves or through an agent (a Customs Agent or a Clearance Agent).
Inspection authority	The agency that is responsible for inspecting food/agricultural products entering the port, for wholesomeness, fitness and compliance with governmental legislation. Cargo samples are inspected and permitted for domestic use based on lab certification.

Organization	Role(s)
Port Authority	This is a governmental establishment in charge of development and promotion of port equipment and ensuring the compliance of regulations. As a land manager, the port authority is responsible for the safe, sustainable and competitive development of the port. The landlord function of the port comprises the development, management and control of the port area, including nautical access and port infrastructure, taking into account safety and environmental issues. The port authority takes care of the allocation of berth, piloting vessels in and out of the port and may also provide other vessel related and cargo related services.
Quarantine	The agency that is responsible for the inspection of hygiene in the ship and amongst the crew, so as to control the spread of infectious diseases from incoming vessels and aircraft. On arrival, Quarantine also issues a de-ratting exemption certificate to the vessel.
Rail operator	The rail operator is responsible for loading and unloading of containers onto or from the trains. Rail operators sometimes have their own rail terminals. The rail operator runs scheduled block train services connecting the container terminals and inland container depots on a regular schedule of services, in some instances offered as a combined rail-road service in partnership with a road haulier.
Road carrier (or road hauler)	Their main interest is the optimal allocation of their fleet by combining runs and preventing *"empty runs"*. The road carrier is the first or final link in the chain from the shipper to the receiving party and therefore it will be confronted most heavily with waiting times. Communication with the terminals is therefore essential.
Shipping Line (Sea Carrier)	This is a company that operates a fleet of ships and other carriers. A shipping line provides the regular service of transporting goods over sea. The may also offer the shipper a comprehensive door-to-door services and integrated logistic packages, in some cases managing the container terminal operation. Their sphere of interest has extended inland and they may responsible for contracting the majority of inland hauls with the road and rail operators.

Organization	Role(s)
Shipping Line Agent / Logistics Service Provider (carrier haulage)	The Shipping line Agent performs particular functions and services on behalf of Shipping Lines. A shipping line agent is either a part of the shipping line’s organization or an independent body. A shipping agent coordinates the arrangements for ship arrival, berthing, discharge/loading and departure of the vessel with port, customs and other regulatory bodies.
Stevedore	Stevedores arrange for the labour gangs to handle containers, trailers, pallets, boxes, etc., maintain the tally sheets, and report the cargo loading and discharging details to the port and shipping agent. They also ensure the transport of handling units though the port area.
Surveyor	The Surveyor is hired by the port/ Importer/Exporter as well as by the shipping agent to survey the vessel, container and cargo. The survey reports are used for ascertaining the damages, if any. The surveyors also monitor the stevedoring activities and co-ordinate stuffing/de-stuffing, movement of containers/cargo and loading/discharging.

Appendix B

Part of TruckAgent Source code

```
package agents;
import jade.content.lang.Codec.CodecException;
import jade.content.lang.sl.SLCodec;
import jade.content.onto.OntologyException;
import jade.content.onto.UngroundedException;
import jade.content.onto.basic.Action;
import jade.core.AID;
import jade.domain.FIPANames;
import jade.domain.FIPAAgentManagement.ServiceDescription;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.MessageTemplate;
public class TruckAgent extends MyAgent {
	private static final long serialVersionUID = -
	7856763770847928308L;
	private String currentOrderID = "no_order";
	private long[] homeLocation = new long[2];
	private long[] currentLocation = new long[2]; .
	private double eta = 0;
	private long statusChanged;
	private String currentOrderPickUp;
	private String currentOrderDelivery;
	private String currentOrderReturn;
	protected ServiceDescription[] sd; // The service description DF registration)
	private String truckStatus;
	private boolean firstOrder = true;
	private long startTime; // The starttime of the truck
	/* ---------------------------------------------- */
	protected void setup() {
		DEBUG = false;
		// The start address of the truck.
		String startZipcode = "2676";
		// Register the codec for the SL0 language.
		getContentManager().registerLanguage(new SLCodec();
		FIPANames.ContentLanguage.FIPA_SL0);
		// Register the ontology used by this application.
		getContentManager().registerOntology(OrderDataOntology.getInstance());

getContentManager().registerOntology(OrderAddressesOntology.getInstance());
		// Register the agent in the yellow pages as an order agent.
		sd = new ServiceDescription[3];
		ServiceDescription tempSD = new ServiceDescription();
		tempSD.setType("truck");
		tempSD.setName(getLocalName() + "-truck");
		sd[0] = tempSD;
		// Register currently a participant; other option is:'initiator'.
		tempSD = new ServiceDescription();
		tempSD.setType("participant");
		tempSD.setName(getLocalName() + "-participant");
		sd[1] = tempSD;
		// Register truck idle; other options are: 'busy' and anticipating'.
```

```
			tempSD = new ServiceDescription();
			tempSD.setType("idle");
			tempSD.setName(getLocalName() + "-idle");
			sd[2] = tempSD;
			register(this, sd);
			truckStatus = "idle";
			statusChanged = System.currentTimeMillis();
			// Start behaviours after receiving rate and time.
			addBehaviour(new InitialBehaviour(this));
		}
		/* --------------------------------------------- */
		protected void takeDown() {
			// Printout a dismissal message
			debug("Agent " + getAID().getLocalName() + " terminating.");
			deregister(this);
		}
		/* --------------------------------------------- */
		public int getOrderStatus() {
			if (!currentOrderID.equals("no_order")) {
				// Request the status from the OrderAgent.
				ACLMessage msg = new
				ACLMessage(ACLMessage.REQUEST);
				msg.addReceiver(new AID(currentOrderID,
								AID.ISLOCALNAME));
				msg.setConversationId("GET_STATUS");
				msg.setReplyWith("status" + System.currentTimeMillis());
				this.send(msg);
				// Wait for a reply.
				MessageTemplate mt =

MessageTemplate.and(MessageTemplate.MatchConversationId("GET_STAT
									US"),

MessageTemplate.MatchInReplyTo(msg.getReplyWith()));
					ACLMessage reply = this.receive(mt);
					// As long as there is no reply, the thread pause (steps
					10
					millisecs)
					while (reply == null) {
						try {
							Thread.sleep(10);
						} catch (InterruptedException e) {
							e.printStackTrace();
						}
						reply = this.receive(mt);
					}
				// Reply received; retrieve the current time in seconds.
				int status = Integer.parseInt(reply.getContent());
				return status;
			} else
```

```
				return 0;
		}
		/* ------------------------------------------------ */
		public double getETA() {
			return eta;
		}
		public String getCurrentOrderID() {
			return currentOrderID;
		}
		public long getStatusChanged() {
			return statusChanged;
		}
		public String getOrderAnticipatingUpon() {
			return orderAnticipatingUpon;
		}
		public String getTruckStatus() {
			return truckStatus;
		}
		public boolean getFirstOrder() {
			return firstOrder;
		}
		public long getRate() {
			return rate;
		}
		/* ------------------------------------------------ */
		/**
		 * Set the current order adresses. Request these from the OrderAgent.
		 */
		public void setCurrentOrderAddresses() {
			// Request the data from the Order Agent.
			ACLMessage msg = new ACLMessage(ACLMessage.REQUEST);
			msg.addReceiver(new AID(currentOrderID, AID.ISLOCALNAME));
			msg.setConversationId("GET_ADDRESSES");
			msg.setReplyWith("order_addresses" + System.currentTimeMillis());
			this.send(msg);
			// Wait for a reply.
			MessageTemplate mt =

MessageTemplate.and(MessageTemplate.MatchConversationId("GET_ADDRESSES"),

MessageTemplate.MatchInReplyTo(msg.getReplyWith()));
			ACLMessage reply = this.receive(mt);
			// As long as there is no reply, the thread will pause (in steps of 10
milliseconds)
			while (reply == null) {
				try {
					Thread.sleep(10);
				} catch(Exception e) {
					e.printStackTrace();
				}
```

```
					reply = this.receive(mt);
				}
				reply.getContent();
				// Initialize the result object
				OrderAddresses result = null;
				try {
					Action a =

(Action)super.getContentManager().extractContent(reply);
					result = (OrderAddresses)a.getAction();
					currentOrderPickUp = result.getPickupAddress();
					currentOrderDelivery = result.getDeliveryAddress();
					currentOrderReturn = result.getReturnAddress();
				} catch (UngroundedException e) {
					e.printStackTrace();
				} catch (CodecException e) {
					e.printStackTrace();
				} catch (OntologyException e) {
					e.printStackTrace();
				}
		}
		/**
		 * Sets the order status of the truck. This means that the OrderAgent is updated on
		 * its new status, and that the TruckAgent updates its description with the DF.
		 * @param orderStatus the OrderStatus of the truck
		 */
		protected void setOrderStatus(int orderStatus) {
				// Update the description of the TruckAgent in the DF registry
				// First, set whether the truck is available or not
				if (orderStatus < 7) {
					ServiceDescription tempSD = new ServiceDescription();
					tempSD.setType("busy");
					tempSD.setName(getLocalName() + "-busy");
					sd[2] = tempSD;
				} else if (orderStatus == 7) {
					ServiceDescription tempSD = new ServiceDescription();
					tempSD.setType("idle");
					tempSD.setName(getLocalName() + "-idle");
					sd[2] = tempSD;
				}
				modify(this, sd);
				// Send a message to the OrderAgent to update it's status.
				ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
				msg.addReceiver(new AID(currentOrderID, AID.ISLOCALNAME));
				msg.setConversationId("UPDATE_STATUS");
				msg.setContent(Integer.toString(orderStatus));
				this.send(msg);
				super.debug("Order status changed; message forwarded to the OrderAgent");
				if (orderStatus == 1) {
					setCurrentOrderAddresses();
```

```
		} else if (orderStatus == 7) {
			// Clear variables.
			currentOrderPickUp = null;
			currentOrderDelivery = null;
			currentOrderReturn = null;
			currentOrderID = "no_order";
			this.setTruckStatus("idle");
			super.debug("Order completed.");
			// Find new order.
			msg = new
			ACLMessage(ACLMessage.REQUEST);
			msg.addReceiver(this.getAID());
			msg.setConversationId("FIND_ORDER");
			this.send(msg);
		}
	}
	/* ------------------------------------------------ */
	protected void setETA(double eta) {
		this.eta = eta;
	}
	/* ------------------------------------------------ */
	protected void setCurrentOrderID(String currentOrderID) {
		this.currentOrderID = currentOrderID;
	}
	/* ------------------------------------------------ */
	protected void anticipateOrder(String orderID) {
		this.currentOrderID = null;
		this.eta = 0;
		//new TruckUtil(this).addOrder(currentOrder.getID());
		// Send a message to inform the OrderAgent that it has a truck
		anticipating
		ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
		msg.addReceiver(new AID(orderID, AID.ISLOCALNAME));
		msg.setConversationId("UPDATE_ANTICIPATE");
		msg.setContent("true");
		msg.setReplyWith("update_anticipate" +
				System.currentTimeMillis());
		super.send(msg);
		super.debug("Truck started to anticipate! Message forwarded to
OrderAgent");
				// Wait for a reply.
				MessageTemplate mt =
					MessageTemplate.and(MessageTemplate.

MatchConversationId("UPDATE_ANTICIPATE"),

MessageTemplate.MatchInReplyTo(msg.getReplyWith()));
				ACLMessage reply = this.receive(mt);
				// As long as there is no reply, the thread will pause (in
				steps of 10 milliseconds)
```

```
                    while (reply == null) {
                        try {
                            Thread.sleep(10);
                        } catch(Exception e) {
                            e.printStackTrace();
                        }
                        reply = this.receive(mt);
                    }
        String answer = reply.getContent();
        // Only if DONE returns, than do:
        if (answer.equalsIgnoreCase("done")){
            // Update the description in the DF registery
            ServiceDescription tempSD = new ServiceDescription();
            tempSD.setType("anticipating");
            tempSD.setName(getLocalName() + "-anticipating");
            sd[3] = tempSD;
            modify(this, sd);
            this.orderAnticipatingUpon = orderID;
            // The message "not_allowed" returned; which means that the
            order is anticipated
            // upon by another truck already; therefore, restart the search
        } else {
            this.orderAnticipatingUpon = "null";
            // Find new order.
            msg = new ACLMessage(ACLMessage.REQUEST);
            msg.addReceiver(this.getAID());
            msg.setConversationId("FIND_ORDER");
            this.send(msg);
            super.debug("Order anticipation went wrong... find next
                        order...!");
        }
    }
    /* ---------------------------------------------- */
    protected void clearAnticipateOrder() {
        // Send a message to inform the OrderAgent that the truck stopped
        anticipating
        ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
        msg.addReceiver(new AID(this.orderAnticipatingUpon,
                    AID.ISLOCALNAME));
        msg.setConversationId("UPDATE_ANTICIPATE");
        msg.setContent("false");
        super.send(msg);
        super.debug("Truck informed order that it stopped anticipation on an
                    order!”);
        // Update the agentdescription in the DF registery
        ServiceDescription tempSD = new ServiceDescription();
        tempSD.setType("idle");
        tempSD.setName(getLocalName() + "-idle");
        sd[2] = tempSD;
        modify(this, sd);
```

```
		// Clear the variable in the agent
		this.orderAnticipatingUpon = "null";
	}
	/* ------------------------------------------------ */
	protected void setFirstOrder(boolean firstOrder) {
		this.firstOrder = firstOrder;
	}
	protected void setOrderAnticipatingUpon(String
				orderAnticipatingUpon) {
		this.orderAnticipatingUpon = orderAnticipatingUpon;
	}
	protected void setRate(long rate) {
		this.rate = rate;
	}
	public void setStartTime(long startTime) {
		this.startTime = startTime;
	}
	/* ------------------------------------------------ */
	protected void setTruckStatus(String newTruckStatus) {
		long currentTime = System.currentTimeMillis();
		String timeActivity = Long.toString(currentTime - statusChanged);
		String activity = null;
		if (this.truckStatus.equals("idle"))
				activity = "truckIdle";
		else if (this.truckStatus.equals("busy"))
				activity = "truckBusy";
		else if (this.truckStatus.equals("anticipating")) {
				addBehaviour(new LogBehaviour(this, "truck", "kmSetup",
							Long.toString(calculateAnticipateKM())));
				activity = "truckAnticipating";
		}
		addBehaviour(new LogBehaviour(this, "truck", activity,
					timeActivity));
		this.truckStatus = newTruckStatus;
		statusChanged = currentTime;
	}

	/* ------------------------------------------------ */
	public void setInitiatorInDF() {
		// Register that the Truck has become the initiator
		ServiceDescription tempSD = new ServiceDescription();
		tempSD.setType("initiator");
		tempSD.setName(getLocalName() + "-initiator");
		sd[1] = tempSD;
		modify(this, sd);
	}
	/* ------------------------------------------------ */
	public void setParticipantInDF() {
		// Register that the Truck has become the initiator
		ServiceDescription tempSD = new ServiceDescription();
```

```
			tempSD.setType("participant");
			tempSD.setName(getLocalName() + "-participant");
			sd[1] = tempSD;
			modify(this, sd);
		}
		/* ----------------------------------------------- */
		public String getCurrentOrderPickUp() {
			return currentOrderPickUp;
		}
		public String getCurrentOrderDelivery() {
			return currentOrderDelivery;
		}
		public String getCurrentOrderReturn() {
			return currentOrderReturn;
		}
		public long getStartTime() {
			return startTime;
		}
		/* ----------------------------------------------- */
		/**
		 * Translates a truckstatus number to human readable string.
		 * @param s the number
		 * @return the string
		 */
}
```

Appendix C

The InitiatorBehaviour Source code

```
package agents;
import java.util.Random;
import jade.core.behaviours.CyclicBehaviour;
import jade.domain.DFService;
import jade.domain.FIPAException;
import jade.domain.FIPAAgentManagement.DFAgentDescription;
import jade.domain.FIPAAgentManagement.ServiceDescription;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.MessageTemplate;

public class InitiatorBehaviour extends CyclicBehaviour {
		private static final long serialVersionUID = -
		2662106840521483150L;
		private TruckAgent myAgent;
		/**
		 * Constructor.
		 * @param a the agent this behaviour belongs to
		 */
		public InitiatorBehaviour(TruckAgent myAgent) {
				super(myAgent);
				this.myAgent = myAgent;
				// Find the first order.
				findOrder();
		}
		/* ---------------------------------------------- */
		/**
		 * This behaviour is activated when a REQUEST-message is send to the agent.
		 *
		 */
		public void action() {
				MessageTemplate mt = MessageTemplate.and(

MessageTemplate.MatchPerformative(ACLMessage.REQUEST),

MessageTemplate.MatchConversationId("FIND_ORDER"));
				ACLMessage msg = myAgent.receive(mt);
				if (msg != null) {
						// INFORM-message received
						myAgent.debug("Received a REQUEST-message from "
									+ msg.getSender().getLocalName() + ";
now acting as initiator.");
						// ... find a new order.
						findOrder();
				} else {
						block();
				}
		}
		/* ---------------------------------------------- */
		/**
		 * FindOrder. First check whether another truckagent is initiator (by consulting  the DF)
```

```
IF yes, check again after a couple of seconds. IF not, claim the
	 * initiator status (update the DF description) and start the process
	 */
	private void findOrder() {
		int numberOfInitiators = 0;
		// Double check, by consulting the DF
		try {
			DFAgentDescription dfd = new DFAgentDescription();
			ServiceDescription sd = new ServiceDescription();
			sd.setType("initiator");
			dfd.addServices(sd);
			DFAgentDescription[] result = DFService.search(myAgent, dfd );
			numberOfInitiators = result.length;
		} catch (FIPAException fe) {
			fe.printStackTrace();
		}
		// In case there is another initiator active, add a new
		DelayFindOrder behaviour
		if (numberOfInitiators > 0) {
			Random generator = new Random();
			int timeToWait = 1000 + generator.nextInt(19000);
			// Another agent holds the initiator-lock, try again later.
			int period = (int) (timeToWait / myAgent.getRate());
			if (period < 1000)
				period = 1000;
			myAgent.addBehaviour(new
DelayFindOrderBehaviour(myAgent,
					period));
			// No other initiator, claim the initiator position, and start
			search for new order.
		} else {
			// Update the DF
			myAgent.setInitiatorInDF();
			// Start looking for a new order.
			myAgent.addBehaviour(new FindOrderBehaviour(myAgent));
		}
	}
}
```